SOVIET ADVANCED TECHNOLOGY:
THE CASE OF HIGH-PERFORMANCE COMPUTING

by

Peter Wolcott

A Dissertation Submitted to the

COMMITTEE ON BUSINESS ADMINISTRATION

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 9 3

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: _____

ACKNOWLEDGEMENTS

This work more than a piece of scholarly research. It is a tapestry weaving together not only data and analysis, but also the lives and experiences of all those who have had a part in it, or whose work is described. I value the human element embodied in it highly.

I have been privileged to work in The Mosaic Group. This study would have been much more difficult had I not had access to the expertise of the members of the group, the data they have collected, and the research tools they have built. Professor and Mrs. Seymour Goodman, Professor Tom Jarmoszko, Dr. Kevin Lynch, Professor William McHenry, Dr. Ross Stapleton, Dr. Joel Snyder and many current and former Mosaic members have given me greatly valued information, criticism, advice, and friendship across many years and many countries.

Each of my committee members, Joey George, Seymour Goodman, William McHenry, and Ram Nidumolu, made distinct contributions to this dissertation. It would have been a much weaker study without the contribution of each.

Without the role played by Sy Goodman, this dissertation would not have been written. Since 1985 he has provided the opportunity, support, guidance, criticism, companionship, and encouragement necessary to carry this work to completion.

This dissertation would have been impossible to write without the assistance of many scientists and engineers who have been intimately involved with Soviet computing, and the development of high-performance computing systems. The friendships and professional relationships I have established with them are particularly meaningful. I cannot list all those who helped me by providing information and insights here. I am, however, particularly grateful for the assistance of the following individuals: B. A. Babayan, N. Ye. Balakirev, V. S. Burtsev, M. N. Dorozhevets, O. S. Gus'kov, V. V. Ignatushchenko, I. I. Itenberg, V. P. Ivannikov, A. V. Kalyayev, Yu. V. Kapitonova, V. Ye. Kotov, A. A. Letichevskiy, V. K. Levin, I. L. Medvedev, V. S. Mikhalevich, N. N. Mirenkov, I. N. Molchanov, A. S. Nabatov, V. M. Repin, V. V. Rezanov, G. G. Ryabov, Yu. P. Selivanov, I. D. Sofronov, V. A. Torgashev, E. Kh. Tyugu, A. V. Zabrodin, A. P. Zamorin.

DEDICATION

This study is dedicated to those who made it possible for me to complete it and maintain my equilibrium:

my parents, Calvin and Berit Wolcott, who have been a continuous source of love and encouragement for many years;

my wife, Ellie, who is a great joy to me, and persistently reminded me that ‘‘it’s the dissertation, stupid!’’[1] and

my daughter, Elisa, who continues to remind me how many things in life are more important than this dissertation.

[1] With apologies to Bill Clinton and his campaign staff.

‘‘Everything has become more difficult now,
but it wasn’t any easier before...’’

–Soviet HPC engineer, 1991

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This study uses Soviet high-performance computing (HPC) as a vehicle to study technological innovation, organizational transformation, and the R&D of advanced technologies in centralized-directive economies in the past and during periods of transition. Case studies are used to identify the factors most strongly influencing the evolution of high-performance systems and the facilities within which they were developed.

Although closely tied to the military, the HPC sector was not able to overcome basic systemic and technological difficulties. HPC illustrates the limits of centralized-directive economic management's ability to coordinate and prioritize development and production of highly complex, rapidly evolving technologies. Projects were delayed by complex bureaucratic structures, the monopolistic nature of the supporting infrastructure, and resistance of production factories. Progress of individual projects was dependent on the degree to which they drove supporting industries, used immature technologies, had an industrial vs. academic orientation, and were developed in conjunction with production facilities.

The benefits of the reforms–direct contacts between organizations, increased local control of finances and research, greater flexibility in the management of R&D, and improved opportunities for international contacts–have been overshadowed by economic decline and fundamental weaknesses in the supporting infrastructure. R&D facilities have been transformed into a collection of loosely-coupled semi-autonomous organizational units, increasing short-term viability, but threatening their ability to carry out large-scale, long-term, integrated development. Links between R&D and production facilities have been disrupted. The upstream infrastructure remains ill-suited for providing the technologies necessary for HPC development. Preconditions to long-term viability are restoration of the integrity of the development-production cycle and reduction of the HPC sector's

dependency on domestic industries. Taking advantage of mass-produced Western technologies will require changes in philosophies of development and architectural approaches.

The concept of a unified sector-wide technological paradigm is not well suited for explaining the diversity of architectural approaches and specific development trajectories. A paradigm consisting of layers of "micro-paradigms" better captures the patterns of continuity and change within projects and features shared between projects.

This study suggests that the nature of the revenue stream and the opportunities for alternative organizational forms have a significant influence on organizational structure.

CHAPTER 1. INTRODUCTION

More than any other country, the Soviet Union was committed to science. The number of scientists per capita and the prestige they enjoyed exceeded that of any other country. Marxist-Leninist ideology viewed science and scientific methods as key to the development and management of the economy. In support of scientific advance, a huge infrastructure of educational, research and development, and production institutions was established.

Achievements in ‘‘big science’’ and advanced technologies in particular were used to validate the Soviet centralized directive form of management of science and the economy in the eyes of the domestic population and the international community. Soviet scientists and policy-makers frequently boasted of the advantages of such a system in marshaling huge resources and focusing them on priority programs such as space exploration, nuclear power, military systems, oil and gas development, computing and others.

Advanced technologies are characterized by their complexity in design and/or manufacture, and a rapid evolution which often establishes the state-of-the-art in the field. Their development depends on a vast infrastructure of supporting industries. Thanks to national security concerns and controls in both the Soviet Union and the Western members of the Coordinating Committee for Multilateral Export Controls (CoCom),[1] complete infrastructures for advanced technologies with military applications in the Soviet Union had to be developed independently of the corresponding Western industries, even when those technologies–so-called dual use technologies like computers–had significant and widespread civilian applications.

[1] CoCom includes Japan and all the member nations of the North Atlantic Treaty Organization (NATO) except Iceland.

In spite of some successes, Soviet achievements in advanced technologies and science more generally have lagged behind those of the West when measured by the level and quantity of results, number of Nobel Prizes earned, etc. From the 1960s through the mid 1980s, the Soviet economy as a whole ‘‘stagnated,’’ experiencing declining growth rates and growing inefficiencies, and an increasing technological lag behind the Western capitalist countries. Against this backdrop, Mikhail Gorbachev launched a program of economic restructuring *(perestroika*) in 1985 in an effort to revitalize the USSR economy. Above all, the economy was to be reformed through the acceleration of scientific and technical progress in a new stage of the scientific-technical revolution (NTR). This stage was to be characterized by accelerated development and assimilation of new technologies, including advances in microelectronics and computing, computer-aided manufacturing, materials, etc. in an effort to increase the output and efficiency of the economy without dramatically increasing inputs.

In the eight years since the start of *perestroika,* the Soviet Union and its successor states have experienced profound economic, political, and social change far exceeding the intentions of Gorbachev and other leaders. Indicators such as the rapidly falling gross domestic product, crime rates, and runaway inflation paint a picture of catastrophe. However, such indicators do not help us understand the underlying phenomena and processes which have led to these results. They do little to help us understand the degree to which, within the chaos, the foundation is being laid (or not being laid) for the future rejuvenation of the economies of the former Soviet republics and, in particular, their scientific and technical base. To understand these transformations one must look not only at macro-level indicators, but also at the processes at work at the micro-level–in individual technologies, research teams and institutes, and within various branches of the economy.

In this study we seek to contribute to an understanding of the development and transformation of Soviet advanced technologies and their organizational and environmental context, both prior to *perestroika* and during the reform period from 1985 through December, 1992, approximately one year after the dissolution of the Soviet Union. Our vehicle will be a detailed examination of research and development (R&D) facilities and projects within the Soviet high-performance computing (HPC) sector. We wish to document the development of HPC systems within these organizations, identify the factors which have shaped their development, and analyze the impact of the social, political, economic and technical changes since 1985 on these technologies and the organizations within which they have been carried out. Through such an analysis, we hope to cast light on the prospects for HPC in particular, and advanced technologies more generally in the Soviet Union's successor states.

Since the dissolution of the Soviet Union in December, 1991, terminology has become particularly problematic. The Soviet Union has been replaced as a political entity by fifteen independent countries, some of which have formed a loose confederation called the Commonwealth of Independent States. All of them have sought to distance themselves from the Communist legacy and have removed the words ''Soviet'' and frequently ''socialist'' from official names. At the same time, it is impossible for any of these states or the organizations and people within them to wipe the slate of history clean. They are bound to and influenced by their history in the Soviet Union. They share many problems, features, and trends precisely because they once were part of the Soviet Union. For this reason, we will use the terms ''Soviet'' and ''Soviets'' to refer to institutions and individuals which had their origins in the Soviet Union and which still exhibit the influence of the Soviet economic, political, technological, and social systems, even when we refer to events after the Soviet Union's breakup. We will refer to the Soviet Union's suc-

cessor states as ‘‘Soviet states,’’ and the republics of the Soviet Union as ‘‘Soviet republics.’’ Only when the distinction is important will we use ‘‘Russian,’’ ‘‘Ukrainian,’’ etc. We are also likely to use the latter when refering to institutions which originated (rather than metamorphised) after December, 1991 and are closely associated with one particular state.

High-performance computing provides an excellent vehicle for studying advanced technology in the Soviet Union and its successor states because of the nature of the technology itself, the nature of the infrastructure which supports this sector, and the importance of this particular technology to many critical spheres of activity in the military and civilian sectors, and its geo-political implications. Its study can cast light on a broad spectrum of issues, from technological innovation, organizational transformation, the Soviet reform process, the conduct of research and development, military and economic competitiveness, technology transfer of dual-use technologies and export control policies, and others.

High-performance computing represents one of the Soviets’ ‘‘best shots’’ in computing. In a tradition stretching back to Lenin, Soviet leaders have viewed technological advance as a key to economic development. The acceleration of scientific-technical progress and the wide-spread introduction of new generations of technologies was proclaimed a ‘‘key lever in the intensification of the national economy’’ [Vasi88]. They were a prominent feature of the ambitious international Comprehensive Program for Scientific-Technical Progress of the CMEA[2] Countries to the Year 2000 (Program to the Year 2000). In this program and for many year prior, high performance computing has

[2] The Council for Mutual Economic Assistance (CMEA) included members of the Warsaw Pact plus Cuba, Mongolia, and Viet Nam. The latter three played an insignificant role in CMEA computing. Those having the greatest impact on computing in the Eastern Bloc were Bulgaria, Czechoslovakia, the German Democratic Republic, Hungary, Poland, and the USSR.

been a high-profile, high-priority advanced technology. Not only are powerful computers tools to be used in scientific, economic, and military activity. They are symbols, often viewed as indicators of a nation's technological capability. The Soviet high-performance computing sector has drawn considerable financial, material, and human resources over many years. Because it has been one of the most innovative sectors of Soviet computing, providing rich research opportunities, high-performance computing has attracted some of the leading computer scientists and engineers in the country. A study of Soviet high-performance computing can shed light on what Soviet science has been capable of in a very important field.

High-performance computers are highly complex systems. They typically consist of thousands of components and subsystems, each of which can be considered a system with hundreds or thousands of elements. Hardware is highly complex, but does not reflect the full extent of a system's complexity. Basic hardware units such as gate arrays are altered to reflect a highly complex logical design. Software can be even more complex, since software designers tend to avoid duplicating code which serves the same, or similar functions. Computers are highly dependent on miniaturization. The ability to place thousands, or even millions of transistors on a single chip measuring only a centimeter or two on a side requires complex, high-precision manufacturing tools and extraordinarily pure input materials and processing conditions; such components cannot be created manually by skilled craftsmen with ''golden fingers.'' As a result, computers have greater complexity per unit volume than any other device created by man. Through a study of these systems, we can identify strengths and weaknesses of the Soviets' ability to design, construct, and manufacture complex technologies.

The final product is dependent on a long chain of activities and products such as fundamental research, applied research, materials production, component and subsystem de-

velopment, machine architecture design, unit assembly, design and production technologies, and systems support services. Weaknesses in any link of the infrastructure can have serious consequences in the final product. Because HPC systems and their infrastructure are so dependent on each other, an examination of the end products can cast much light on the nature of the infrastructure and the changes that it might be undergoing. HPC systems can contribute much to our understanding of Soviets' ability to develop and manufacture advanced technologies with are also complex and highly sensitive to a wide range of inputs and upstream products and processes, both prior to 1985 and after. The study of high-performance computers is futher important because they not only are an output of this infrastructure, but also help shape its nature. They are critical tools for the development of next generation systems and components, and in both the East and the West traditionally have been among the most advanced systems, creating demands which have caused the entire infrastructure to raise its technological level.

The Soviet high-performance computing sector is inter-administrative. To a greater degree than other parts of the computing and electronics sectors, high-performance computing involves the participation of a range of administrative entities. Nearly all projects include significant research at Academy of Sciences or VUZ (Institutions of Higher Education) institutes. Research, development, and production were carried out in industrial ministries such as the Ministry of the Radio Industry and the Ministry of Instrument-Building, Means of Automation, and Control Systems (Minpribor). Inputs such as integrated circuits, materials, and selected subsystems come from other ministries. The military also has played a significant role as a key customer and project sponsor. Decisions about HPC involved policy makers at the highest levels of the Academy of Sciences, branch ministries, the USSR Council of Ministers, state agencies, and the Military-Industrial Commission (VPK). Successful HPC development required the coordination

and cooperation of multiple administrative entities. As a result, Soviet high-performance computing provides a lens into basic decision making from the highest levels of government down, R&D processes within many industrial and academic branches of the economy, and the nature of the relationships between them. HPC is therefore also a useful vehicle for examining the changing roles of these organizations and their interaction.

High-performance computers are important enabling technologies in many areas of science and industry. Increasingly, supercomputers are becoming an essential tool in the development of advanced technologies in aerospace, computing, weapons design, materials and others. These industries rely on supercomputers for design, analysis, simulation, and visualization to improve product quality and functionality, shorten development cycles, and reduce labor and material costs. Computational methods have emerged as a third pillar of scientific advance, together with analytical and experimental methods. Numerical analysis and visualization have emerged as basic tools in a growing number of fields such as weather prediction, hydrodynamics, plasma physics, atomic and molecular structure, etc. If the Soviet states are to be broadly competitive internationally in these and other disciplines, their researchers and engineers must have access to high-performance computing systems.

High-performance computers are also useful for examining geo-political aspects of advanced technology. HPC systems are an excellent example of a dual use technology with broad applicability to both military and civilian applications. Few dual use technologies have been as consistently and tightly controlled by the CoCom nations as HPC. At the same time, HPC underscores the issues and difficulties surrounding export control regimes during the post-Cold War era. Thanks to rapid technological developments, high-performance technologies are becoming much more difficult to control. Western high-end supercomputers, still manufactured in small quantities and requiring consider-

able on-site support, remain relatively controllable. Advances have been so rapid, however, that today's workstations and mid-range systems offer functionality and performance available only on supercomputers a decade ago. These systems are often manufactured in large, uncontrollable quantities (100,000 or more) and are small enough that transporting them is trivial. Leading-edge technologies move into the commercial mainstream only a few years after introduction, and many Western massively parallel high-performance computers contain a very high percentage of off-the-shelf parts. At the same time, the relationship between the West and the Soviet states has become much less adversarial, but the latter's future stability and sensitivity to the former's national security concerns are by no means assured. These factors are forcing a reconsideration of Western and Russian export control policies [Nrc93].

Because the Soviet high-performance computers have for decades been developed independently of the West in a rather closed sector with limited transfer of technology from outside the Soviet Union, they provide a microcosm in which to study issues of technological innovation and the nature of technological advance within a specific set of social, economic, and political systems. Prior to the *perestroika* reforms, these systems were rather stable. Since 1985, they have experienced drastic changes, and the isolation of this sector from the West has eased considerably. Under these conditions, Soviet HPC technologies should help us learn more about technological innovation during highly dynamic periods which create drastically new opportunities and constraints.

Similarly, Soviet HPC provides a microcosm in which to study organizational development and transformation under changing conditions. Traditionally, organizational structures in the Soviet Union were very stable and given types of organizations throughout the country had very similar structures. The reforms have created great pressures on organizations and radically new opportunities for transformation.

Besides the reasons mentioned above, the Soviet HPC sector provides a useful vehicle for studying these issues because it is relatively small. It is possible to study nearly all the major projects and R&D centers in the entire sector. In doing so, we avoid many of the methodological problems associated with selecting a study sample and generalizing to the entire sector.

We will investigate HPC-related activities within one component of the Soviet high-performance computing sector–research and development facilities in industry, the Academy of Sciences, and the institutions of higher learning (VUZ). As pointed out in [Meye90], a study of the activities of individual firms should not be equated with the study of an industry or industrial sector as a whole. But a study of research and development facilities and their products provides an important piece of the over-all picture. R&D facilities are crucial to the health of advanced technology industrial sectors. In them the ideas, abstractions, and theorizing meet the realities of the physical world. Innovation is strongly shaped by not only the theoretical ideas generated, but also the opportunities and constraints imposed upon the process by a host of material, technical, social, and political factors. The results of R&D strongly affect the quality, performance, and innovativeness of products manufactured within an industrial sector. While improvements in R&D do not necessarily lead to improvements in the output of an industrial sector as a whole, a deterioration within R&D facilities will almost certainly have a negative impact on the sector's output. R&D facilities are also a critical point of interaction between sponsors, planners, developers, manufacturers, and users. While this study focuses on HPC R&D facilities and the technologies developed within them, we will necessarily touch on these other elements as well.

In chapter 2 we discuss several bodies of related research and the contribution a study of Soviet HPC can make to each. We also present the specific research questions and the

case study methodology used to address them. Chapter 3 gives an overview of the HPC sector. We present some of the sector's history, and place the numerous industrial and academic systems in the context of the sector as a whole. This sets the stage for three detailed case studies, covered in chapters 4, 5, and 6. Chapter 7 contains sections that discuss HPC projects (or sets of projects) not covered in the detailed studies. This chapter ''rounds out the field,'' making our study quite comprehensive in its coverage. It also illustrates that the processes at work in projects discussed in earlier chapters are not unique, but apply more broadly throughout the sector. Chapter 8 provides a cross-cutting analysis of the individual projects and conclusions drawn from this study. In chapter 9, we discuss the implications of our study for Soviet HPC developers and users, and Western and Russian policy-makers, and suggest directions for future research.

CHAPTER 2. RELATED RESEARCH AND METHODOLOGY

Our study extends four areas of research: innovation in the Soviet Union, computing in the Soviet Union, Western research on innovation, and organizational development. In this chapter we discuss the literature in these areas, the research questions, the research methodology and the conceptual framework which serves as a platform for the study.

2.1 Technological Innovation in the Soviet Union

2.1.1 Characteristics of Soviet Science

Loren Graham has commented that the Soviet Union was ‘‘a nation with an explicit commitment to science, including a value system and a philosophical world view based on science, which is unmatched in intensity by any other nation in the world’’ [Grah75, 12]. Although Soviet thinking about the role of science in society and the drive towards a communist society has varied since the inception of the Soviet state, it has always figured prominantly. During the 1960s science was given enhanced status when it was declared a direct productive force in its own right, on a par with production, rather than something which lagged behind, and was subordinate to, production. The post-Stalin years saw a rapid increase in the number of researchers and research facilities. Statistics on the number of research personnel are difficult to obtain, interpret, and compare, but many who have examined this question have concluded that the USSR had considerably more people working in research than any Western country, both overall and as a percentage of the working population [Fort90, 8]. High-ranking scientists have enjoyed considerable prestige and benefits in the Soviet society. Science as a whole has been rather well funded for many decades under Soviet rule.

Nevertheless, the Soviet Union trails the United States in many common indicators of scientific performance: number of Nobel Prizes, origin of major breakthroughs, fre-

quency of citation by fellow specialists [Gust80, 31]. Both Western and Soviet writers have for many years been critical of the Soviet Union's inability, with some exceptions, to generate world-class research results and innovations, and see the latter through into production and use within the national economy, particularly from the 1960s to the present. (See [Fort90; Nolt88] for references to this literature.)

A number of researchers have identified some of the general features of Soviet science. Thane Gustafson describes five dominant charateristics: 1) Soviet pure science is strongest in fields that depend the least on material support (instrumentation, sophisticated materials, and equipment); 2) Soviet science is often slow to accept radical conceptual changes or take up new approaches; 3) in several fields, Soviet scientists have achieved results through long-term efforts in traditional specialties and established methodologies; 4) when they do make crucial breakthroughs, Soviet scientists are often unable to maintain their lead; 5) Soviet science holds leading positions in fields that enjoy high-level attention [Gust80, 32-33]. Others focusing more on applied research point out the long implementation cycles, weak links between science and production, the reluctance of industry to use the results of science, the relative isolation of the Soviet scientific community from its Western counterparts [Fort90].

2.1.2 Analyses of Innovation in the Soviet Union

Western and Soviet scholars have identified numerous factors which have contributed to the state of Soviet science and its ability to produce quality, useful results. Joseph Berliner's study on the decision to innovate in Soviet enterprises was the first in-depth look at the question of innovation in the Soviet context [Berl76]. Berliner does not attempt to explain fully the rate of innovation throughout the entire economy, or the genesis of innovation itself. Rather, he focuses on the civilian industrial sector and deals with the factors which affect the assimilation of new products and processes into production, rather than

those that impact the earlier stages of technological process [Berl76, 3]. Neither does he address questions of economic and science policy, the technical characteristics of society such as tastes and values which influence demand, or historical or cultural influences. The four structural properties of the economy which he examines are prices, rules which guide decision makers, incentives, and the organizational structure of the economic system as a whole [Berl76, 12]. Organizational structure includes both the units that comprise the system and the ways in which they relate to one another [Berl76, 29].

In a series of two volumes [Aman77; Aman82], Amann and Cooper have assembled a number of case studies examining the technological level of various branches of industry and technological innovation within the Soviet Union. These volumes have tried to provide a greater level of detail about the technological level of Soviet industry, and address what they feel is a weakness of the general surveys–their tendency to present general patterns which reveal little of the patterns of variation between industries, and their emphasis on national policy and planning or on the individual enterprises with little exploration of the intermediate relationships between the ministries and their subordinate research and development institutes [Aman82, 8].

The Berliner and Amann and Cooper studies complement each other, and provide a solid view of technological innovation under the pre-*perestroika* system. Much has changed since 1985, however, and a goal of our research is to examine how the process of technological innovation has changed, within one sector. [Aman82] and [Berl76] suggest that given the changes observed in the Soviet Union, one might reasonably expect to see significant change in the process of technological innovation. In his introduction, Amann comments that ‘‘[a]bsence of competition and user feedback represents perhaps the most potent single factor inhibiting the pace and scope of innovation in the USSR’’ [Aman82, 12,254]. Berliner suggests that ‘‘[a]ny structural reforms designed to acceler-

ate the rate of innovation must therefore alter the traditional balance of reward and risk....The appropriate alteration of the balance of risk may be accomplished by a structural change designed to increase the degree of enterprise autonomy over its transactions with other enterprises and organizations'' [Berl76, 522]. Hage calls the amount and variety of strategic decisions made by members of the organization organizational autonomy and regards it as a significant variable [Hage80, 387]. As we shall see below, the reforms have introduced greater levels of competition, feedback from users, and local autonomy. Are these changes bringing about the improvements in innovation predicted by Berliner, and Amann and Cooper?

Other researchers have examined innovation as part of a broader look at scientific research and development as a whole, within the Soviet context. Many features of science more generally and technological innovation in particular are directly or indirectly related to the centralized nature of the management of the economy. While it is generally not the case that science projects are determined in a ''top-down'' fashion with individual scientists having little input in the planning process [Fort86], the Soviet system was characterized by vertical administrative controls which strongly affected the allocation of resources, the inter-organizational coordination, and the indicators and incentives which shape R&D and production. Bruce Parrott describes some of the administrative rigidity of the management of science even during the 1960s and 1970s when efforts were being made to decentralize somewhat the highly centralized system inherited from Stalin [Parr80, 75]:

> Administratively, all changes in the internal structure of individual scientific research bodies continued to require the consideration and approval of their ministerial overseers. Financially, research units were asked to submit highly detailed annual budgets, and once these estimates were approved, transfers among budgetary categories on the units' own authority were not permitted. In addition, these estab-

> lishments depended on centralized material-technical supply; they could not, without higher approval, sell or trade equipment that was no longer useful to them. Finally, in selecting research topics, each unit had to submit for the confirmation of its superior ministry or department a thematic research plan that specified most of the research to be undertaken in the coming year.

Besides limiting a research institute's ability to respond in a flexible manner to changing research conditions or advances in the field, such an arrangement can greatly increase the length of research cycles. Needs must be specified months or years in advance with little guarantee that they will be fulfilled. It is difficult under these circumstances to make changes–to acquire unanticipated supplies or equipment–in the middle of a plan period [Gust80, 52].

The centralized, directive-planning system does more than reduce the flexibility of individual organizations; it can also skew a research effort away from providing the ''best'' results under a given set of circumstances. One characteristic of the Soviet system of economic management was the use of a variety of indicators by which to measure progress and the fulfillment of plans and goals. Alec Nove has identified the fundamental problem with using quantitative aggregates (tons, meters, rubles, numbers of units, etc.) as indicators [Nove86, 75-112]. If a quantitative aggregate is used to measure success or failure, it tends to distort the production process in favor of this indicator. For example, if the primary indicator is numbers of microchips manufactured, then reliability is likely to be sacrificed to meet the target plan. If the total value of output is the primary indicator, then the product mix is likely to shift in favor of the more expensive products, etc. Aggregate indicators cannot give the proper priority to all necessary product parameters at the same time without overwhelming the planning process with information.

In practice, although multiple indicators were included into a given plan, some indicators were understood to be more important than others. Thus in scientific research, pro-

viding a finished piece of successful research was often less important than making sure a given amount of money was spent on research in a specified timeframe [Fort90, 133]. The plan would be fulfilled when the money was spent, not when a successful result was delivered. To be sure, this picture is oversimplified and results were not unimportant, but the overall effect of many of the planning mechanisms was the decoupling of funding and provision of inputs from the provision of high-quality research, and to substitute easily-measured indicators in place of research ‘‘success’’ (something very difficult to measure in the abstract).

Traditionally, the argument for centralized management of science has centered on the ability to concentrate resources on priority projects and to reduce duplication and waste of effort. As Parrott notes, such management has a number of weaknesses, including the possibility that the administrators do not have the expertise necessary to make the best allocation of resources or evaluate the return from such allocation, the elimination of competition which serves as a powerful motivator in the West, and the reduced flexibility and ability to respond quickly to new developments [Parr80, 72-73].

Another consequence of the hierarchical nature of the Soviet economic system was departmentalism (*vedomstvennost’*). The lines of authority, planning, and resource allocation ran vertically, from enterprises and institutes up through their respective ministries. Horizontal links between organizations typically were established higher up in the hierarchy. The greater the ‘‘administrative distance’’ between two organizations, the higher one needed to go in the respective hierarchies to establish contact. Departmentalism tended to make links between players in the development and supply cycles longer, more rigid, and on the whole less efficient.

Departmentalism has been a particularly strong factor in the relationships between Academy research institutes and ministry production facilities. During the early 1960s, in

an effort to delineate more clearly applied and fundamental research, many production and applied R&D facilities were transfered from the Academy to industrial ministries. Ideally, under this arrangement, Academy institutes would transfer their results to industry, at which point industrial design bureaus would advance the technology to the point of series production. In practice, the Academy typically encountered resistance to its results by the ministries, from the production facilities, branch science, and especially the lead institutes in branch science [Lakh90, 40]. Lakhtin summarizes the most significant reasons for this state of affairs [Lakh90, 42]. Among them are the ‘‘not invented here’’ syndrome (not uncommon in the West, either) which makes branch science reluctant to invest time, energy, and facilities in furthering Academy projects, the overloading of production facilities with their own plans, and competition with ideas and research directions originating in the industrial ministries.

Lakhtin views the root cause as the lack of an effective, single authority overseeing all of science; in otherwords, insufficient centralization. While the Academy and the State Committee on Science and Technology (GKNT) in principal had responsibility for establishing national research policy and coordinating research efforts throughout the economy, in practice they had little direct influence over ministerial R&D and production facilities. It is not clear that such a single authority would have resolved the problems, however. The experience within individual ministries showed that departmentalism played a significant role even when a single individual, the minister, had authority over the whole ministry. There was always resistance to getting organizations or organizational components to work together when such work ran counter to their perceived interests.

A further hinderance to innovation has been the lack of slack resources. Gustafson points to the lack of instrumentation and adequate supplies as a principal factor shaping

the state of Soviet science [Gust80, 48-54]. Clearly, research projects are hindered if high-quality tools and supplies are not available, or require considerable time and effort to acquire. Gustafson cites as reasons for this the insufficient production of supplies and instruments; general lag in sophistication; slow and unresponsive planning of supply; lack of effective communication between users and producers of equipment; and a lack of coordination among industrial ministries.

The Soviets also have provided analyses of the state of their scientific establishment and its ability to innovate and have proposed many measures to improve it over the years. Louvan Nolting has summarized the Soviet perspective in [Nolt88]. Soviet publications suggest five major reasons for shortcomings in the innovation process [Nolt88, 2-4,37-105]:

- There has been a lack of capital investment in the advancement of innovations, limiting the resources available to move an innovation from R&D to production. In addition, depreciation policies designed to extend the use of existing equipment slows the turnover of old equipment. Existing capital has, overall, favored current production over innovations.
- A shortage of necessary input materials to carry out R&D projects hinders both the manufacture of new products and the development of prototypes. Shortages are aggravated by incomplete standardization of industrial items and the low degree of specialization in production, making it difficult either to build new items on the basis of existing technology or to acquire the specialized technology needed for many projects.
- A poor integration of innovation plans with production and other economic plans means that supplies needed for innovation are poorly coordinated, that

innovations which take place are often poorly suited for actual production conditions, and that the various stages of R&D are poorly coordinated.

- Bureaucratic barriers prevent or delay communication between developers of innovation and producers.
- The incentives for production frequently are at odds with those for innovation, causing producers to favor production over innovation.

As Nolting points out, while there is considerable overlap between Western and Soviet analyses of innovation in the Soviet Union, the Western analyses tend to place greater emphasis on problems which they view as inherent in the centralized directive system of economic management [Nolt88, 111]. Generally speaking, Soviet analysts tend to stress the lack of sufficient capital and supplies, and the inefficiency of many procedures needed to carry out innovation [Nolt88, 112]. Implicit is the notion that to improve the state of innovation, one need only make the existing system function better.

2.1.3 Efforts to Reform Soviet Science

So far we have not discussed efforts that the Soviets have made to reform their system to encourage innovation and improve the state of science. Since the Soviet Union was formed reform programs have been carried out on a regular basis to address the problems identified above. These efforts, which we briefly describe below, reflect some of the underlying tensions which have shaped Soviet science as it has evolved. The primary tensions have been between centralized and decentralized mechanisms for managing science, the use of directive versus economic mechanisms as the agents of management, and the balance between fundamental and applied research and which organizations should be responsible for which.

Lakhtin has broken down the evolution of Soviet science into four phases [Lakh90, 6-19]. The first, from the October Revolution to the late 1920s, was marked by efforts to organize science in the service of socialism. Science was not considered a productive force in itself, but rather was subordinate to efforts to increase the means of production. Organizational differentiation was carried out as research institutes were created, and organizations such as the Academy of Sciences were transformed from a ‘‘collection of scientists’’ to a scientific organization with a growing number of research institutes subordinate to it [Lakh90, 8]. Increased emphasis was placed on the training of specialists, with the creation of a system of institutions of higher education (VUZ).

The second stage, from the late 1920s to the mid-1950s, the reign of Stalin, was marked by a sharp move towards a strong form of centralization. The prevaling philosophy was that science was subordinate to philosophy and Party ideology [Fort86, 15]. These years were characterized above all by a very rapid pace of industrialization within the framework of five-year plans. A large number of research institutes were created, chiefly within industry with a strong emphasis on applied research. A characteristic of this period was the drastic absorption of science (as well as the rest of industry) into a centralized directive form of management. For example, 1936 marked the beginning of a sharp decrease in contract work (*khozdogovor*) carried out between a research institute and an industrial customer. In 1932 57% of all research done in branch institutes was carried out under contract; by 1937 it was only 14%. By 1950, 100% of the work of scientific research institutes was financed through the state budget, or 0% under *khozdogovor* [Lakh90, 157].

During the third stage, from the mid-1950s to the mid-1980s, science began to be treated as a direct production force in the national economy and grew into a large branch of the economy. The system of management of science, exemplified in the creation and

growth of the State Committee on Science and Technology (GKNT), expanded considerably. Under the campaign for ‘‘scientific-technical progress’’ increasing numbers of large-scale scientific programs were initiated and corresponding super-institutional management organizations and systems were created. New branches of science were created (electronics, atomic energy), as were many new research institutes and research centers such as the Academic City (*Akademgorodok*) in Novosibirsk.

A more scientific approach to administrative science and management of the economy emerged after 1956 based on econometric models, information processing and feedback, and scientific management of labor [Thom83]. Such economic measures also penetrated the management of science with the evaluation of scientific results through the use of economic indicators. Inherent in such efforts was the notion that science was a productive force in the economy and as such a partner to production.

During the third phase of Soviet science, a number of efforts were made to reform the scientific system. During the mid-1950s the ministerial bureaucracy was coming under increasing pressure to speed up the pace of technological advance. Khrushchev’s reforms of 1957 called for a rather drastic decentralization of industry through the transfer of much of the power and personnel of the central economic bureaucracy to regional economic councils *(sovnarkhozy)* [Parr83, 172-177]. Even in their most radical, proposed form these reforms did not achieve significant centralization by Western standards, but science was granted less decentralization than were other parts of the economy. The State Scientific-Technical Committee (transformed into the State Committee for the Coordiation of Scientific-Research Work in 1961, finally becoming the State Committee on Science and Technology in 1965) was created in 1957 to coordinate scientific research [Lakh90, 29]. At the same time, a large percentage of R&D facilities in the civilian sector were transfered to the jurisdiction of Gosplan, the State Planning Committee, although

Gosplan soon shifted control of them to other state committees as the administrative load became unbearable [Parr83, 174-175]. Some decentralization had taken place, but without the introduction of incentives at the lower levels of the economy, innovation and R&D continued to lag.

During 1961 the Soviet government tried to increase the low-level incentives for innovation through the introduction of a form of *khozraschet*, or economic self-accounting, and self-financing, and improving the dissemination of scientific and technical information [Parr80, 74-75; Lakh90, 157]. In part, this represented an effort to substitute economic links for the organizational links which had been disrupted by the 1957 transformation of industry from a highly-centralized to a more regional structure. With the rapid growth of the economy and the scientific community, there was a need to try to use resources as effectively as possible, and it was felt that *khozraschet* could accomplish this [Lakh90, 157-158]. This form of *khozraschet*, intended to make research more responsive to the needs of concrete customers, was in effect only at the level of enterprises rather than at higher-level administrative structures. As a result, projects involving multiple organizations and higher levels of coordination were often not carried out on *khozraschet* principals. In addition, the version of *khozraschet* implemented in 1961 involved ‘‘payment for the process’’ rather than ‘‘payment for the product,’’ weakening the accountability of the researchers to the customers [Lakh90, 158]. Finally, administrative controls on budgets, material-technical supply, and organizational structure remained very tight [Parr80, 75].

During the late 1950s fundamental science was also coming under attack for, in particular, the low number of Nobel Prizes awarded to Soviet scientists relative to those of the United States or Western Europe. There was a perception that with the reorientation of science in the early 1930s towards applied research, fundamental research had been

neglected. One of the measures taken to increase the relative weight of fundamental science was the transfer in 1961 of 92 Academy of Sciences institutes to the industrial ministries. Not surprisingly, the theoretical underpinings of much applied research was diminished and the Academy lost much of its ability to carry out the prototype development work necessary to pursue avenues of research [Lakh90, 39-40].

Following Khrushchev's ouster in 1964, the Brezhnev administration initiated a broad set of economic reforms, seeking to institute ''scientific decision-making'' and major decentralization to improve the efficiency of the economy. Spearheading the effort was Prime Minister Kosygin, who favored exposing enterprise and institutes to market forces to make them more responsive to customers. Between 1965 and 1969 a series of reforms were instituted which, in the balance, resulted in greater decentralization of management of science, but which also reflected the tension between proponents and opponents of decentralization. The results throughout this period had a compromise nature, incorporating a mix of centralizing and decentralizing measures.

In 1965 the regional system of economic management was converted back to a branch-ministry system with the abolishment of the *sovnarkhozy*. Responsibility for R&D coordination was brought back more tightly under the control of Moscow-based organizations, and production targets, product-mixes, and inter-enterprise exchange of goods were strongly centralized. As Parrott points out, the decrees at the same time removed a number of ministerial controls and put more emphasis on financial levers and incentives to stimulate enterprise performance [Parr83, 215-216].

Officials in the Academy of Sciences and the GKNT, seeing the increased authority of enterprise managers, pushed for a greater loosening of controls in science as well. Their proposals included increasing the role of contract work within Academy institutes and basing allocation of resources more on competition. These proposals met with oppo-

sition both within the scientific community and the government bureaucracy, and the 1967 resolutions regarding R&D facilities also reflected a compromise. Central agencies kept the responsibility of approving research programs for institutes, but the institutes themselves received more freedom in shaping institute staff and in shifting funds among research projects [Parr80, 79-80; Parr83, 223-224; Lakh90, 160]. Profit also became a more important indicator [Lakh90, 160]. The pricing system had a negative effect, however, in that it did not give enterprises enough profit on new products and enabled them to manipulate cost data to obtain high profit levels of the production of products already in series manufacture, encouraging the continued production of the latter rather than introducing new technology [Parr83, 226-227].

Dissatisfied with the progress, the Soviet leadership adopted a wide-ranging package of resolutions reforming science in 1968-1969. The reforms called for the broad-based introduction of competition in R&D, provided for the establishment of personal incentive funds within organizations, created ministerial funds to support projects selected by the technical councils of individual institutes, increased the legal liability of R&D facilities for their contracts.

The *khozraschet* mechanism was further tuned in 1969 with the introduction of an arrangement in which developing organizations could receive a profit of up to 1.5% of the ‘‘economic effect’’ gained from the use in industry of their research results [Lakh90, 161; Parr80, 83-84]. The ‘‘economic effect’’ indicator was largely meaningless [Mche85], but this form of *khozraschet,* involving payment for the process plus a bonus for the effect of the result, was first implemented in the Ministry of Electrical Equipment Industry (Minelektrotekhprom) and later spread throughout industry.

Also introduced in Minelektrotekhprom at this time was a ‘‘Single Fund for the Development of Science and Technology’’ (EFRNT) as an effort to implement a greater

concentration and coordination of research funds and projects [Fort90, 134-135; Lakh90, 161]. This system, in which most of branch R&D spending was drawn from a single centralized ministerial fund and distributed via so-called lead-organizations for large-scale projects, gradually gained acceptance and by 1979 had been implemented in all non-defense industries [Fort90, 134-135]. Another measure which provided a counterweight to much of the decentralization that was taking place at this time was the use of the ‘‘program-goal method’’ (*programmno-tselevyye metody, PTsM*) in the planning of science. This method of planning, which was to govern 25% of total scientific expenditure by the early 1980s, was based on the high-level specification of long-range (15 and 20 year) priorities. Like the EFRNT, PTsM was to give central authorities more focused control of projects, but throughout the country rather than in just a single branch. In addition, PTsM sought to overcome some of the lack of integration between research and production plans [Fort90, 135-136].

In a further effort to close the gap between R&D and production, another 1968 resolution was passed creating ‘‘scientific-production associations’’ (NPO). The underlying idea was to combine R&D and production facilities into a single, relatively low-level administrative structure [Fort90, 115-116]. This resolution was implemented slowly. The resolution was largely voluntary (to be taken ‘‘in necessary cases’’). There was considerable room for interpretation in exactly what an NPO was and how large it should be, and a variety of opinions about how much authority they should be given relative to ministries, *glavki* (main administrations), and individual enterprises. Regional officials tended to favor the creation of small associations, as did the GKNT and Gosplan, while the central ministry officials favored the creation of fewer, but larger associations and a more centralized, Moscow-based, form of management [Parr83, 286-289]. In 1973 an attempt was made to break this log-jam in a resolution calling for wide-spread reorganization of

industry. It, too, represented a compromise between the centralization and decentralization forces. While most ministerial *glavki* were to be abolished and the association was to become the primary economic unit, the associations would range widely in size and scope of authority. The largest were the all-union associations (VPO) which were to have nationwide authority over their industrial sector and made it likely that there would be little effective decentralization of authority [Parr83, 289]. However, it appears that the NPO were successful in decreasing R&D cycles and increasing the number of innovations generated. In his summary of a number of assessments Parrott points out that by some indications the average reduction in R&D cycle time was 25-30%. Nevertheless, it appears that the overall level of innovation on a nation-wide basis, as measured by the number of new types of machines and equipment created, in fact decreased between 1970 and 1976 [Parr83, 290]. Nevertheless, during the 1970s and 1980s the number of NPO grew sharply, from about 80 in 1973, to 100 in 1975, to 250 in 1985 and to twice that number in 1988 [Parr83, 289; Fort90, 116; Lakh90, 204].

Having been largely stripped of its ability to conduct applied research during the early 1960s, the Academy of Sciences gradually grew more involved in development work during the late 1970s and 1980s. In 1977 a number of changes were made to the Academy's statute which included technical sciences within its sphere of appropriate activities. Subsequently some of the applied institutes were returned to the academy and new ones were established [Fort90, 50]. This trend was to have a significant impact on high-performance computing, setting the stage for a number of parallel-computer prototype development projects.

In spite of the reform efforts mentioned above, levels of technological innovation and implementation remained unsatisfactory. The 26th Party Congress called for enhanced measures to accelerate scientific and technical progress as a key factor in intensifying the

nation's economy. During Andropov's tenure, on August 18, 1983, a joint Central Committee-Council of Ministers resolution ''On means of accelerating scientific-technical progress in the national economy'' was issued. Among these measures was the expanded application of program-goal methods in the area of science as a means of converting the 20-year Comprehensive Programs into five-year plans through the use of a hierarchy of all-union, republic, branch, and regional scientific-technical programs [Arkh86, 16; Kush86, 26]; the streamlining of the economic indicators and sources of funding to the components of the NPO; and the initiation of new forms of scientific organization, such as the temporary scientific-technical collectives of which the START new-generation computing initiative, discussed in chapter 6, was the prototype. The resolution also pushed the *khozraschet* mechanism a step further in the direction of ''payment for a product'' rather than for a process, or a process plus bonus, as had been the case in earlier reform measures. Lakhtin sums up the success of the latter as follows: ''But the implementation of this *khozraschet* principal was bogged down in part because of the lack of willingness of the ministries to lose their traditional control levers, and in part by a lack of the resources necessary with which to get their institutes ready'' [Lakh90, 163].

In sum, the rather large number of efforts to reform science in the three decades prior to Gorbachev reflect an on-going dissatisfaction with the level of technological innovation and implementation in the Soviet economy, but a considerable lack of agreement among policy-makers about how to best address the issues. Many measures have been experimented with: a variety of *khozraschet* mechanisms, funding systems, organizational structures at all levels, and economic indicators. The reforms reflected the tension between efforts to centralize and decentralize science. The underlying desires to tighten the links between entities in the R&D chain, improve the responsiveness of R&D facili-

ties to the users of innovations, and in general improve the level of applied and fundamental research were addressed alternatively (and often concurrently) by, on the one hand, efforts to enhance low-level responsibility, incentives, and flexibility and, on the other, attempts to add higher level structure, control, and coherence to science on the branch and national scale.

In part, however, the reforms reflect the sometimes conflicting needs and goals of the conduct of science. On the one hand, it was felt that reducing centralized control would improve initative, responsibility, and responsiveness to real-world needs at the lowest levels. On the other hand, conducting scientific projects which are large in scope or effect, which involve the participation of many different organizations as suppliers, executors, or users was perceived as requiring a comparable centralized management structure. The criticism of ‘‘*melkotem'ye*’’ (excessive concentration on minor tasks) has been frequently used as a justification for greater centralized funding and control even though small, short-term projects filled needs of both developers and customers [Fort90, 4,92; Lakh90, 147-148].

At all times Soviet science was conducted within the general framework of the centralized, directive system of economic management. The five-year and annual plans continued to provide the blueprints for institute activities and the material-technical supplies needed. A set of state standards continued to dictate the steps to be taken in a research project. One set of standards, formalized in 1973 as the Unified System of Design Documentation (YeSKD), specified the stages of work and the documents to be filed and approved at each stage. Other standards specified the formulas for calculating a variety of indicators, the cost of development and manufacture, etc.

2.1.4 The *Perestroika* Reforms

While the results of any particular reform package are difficult to measure, there is little question that by the time Gorbachev was appointed General Secretary in 1985, the hoped-for improvement in technological innovation and implementation had not been achieved. At the same time, advances in science and technology and their broad application were seen as the cornerstone to a revitalization of the Soviet economy. At the April, 1985 Plenary Session of the Central Committee of the Communist Party Gorbachev stated that

> [t]he task of accelerating growth rates–substantially accelerating them–is completely feasible, if the intensification of the economy and scientific-technical progress are placed at the center of our work, if management, planning, and structural and investment policy are restructured, if organization and discipline are enhanced everywhere, and if the style of activity is fundamentally improved...As a primary strategic lever for the intensification of the national economy and the better utilization of accumulated potential, the Party is bringing to the fore the cardinal acceleration of scientific and technical progress [Gorb85].

To achieve these goals, Gorbachev proposed a multifaceted approach which included both greater centralization and decentralization, increased reliance on economic mechanisms, and the two-pronged advance of both applied and fundamental research. At a conference on questions of the acceleration of scientific and technical progress in June, 1985, he stated that

> [w]e should advance along lines of the further strengthening and development of democratic centralism. Increasing the efficiency of the centralization principle in management and planning, expanding the independence and responsibility of enterprises, making vigorous use of more flexible forms and methods of management, economic accountability and commodity-money relations, broadly developing

> the initiative of the masses–this is the fundamental essence of restructuring [Gorb85b].

He called for an expansion of the role of the Academy of Sciences in both fundamental and applied research. He strongly criticized branch science for the low qualitative and quantitative level of its research, and called for greatly increased efforts to include R&D institutes into associations and enterprises, reducing the ‘‘isolation’’ of institutes and design bureaus from production (by March, 1987 the number of NPOs had reached 376; by the end of 1990, 500 [Lakh90, 208]). In addition, he proposed the creation of integrated interbranch scientific and technical centers to coordinate work in key technical areas. In searching for a mechanism which would spur collectives to achieve success in accelerating scientific and technical progress, he proposed a set of measures which would strengthen the customers' influence on the technical level and quality of output. These included a shift to full *khozraschet* with profit as the primary indicator, a decrease in the number of planned assignments and a reduction in the number of plan indicators. Underlying the measures was the notion of ‘‘intensification’’–that the measures to accelerate scientific and technical progress should pay for themselves [Gorb85b]. In other words, in contrast to past industrialization campaigns, progress would be made not through extensive increases in funding or allocations, but through a more efficient use of the resources already available.

During the following years, many of the ideas proposed by Gorbachev were encoded in law and policy. Among the *perestroika* reforms most strongly affecting science were:

- February, 1986 decisions made at the 27th Party Congress to enhance the role of the Academy of Sciences as a coordinatator of scientific research within the country, strengthening its responsibility for the creation of the theoretical foundations of fundamentally new forms of technology [Lakh90, 46].

- the creation of inter-branch scientific-technical complexes (MNTK) starting in 1986.

The MNTK were intended to overcome the departmentalism between branches of industry and academia and the resulting gap between research and production. They were headed by a major research institute, often from the Academy of Sciences, and contained a number of other institutes, design bureaus, prototype development facilities, and production enterprises, drawn from a variety of ministries. The MNTK have not generally been perceived as successful, suffering from insufficient real authority to compete with the ministries from which the constituent organizations were drawn, weak support from the Academy of Sciences, etc. [Fort90, 118-123; Lakh90, 210-214].

- March, 1987 decree ‘‘On raising the role of VUZ science and the acceleration of scientific-technical progress, improving the quality of training of specialists.’’

Among other things, this decree called for a doubling or trebling of the amount of exploratory and theoretical research done in the VUZy by the end of the 12th Five Year Plan (1990). This was to be accompanied by some additional budget appropriations, but also through an increase in the amount of contract-based funds [Fort90, 92-93]. The intent of the decree was to involve the VUZy to a greater degree in large-scale scientific and scientific-technical problems [Lakh90, 80].

- A decree passed at this time expanded the authority of Academy of Sciences departments, giving them powers previously held only by the Presidium.

These included the right to manage all material and financial resources and distribute them among subordinate institutes, authorize the latter’s plans, establish

international contacts, and develop and authorize the basic directions of fundamental research [Lakh90, 44].

- June 1987 resolution ‘‘On state enterprises (associations)’’ [Prav870701].

This far reaching resolution, more than any other, altered the conduct of science within scientific-production associations. To a considerable degree, this was the resolution which gave the proposals advanced by Gorbachev in 1985 and confirmed at the 27th Party Congress in 1986 a legal foundation. In general, it

> ‘‘deepens the principle of centralization in the accomplishing of highly important tasks of the development of the national economy as a whole, provides for the strengthening of economic methods of management, the use of full economic accountability and self-financing, the expansion of democratic principles and the development of self-management and defines the relationship between enterprises (associations) and bodies of state power and management.’’

The resolution established full *khozraschet* as the basis for economic activity of enterprises and associations, making the income earned through contracts the primary source of funding for the various technological development, wage, and social funds. Full *khozraschet* was to penetrate to the lowest levels, that of laboratories, shops, etc. The enterprises (associations) were given considerable rights to keep this income and use it at their discretion. The resolution gave them the responsibility of formulating their own plans and contracts (albeit ‘‘[g]uided by control figures, state orders, long-term scientifically substantiated normatives and ceilings, as well as consumers’ orders...,’’ effectively giving the centralized authorities considerable power in determining the R&D and production schedules.) Enterprises and associations were given the right to give, sell or trade materials, equipment, buildings to each other and establish contracts with each other. The resolution also gave the enterprises and associations the right to determine their internal structure and elect their own executives. Large-scale reorganizations were still sub-

ject to the approval or higher-level agencies. Enterprises and associations were given the right to engage directly in joint projects, joint ventures, and commercial activity with CMEA and Western organizations, although these were subject to approval by central authorities.

The resolution did not free prices, or take off all restrictions on wages. Prices on the results of research and development could be negotiated between the provider and the customer, however. The resolution also made it clear that orders from the state had to be fulfilled before contracts from individual enterprises or associations could be satisfied.

In August, 1989, this law was amended to grant structural units and autonomous enterprises belonging to an association the right, by a decision of their labor collectives, to withdraw from the association, provided they observe the contractual procedure and obligations established during the formation of the association. Enterprises and associations could also the right to withdraw from their ministry [Izv890811].

- September, 1987 decree ‘‘On transferring scientific organizations to full *khozraschet* and self-financing.’’

Embodying the notion that a research product is something which can be bought and sold like a commodity, this decree stipulated that scientific organizations were to earn enough through contracts with users to cover their development costs and the associated institutional overhead. Technically, organizations which failed to do so could ‘‘cease to operate.’’

- 1988, introduction of competition-based financing of projects financed by the Academy of Sciences and the GKNT.

Much financing of institutes was replaced with directed funding of specific projects, selected from competing proposals. Projects were to be selected on the basis of recommendations of councils of experts drawn from the Academy of Sciences, the GKNT, and

branch industry. In 1988-1989, nearly a third of the resources allocated to institutes in the Academy of Sciences' Department of Informatics, Computer Technology, and Automation (OIVTA) was distributed on a competitive basis [Nemo88; Veli89, 22-23].

- 1989,1990 Decrees on small enterprises.

The ''Statute on the organization of the activity of small enterprises,'' approved in June, 1989 by the USSR Council of Ministers Commission for the Improvement of the Economic Mechanism, allowed large state organizations to create small enterprises from individual shopes, subdivisions, and separable production units [Krol90, 56]. Many problems, including property ownership, supplies of material and equipment, and hiring practices remained unresolved and establishing small enterprises was difficult. Matters were improved somewhat by the statutes ''On measures for creating and developing small enterprises'' and ''On the general foundations of the activity of small enterprises,'' of August 8, 1990 [Ezh9008; Ezh9008b]. Small enterprises in scientific spheres were limited to 100 employees.

The efforts since 1985 to reform Soviet science to a great extent incorporated ideas which were not new. Efforts to decentralize parts of the centralized system of management had been tried to various degrees during the 1960s. The *khozraschet* mechanism had, in various forms, been experimented with for decades. Production and scientific-production associations were products of the 1960s. The program-goal method of planning and the 15- and 20-year Complex programs were continued, most prominently in the Comprehensive Program for Scientific and Technical Progress of the CMEA Countries to the Year 2000 (Program to the year 2000). The introduction of state orders preserved the essential features of centralized directive planning, in spite of its ostensibly being fundamentally an economic mechanism. The previous reform efforts and the discussion sur-

rounding them had produced a considerable pool of ideas and experience on which Gorbachev could draw.

What was new was the extent to which some of these measures were applied, and their combination with a number of measures which had not been tried before. Full *khozraschet* with its emphasis on profit as the primary indicator was implemented to a much greater extent than any time since the New Economic Policy of the early 1920s, extending to Academy of Sciences and VUZ institutes as well as branch institutes and enterprises, and emphasizing the penentration of this mechanism to sub-institute levels.

The measures promoting ‘‘small business’’ also opened the door for much greater de-centralization of scientific and economic activity than had been possible before.

The question we seek to answer is how R&D institutes within one particular sector–high performance computing–fared under the various reforms over the years, but under the *perestroika* reforms in particular. Most analyses of Soviet science focus on science as a whole. Issues of technological innovation have been primarily addressed at the aggregate level: the overall level of technology, the number of innovations in the various branches of industry, the economic effect of innovation, etc. Few have investigated individual technologies or specific R&D facilities over time. We seek to examine the results of the reforms mentioned above at these lower level units of analysis. What has changed within the HPC R&D institutes, both in terms of the technical characteristics of R&D projects, the R&D cycles, and the structure of the organizations within which they are carried out? The changes within the Soviet economic, social, and political structures took on a momentum of their own well before the breakup of the Soviet Union which was unanticipated by Gorbachev and his advisors when he assumed leadership of the country, with many negative unintended consequences for the established order. Nevertheless,

can we observe in the response of HPC R&D facilities to the reforms the planting of the seeds of a future, more effective R&D system for the Soviet states ?

2.2 Computing in the Soviet Union

The Soviet Union has a long history of research and development in digital computing. The first Soviet digital, stored-program computer, the MESM (Small Electronic Calculating Machine) was built under the leadership of S. A. Lebedev between 1946 and 1951 in Kiev under very difficult, post-war conditions [Khom89; Crow93]. From 1951-1970, at least 60 known computer models were developed in the USSR, although only a third of these were produced with more than 100 units apiece [Davi78, 95; Rudi70]. In 1950 Lebedev moved to Moscow where he continued developing digital computers at the Institute of Precision Mechanics and Computer Technology (ITMVT) [Bard87]. The machines developed here between 1950 and 1965–the BESM, BESM-2, -3, -4, and -6 –were respectable efforts; the BESM-6 in particular was nearly a world-class machine at the time it was built, both in performance and in levels of innovation. Rudins provides a reasonably comprehensive survey of the earliest Soviet machines in [Rudi70].

In 1967 a program to develop a ‘‘Unified System’’ (ES) of upwardly compatible mainframe computers based on IBM’s System/360, /370 mainframes was announced. In the years that followed, a very extensive international computer industry was built as part of large-scale programs to develop the ES mainframes as well as the Small System (SM) series of minicomputers, based on minicomputers developed by Hewlett-Packard and Digital Equipment Corporation [Davi78; Hamm84]. Goodman and other writers have provided surveys of other aspects of Soviet and CMEA computing such as software, microcomputing, networks, high-performance computing systems, and the ‘‘Soviet-style information society’’ [Good79; Stap85; Mche88; Wolc88; Wolc90; Good87; Good88].

In their survey and analysis of the ES program, Davis and Goodman document the achievements and shortcomings of the program, and firmly establish that Soviet computing cannot be properly understood from a technical perspective alone. A proper understanding of the development and impact of computer systems can only be gained by considering economic, political, and social factors as well. McHenry applies a web model to the study of computer-based information systems within Soviet enterprises [Mche85; Mche86]. The web models, discussed by Kling and others in [Klin82], ‘‘view computing developments as complex social objects which are constrained by their context, infrastructure, and history.’’ McHenry’s work and that mentioned above advance the discussion of the development and use of complex technologies under centralized economic regimes based on an ideology of scientific management.

Our work extends existing research on Soviet computing along several fronts. First, it provides an in-depth survey and analysis of Soviet computing within one sector–high-performance computing–which has not been covered adequately in the literature. While the work of authors mentioned above documented the inferior state of Soviet computers widely used in the civilian sector, the possibility remained that HPC was an exception to this pattern. Our work will fill in this missing piece and determine whether or not this was the case, and why. Second, our work further contributes to the discussion of the development of computer technologies under the pre-*perestroika*, directive economic and political system through a more detailed discussion of the R&D process for such machines. Third, our work examines, at the level of technological systems and organizations, the changes occuring within this sector under the *perestroika* reforms. The writings of Goodman, McHenry, and others have described the difficulties encountered by the computing industry in the pre- and early-*perestroika* years. What has been the effect, both direct and indirect, of the changes in the industry’s context and infrastructure?

Fourth, our work represents a significant change in data gathering techniques and abilities. Before *perestroika,* Western researchers had little access to individuals, institutions, and systems in most branches of science, let alone computing. The high-performance computing sector was particularly closed to Westerners. Research was conducted largely on the basis of extensive reviews of published Soviet literature. While such material was rather extensive and useful when properly viewed, it provided a poor picture of the internal operations of research institutions and R&D teams, and provided little discussion of the development process–the design decisions, the tradeoffs, the factors affecting the development and production of a new system. Information on these had to be inferred from reading between the lines, or from limited personal contacts. Under *perestroika*, we have gained unprecedented access to key individuals and institutions and have been able to complement published literature with interviews and direct observation. Such access has both provided a more robust set of data and made it possible to discuss issues which, in the past, could not have been discussed for lack of data.

2.3 Western Literature on Technological Innovation

2.3.1 Technological Paradigms and Trajectories

Joseph Schumpeter is often credited with being the first to identify innovation as the engine of economic development in capitalist economies [Schu34; Schu39]. Since his pioneering work, a large volume of literature has been written about the sources of invention and innovation (large firms, small firms, private inventors), the kinds of organizational and other factors which are associated with successful choice and carrying out of a project, the relationship between state policy and innovation, etc. [Torn83].

Nelson and Winter [Nels77] and subsequently Dosi [Dosi82; Dosi84; Dosi88b] have tried to capture some of the essence of technological innovation and present it in a more

unified framework. Their work attempts to account for a certain unpredictability in the direction and rate of technological innovation on the one hand, and, on the other, the fact that invention and innovation do not occur *in vaccuo* but are cumulative activities. Technological innovation is uncertain, but the development of a given technology has a certain orderliness to it in which new developments are built on past achievements.

Nelson and Winter introduce the concept of a ‘‘natural trajectory’’ to describe such a path. ‘‘...[I]t may be that there are certain powerful intra-project heuristics that apply when a technology is advanced in a certain direction, and payoffs from advancing in that direction that exist under a wide range of demand conditions. We call these directions ‘natural trajectories’’’ [Nels77, 56]. According to these authors, progress along a natural trajectory is determined by the dynamic of a ‘‘heuristic search process’’ or research strategy that guides developers in their activities, and of a ‘‘selection environment.’’ A selection environment consists of market and non-market forces that serve to identify the more ‘‘worthwhile’’ innovations [Nels77, 62]. These forces, reflecting the values (not necessarily monetary) of organizations and individuals or non-market features such as regulation and policy, favor certain innovations over others. They communicate powerful signals to the innovators about what development paths are worth pursuing. ‘‘Selection criteria’’ are those criteria by which innovations are judged advantageous or not.

Dosi presents the notion of a ‘‘technological paradigm,’’ which he defines as ‘‘an ‘outlook,’ a set of procedures, a definition of the ‘relevant’ problems and specific knowledge related to their solution’’ [Dosi82, 148; Dosi84, 14]. It includes both the set of exemplars–examples of the objects which are to be developed and improved–and the set of heuristics that specify what the next step should be [Dosi88b]. He defines a ‘‘technological trajectory’’ as ‘‘the direction of advance within a technological paradigm’’ [Dosi82, 148]. Like Nelson and Winter’s natural trajectory, Dosi’s technological trajec-

tory embodies the idea that technologies progress along a path which is neither certain or predetermined, nor random. Nelson and Winter wondered if certain technologies exhibited ‘‘natural’’ paths of development, inherent in the technologies, which would be followed under a wide variety of conditions. Dosi’s technological trajectory, on the other hand, places a greater emphasis on the momentum a technology shows once a technological trajectory has been established.

The notions of technological paradigms and trajectories have received considerable support within the technological innovation research community. Nevertheless, a number of issues remain unresolved, which we would like to investigate in some detail through a study of a number of Soviet high-performance computing projects. There is a basic lack of clarity in the definition and scope of the concept of a ‘paradigm’ which, while making it attractive in the abstract, limits its usefulness in practice. The notion of a technological paradigm has its roots in Thomas Kuhn’s work on scientific paradigms which he used to analyze the nature of scientific progress. His major thesis was that scientific progress is strongly influenced by the nature of the ruling ‘paradigm’ [Kuhn70]. As Clark points out, Kuhn had a difficult time defining unambiguously what a ‘paradigm’ is, but the term has become popular and widely used in the sociology of science [Clar87, 28]. This term, and others similar to it, have been used by Dosi and others in relation to technological change, but with no greater precision.

A more serious issue, perhaps, is the lack of clarity about the scope of usage. In Kuhn’s work it is clear that the term refered to a conceptual framework, a set of heuristics and accepted practice that was predominant throughout a scientific community. As used by Dosi, the term certainly refers to something held by a community or an industrial sector, refering to a category of technology (‘‘automobiles,’’ ‘‘tractors,’’ ‘‘high performance computers,’’ etc.). At the same time, it isn’t clear that the term doesn’t also refer to the

‘‘outlook, the set of procedures, the definition of the ‘relevant’ problems and specific knowledge related to their solution’’ held by individual researchers working on specific projects. Certainly the two levels of application are closely related since a paradigm at the community level reflects the beliefs and practicies of individuals and vice versa. Dosi seems to imply that the regularity observed in technological development–the technological trajectory–, even within individual projects, always occurs within the bounds of some technological paradigm which guides development. But is all such regularity the result of a prevailing paradigm? Here the problems of scope of usage become evident. It is conceivable that regularity in development of a specific technological system over multiple generations does not reflect practice which is standard or accepted throughout the community of those addressing similar issues. In other words, micro-level regularities do not necessarily imply macro-level regularities and the use of a single term either implies that the two are tightly coupled, or that the difference is not significant.

A third unresolved issue is the temporal nature of a paradigm, or how paradigms change over time. Little has been said about patterns of evolution of the paradigms themselves. Assuming the existence of one or more paradigms, what can be said about how one paradigm evolves into another, or how one displaces or is absorbed into another? Under what conditions do multiple paradigms coexist over time?

What is clear, however, is that technological trajectories and paradigms rely to a considerable extent on the beliefs held by those involved in the development of a technology and a host of environmental factors which shape these beliefs and form the ‘‘selection environment’’ which provides strong signals to developers about which developments are possible and beneficial. Beliefs, strategies, and environmental factors are important components of the conceptual framework described in section 2.6.3 which is used as a starting-point for this research.

We use a set of case studies in the Soviet high-performance computing sector to explore these issues and to try to add some more precision to the notions of technological trajectories and paradigms. In particular, we wish to find out if, within the bounds of the existing, imprecise definitions, something which can be called a technological paradigm exists within the high-performance computing sector. Second, we seek to identify the patterns of regularity, the technological trajectories, within Soviet HPC development and understand their causes and the factors which influence them. Third, during a period of considerable change at all levels of Soviet society, are there observable shifts in the paradigms or patterns of regularity?

Within the domain of Soviet computing, the high-performance computing sector provides the best opportunities to explore these issues. Implicit in the work of Nelson and Winter, Dosi, and others is the idea that the shape of a technology is unpredictable and the result of an accumulation of decisions made by individuals directly involved in the development of technology. In the case of the majority of Soviet computers, the model for development–computers from IBM or DEC–was largely imported. The task for engineers was not the development of conceptually new technology, but the imitation of an existing technology. This is an instance in which a technological paradigm, if it can be called that, was directly imposed on an industrial sector, and therefore is not well suited for exploring the dynamics underlying the paradigms themselves. In contrast, Soviet high-performance computers were shaped to a much greater degree by research and development done by Soviet researchers and engineers themselves.

2.3.2 The Innovation Process

The work of Nelson and Winter, Dosi, and others focuses on the patterns of technological change over time but speaks relatively little about how individual acts of innovation take place and the factors–organizational, technical, environmental–which promote

it. Other authors have considered such matters at length and have proposed an impressive number of variables which affect the degree to which product innovation takes place within an organization. These have included organizational size, education and skills of innovators, the available pool of theoretical and practical knowledge, degree of centralization of authority, the resources devoted to research and development, and many more [Saha81; Rahm89; Roma90]. Kanter has argued that there are four innovation tasks which take place as an innovation process unfolds and the conditions which promote innovation can be best understood through their impact on these tasks. The tasks are a) idea generation, b) coalition building to garner the support necessary to turn the idea into reality, c) idea implementation, or turning the idea into a prototype, and d) the transfer or diffusion of the innovation into practice [Kant88, 173]. Clearly, these tasks are not carried out in a linear fashion, although there is a rough temporal ordering to them.

In her review of the literature and in her own investigations, Kanter has identified a number of factors which facilitate the tasks mentioned above [Kant88, 173-205]. Factors facilitating idea generation include degree of interaction with demanding customers, cross-disciplinary contact with others inside and outside of one's organization, jobs which are defined broadly rather than narrowly, and organizational expectation of innovation. Crucial to coalition building is the presence of backers and supporters, sponsors and friends in high places. The ability to build a coalition also depends on the patterns of resource allocation within an organization, the nature of coalition relationship, open communication patterns which make it easier for individuals to find partners, etc. Idea implementation is facilitated by a certain amount of isolation from administrative interference and distraction, continuity of personnel, flexibility of organization which permit easily changing plans, and a balance between autonomy and accountability. Transfer and diffusion of a technology is helped by an organization or market which is anticipating the in-

novation, by individuals who can serve as a bridge between developers and users, and a receptive institutional environment.

The factors identified by Kanter may or may not constitute a complete or primary set and may or may not reflect the factors at work in the Soviet Union or its successor states. As we evaluate the changes that have taken place within R&D facilities of the Soviet high-performance computing sector we will want to be able to draw some conclusions about whether they are proving helpful or not to the innovation process within these organizations. Our case studies will more precisely identify the key factors at work, but evaluating them in terms of Kanter's four tasks will enable use to come to conclusions about their impact.

2.4 Organizational Development

High-performance computers are developed within an organizational context designed, ideally, to facilitate R&D by providing needed financial and material resources, procedures for decision-making, and overall coordination of activities. Historically, Soviet organizations were given little freedom to determine, independently of higher administrative entities, the organizational structure or how resources are to be used. Precise regulations tightly governed the allocation of wages, the types of positions within the organizational structure, the composition of the directorate, the use of revenue received from the state budget or from contract work, the aquisition or sale of capital equipment between organizations, and many other aspects of organizational life and operation. Under the reforms, legislation regarding organizations changed drastically, giving greater freedom to the organizations themselves to determine their own structure and use of organizational resources. An important part of the study of high-perforamnce computing is the manner in which the R&D facilities are changing and the impact of these changes on the HPC projects, or vice versa.

An enormous body of literature has been written about organizations, encompasing an impressive spectrum of perspectives on what the salient characteristics of organizations are. Gareth Morgan provides an excellent overview of some of the major perspectives, which he calls metaphors, which have been used to understand organizations, identifying the origins, strengths, and limitations of each [Morg86]. He discusses organizations as machines, as organisms, as information processing entities, as cultures, and as political systems. Each provides a different although often complementary view of an organization's structure, the role of an organization's environment, and the nature and forces behind organizational change.

It is clear that the environment for Soviet organizations has changed dramatically since 1985. The organizational perspective which places great emphasis on the interaction between an organization and its environment is the organism, or sysems perspective. Here organizations are viewed as a set of interrelated subsystems which rely on exchanges of resources, products, etc. with their environment for sustenance. The organism metaphor has made valuable contributions to the understanding of organizations. It stresses the importance of understanding the relationship between an organization and its environment as well as the relationship between parts within the organization (the organization's structure). Something of a catch-all for anything lying outside the bounds of an organization, environment is a very broad concept treated in many ways in the literature. Individual theories differ on the specifics, but in general the environment plays the following roles:

- The environment is a source of financial resources, material resources, knowledge, qualified people, and technology. In some cases these resources can be generated interally to an organization (e.g. in-house R&D, training), but ultimately if an organization does not draw in resources from its environment, it

will lose its ability to function at current levels, to change, and to grow. The environment is also a sink which receives an organization's output.

- The environment is a source of regulation. Explicit regulation through legal structures can affect domains of activity, relationships to other entities in the environment, means of carrying out activities, etc.
- The environment is a source of change and uncertainty. The organism organizational metaphor stresses that change and uncertainty can affect an organization's ability to function and that organizations need to be able to adapt to the environment.

The systems perspective has promoted organizational forms which are more fluid and flexible in structure, better able to adapt to changing conditions and pursue innovative ideas. In its purest form, however, this perspective has weaknesses. It assumes that the environment and an organization can be quantified in a manner which allows concrete measurements of the organization's relationship to its environment to be made. There is also the tendency to downplay the fact that organizational subsystems, unlike biological subsystems, have a will of their own and do not necessary work together to form a ''unified whole'' [Morg86, 71-75]. Nevertheless, this perspective provides a useful starting point for studying Soviet organizations in a rapidly changing context.

Growing out of the perspective of organizations as organisms, or systems, contingency theory refers to the large body of literature directed at the relationship between certain features of the organization's environment, and certain characteristics of the organization itself. In its most basic form, contingency theory postulates that the effectiveness or performance of an organization depends on how well an organization's structure ''fits'' the nature of its environment. Contingency theories originally arose as an alternative to classical organizational theory which postulates that there is an ''ideal'' organiza-

tional form. Contingency theories, on the other hand, claim that the ‘‘ideal’’ form of an organization (however that is defined) depends on, or is contingent on, the organization’s environment.

A study by Burns and Stalker [Burn61] is considered an early contingency theory classic. In British and Scottish industrial firms, Burns and Stalker identified two management systems which they labeled mechanistic and organic. In a mechanistic system, corresponding to the machine metaphor, the structure and functions are precisely defined, and exhibit high degrees of functional specialization. In an organic system, however, the structure and functions are much less precisely defined. Burns and Stalker’s conclusion was that mechanistic firms functioned best in stable environmental conditions, while organic firms functioned best when the environment was unstable and changing rapidly or unpredictably. Lawrence and Lorsch extended this research in an examination of the relationship between organizational structure, environment and performance in selected U.S. industries [Lawr67, 151-158; Mile80, 259]. They confirmed parts of the theory established by Burns and Stalker, but added considerable detail to an understanding of what types of organizational structures and processes are best suited to dynamic and complex environments. Other authors have extended the list of organizational elements that must be in alignment to include people and processes [Beer80]; mission, strategy, politics, and culture [Tich83; Smit87; Tosi84]; and technology [Wood65; Perr67] (see [Draz85] for additional references.)

Contingency theories have enjoyed considerable popularity, but support for them in their strongest form has waned. The empirical research has given mixed results. (See [Tosi84] for a review of this research.) There appears to be general support for the idea that organizations and their environments are interdependent, but not strong support for any particular casting of this relationship. Part of the difficulty has been finding useful,

broadly applicable definitions for such concepts as ‘‘structure,’’ ‘‘environment,’’ ‘‘performance,’’ ‘‘fit,’’ or other concepts which have been introduced in various contingency theories. Writings such as [Schr80] question some underlying assumptions such as the idea that there is a single best structural ‘answer’ to a specific contextual situation, and that organizations have no influence in shaping their environment. Additional complications arise from the fact that the use of many different units of analysis make cross study comparisons difficult.

Several authors have investigated the relationship between organizations’ structure and the technologies they develop. Scott provides a review of this category of contingency theory [Scot90]. Much of the work has been devoted to testing postulates such as:

- the greater the complexity of the technology developed within an organization, the greater the organization’s structural complexity in terms of differentiation of function, level or location;
- the greater the certainty about what problems or procedures are to be encountered the lower the formalization and centralization of decision-making;
- the greater the extent to which the items and elements involved or the work processes are interrelated, the more resources must be devoted to coordination mechanisms.

He concludes that [Scot90, 117]:

> given the many types of problems cited (multiple conceptions, variables, indicators, samples), the results of empirical studies conducted up to the present provide evidence for linkages among technology and organizational structure. The evidence is reasonably consistent but not particularly strong.

Scott proposes a number of measures to improve the arguments and strengthen the evidence for contingency theories. These include narrowing the focus to the portions of

an organization which is concerned with a single, or dominant technology; examining a looser coupling of structure and technology which does not imply a single, deterministic relationship but allow for the possibility of multiple suitable structure-technology relationships; and recognizing that organizations are affected by many factors in their environment (‘‘wider rules and regulations, belief systems, and legal frameworks that surround, support, and constrain organizational forms’’) which operate to reduce or constrain technological effects. Scott also recommends an increased temporal emphasis, noting that the relationship between structure and technology is not static, but is better viewed as an interdependence evolving over time.

There are a number of implications for the study of Soviet high-performance computers and their associated R&D facilities. First, Western writing on contingency theories for the most part is based on an assumption that until recently did not hold in the Soviet Union–that individuals and groups within an organization have the authority to modify the organization’s structure. Prior to *perestroika*, principal decisions about the structure of an organization such as an enterprise, a factory, or a research institute were made, or at least authorized, outside of the organization. Proposals for the creation of an organization or the modification of its structure and function had to be approved at higher levels within the ministry, or the Academy of Sciences, often at the level of the Council of Ministers. The formal structure of a given category of organization, such as research institutes, was very similar throughout the Soviet Union. This can be viewed as an extreme case of environmental influence on the structure of the organization, but this degree of determinism is not supposed by even the most ardent contingency theory determinists in the West.

One of the changes under *perestroika* has been the greater delegation of decisions regarding organizational structure to the organizations themselves. The influence of (cer-

tain components of) the environment has become less direct. The Soviet condition to a greater degree satisfied a basic assumption of contingency theories. It has therefore become more appropriate to investigate the environment-organization relationships of Soviet R&D facilities since 1985 in light of some of the research on contingency theory in the West.

Given the lack of a clear and unified set of results in the Western contingency theory literature, a second implication for the study of Soviet high-performance computer R&D facilities is that a narrow investigation of linkages between a small, preselected set of variables would probably be misguided. Following some of Scott's recomendations, we focus our study on divisions within R&D facilities where a single technology is dominant. We will also examine the technologies and organizational structures within a broad context, recognizing that the relationships between technologies, organizational structure, and the organizational environment can be loosely coupled and evolving over time, and that there can be a multitude of factors which influence them simultaneously.
While in this study we have selected a limited number of dependent variables–high-performance computing systems and the structure of their R&D facilities–we leave the set of independent variables rather loosely bounded and defined.

2.5 Research Questions

To address the issues raised thus far, we focus on four primary research questions:

- How have high-performance computing systems developed within the Soviet Union and since its breakup? Which factors best explain the direction and nature of their evolution?
- How have Soviet high-performance computing research and development facilities and the computing systems they developed changed since the start of

the *perestroika* reforms in 1985? We will not limit ourselves to direct effects of the reforms, since changes in laws and policy have secondary or tertiary effects which can be as important as those originally intended. The study will examine the evolution of HPC technologies, changes within HPC R&D organizations, and contributing factors in the broader political, technical, social, and economic environments as they influence developments within the HPC sector.

- What conclusions can be drawn about the nature of Soviet R&D, Soviet ability to develop and produce advanced technologies, and the viability of HPC R&D in the domestic and international context? Is there evidence that the changes are laying a foundation for more effective R&D and greater contribution and participation in global developments in science and technology in the future?
- How well do Western theories and models about technological change and organizational development help us understand the changes in progress in Soviet high-performance computing? What are the strengths/weaknesses of the theories and models in the context of Soviet HPC? In what ways must they be modified to explain changes in this sector properly?

2.6 Research Methodology

In this section we outline the multiple case study with embedded units of analysis methodology used to seek answers to the questions presented in section 2.5. In addition to specifying the units of analysis and constituent cases, we discuss the case study methodology in light of the theoretical perspective which we feel should be used in this study. As we will explain, theories can differ not only in their content–the specific variables used and the manner in which they are related–but also in their assumptions about what

Markus and Robey [Mark88] term the causal structure, or the fundamental assumptions about the nature and direction of causality a theoretical framework claims to represent. It is important that the methodology be well adapted to the causal structure of the theoretical perspective.

2.6.1 Units of Analysis

We will focus on change in two areas: computing systems developed within Soviet high-performance computing R&D facilities and the structure of these facilities.

The hardware and software systems developed at Soviet R&D facilities represent the end result of an intricate interplay between technological, economic, social and political factors. While this interplay is quite complex, we are interested in the factors which play the most significant roles in influencing the nature of the final outcome, the computing system. We analyze the physical, logical, performance, and, to the degree possible, operational characteristics of specific systems, identifying how these characteristics have changed over time in either the same models or a succession of different models.

As researchers like Dosi have pointed out, technology is more than simply physical artifacts. Dosi defines technology as ‘‘a set of pieces of knowledge, both directly ‘practical’ (related to concrete problems and devices) and ‘theoretical’ (but practically applicable although not necessarily already applied), know-how, methods, procedures, experiences of successes and failures and also, of course, physical devices and equipment...’’ [Dosi82, 151]. In our study of Soviet HPC, we will consider the non-physical parts of technology, but only to the degree that they are reflected in the physical. We will focus on the physical systems and the construction, designs, and architectures, the means of development and manufacturing, the underlying concepts and design principles, and the knowledge embedded in them rather than abstract knowledge possessed by a developer but not necessarily applied.

Our second unit of analysis is the structure of R&D facilities within the Soviet high-performance computing sector. As has been discussed, there are many different perspectives on what an organization is, and what constitutes its structure. The organizational unit we are most concerned with is the division. An institute may have many divisions carrying out many kinds of research in many fields, some HPC and others not. Within a division, however, work generally is unified around a single product, or family of closely related products. We agree with Scott that [Scot90, 119]:

> [t]he difficulty [in analyzing the relationship between technology and organizational structure], of course, is that such averaging approaches ignore the realities of both technological diversity and structural differentiation–the coexistence within organizations of subunits with varying technologies and structures. For this reason, it is my belief that technology-structure arguments are best suited to the subgroup (for example, departmental) level. Most departmental units can be characterized as containing one or a least a dominant technology.

Hage also supports this perspective. In his view, an organization is oriented around a rather specific set of processes, knowledge, and technologies [Hage80, 10]. In his definition, large companies might consist of separate organizations–divisions–which produce product lines on a rather autonomous basis. Such divisions often differ in the processes they use, the knowledge they apply, and the technologies they utilize. They deserve to be examined separately.

Although organizational structure has been defined and measured in many ways in the literature, there is, as Ford and Slocum point out, some agreement that three principal dimensions are complexity, formalization, and centralization [Ford77, 562]. Complexity refers to the extent of differentiation in a system, be it vertical (the number of hierarchical levels), horizontal (the number of functions, departments or jobs), or geographic. Formalization refers to the extent to which rules and procedures are specified and/or adhered

to. Centralization is defined as the locus of formal or informal control within a system. A fourth dimention, prevalent but not as widespread, is administrative intensity which refers to the size of the administrative component relative to the direct labor component.

2.6.2 Rationale for Using the Case Study Methodology

We employ a multiple-case case study with embedded units of analysis. Case studies are appropriate for studying ‘‘how’’ questions focusing on contemporary events in which researchers have no control over the research variables [Yin89]. We are interested in the manner in which computer systems and R&D facilities within the high-performance computing sector have changed, rather than in a quantitative assessment of such change. Case studies are also appropriate when the number of variables is large, and their relationship to each other is complex and/or poorly understood. We focus on a number of variables which we feel have the greatest explanatory power, but other variables which emerge as significant during the investigation will also be taken into account.

The case study is appropriate given the nature of the theoretical structure which should be used for this kind of study. Markus and Robey have written an insightful article on causal structure in theory and research [Mark88]. While their article specifically focuses on theories about the relationship between information technologies used in organizations and organizational structure (we focus on technologies which are created rather than used in organizations), their argument is applicable to our study.

Rather than analyze theories in terms of their content, Markus and Robey examine theories in terms of their structures–the theorists’ assumptions about the nature and direction of causal influence. Specifically, they examine three dimensions of causal structure: causal agency, logical structure, and level of analysis.

They argue that there are three conceptions of causal agency (defined as the analyst's beliefs about the identity of the causal agent, the nature of causal action and the direction of causal influence among the elements in a theory) which they term the technological imperative, the organizational imperative, and the emergent perspective [Mark88, 585-589]. In the technological imperative perspective, information technology is viewed as a cause of organizational change; in the organizational imperative perspective, information technology is a dependent variable and a great deal of choice, control and ability to select information technologies and shape organizational structure is assumed on the part of individuals within the organization. In the emergent perspective, organizational change emerges through the interaction between the information technologies and users in a manner which is not very predictable. In other words, there are no clear and simple causal relationships.

The emergent perspective is most appropriate in our study for several reasons. Redirecting Markus and Robey's focus on information technology as a tool used within an organization to information technology as a product created within an organization, we would interpret the technological imperative as claiming that the nature of the technology being developed determines organizational structure and the organizational imperative as giving entities within an organization great freedom in determing the direction of technology development. There are several reasons why these perspectives are too limiting for our study. The technological imperative assumes a high level of rationality on the part of decision-makers throughout the organization, that their decisions are based on objective analyses of the characteristics of the technology and the organizational structures best suited for its development. It leaves little room for decisions based on factors other than the tools' objective characteristics, such as any social meaning the technology may embody, or factors external to the organization which influence development. The organiza-

tional imperative gives decision-makers considerable degrees of freedom to determine the course of development without placing much emphasis on constraints inherent in the technology, and environmental factors beyond the control of decision makers which enable or constrain development.

In general, but especially during a time of rapid change in the Soviet political, economic, and social arenas, the likelihood is high that the development of technology and changes within organizational structure are best described not by simple, unidirectional causal relationships, but by a complex interplay of many factors. The emergent perspective best enables us to capture this complexity. The case study, with its attention to the richness and complexity of actual situations is well suited to investigations from the emergent perspective.

A second dimension of causal structure is logical structure, the logical formulation of the theoretical argument. Markus and Robey, following other researchers, distinguish between variance and process theories. Variance theories are generally rather static in nature, seeking to predict levels of dependent variables from the values of independent variables. In other words, they assume that the outcome will invariably occur when necessary and sufficient conditions are present. As stated in [Mark88, 590], ‘‘If X, then Y; If more X; then more Y.’’ In contrast, process theories hold that certain conditions may be necessary for an outcome to occur, but their presence does not necessarily imply that the outcome will occur.

A knowledge of process is of great importance. Quoting L. B. Mohr, Markus and Robey state that ‘‘necessary conditions can comprise a satisfactory causal explanation when they are combined in a ‘recipe that strings them together in such a way as to tell the story of how [the outcome] occurs whenever it does occur’’’ [Mark88, 590]. In his study of organizational metaphors, Morgan also proposes ‘‘the most effective story line’’ as an

analysis tool well suited to capturing the richness of change over time [Morg86, 329-337]. This strategy allows us to draw on multiple perspectives on organizations and technology to find the best explanations for observed events and situations, and link these insights together into an integrated picture which not only describes what has transpired and why, but can provide a foundation for prescriptive or prognostic analyses of what some possible avenues of change are, or how matters may advance in the future.

A knowledge of the temporal relationships between elements in a setting is crucial to understanding final outcomes. Process theories have the advantage of greater empirical fidelity, albeit at the cost, potentially, of generalizability. Markus and Robey are careful to point out that they do not give up all claims to generalizability, however: ‘‘By accepting a more limited definition of prediction, one in which the analyst is able to say only that the outcome is likely (but not certain) under some conditions and unlikely under others, process theorists may be able to accumulate and consolidate findings about the relationship between information technology and organizational change’’ [Mark88, 593].

The temporal component is crucial to our study. In seeking to accurately describe the processes and consequences of change in a wide range of factors in Soviet high-performance R&D facilities and technologies, it is crucial to investigate how the factors change and interact over time. Reform in the Soviet Union consists not of an isolated event but of a series of changes in political, legal, social, economic, technological spheres that build on each other over time. While it is theoretically possible to study this phenomenon using variance theories, such an approach is likely to miss a great deal of the richness of change and give results which, while ostensibly more generalizable, reflect actual circumstances only weakly.

The case study is an appropriate methodology for use with process theories. It captures the richness of the process of change and enables, indeed encourages, a longitudinal

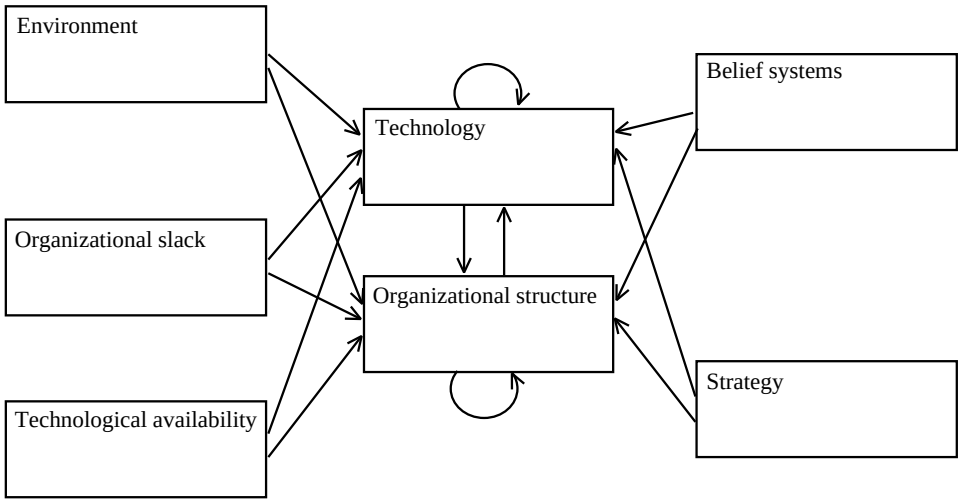


Figure 2-1 Conceptual Framework

approach, tracing change and its impact over time. It permits the capture of the dynamics of the interplay between factors as they evolve and change, something difficult to do using other methodologies.

2.6.3 Conceptual Framework

The conceptual framework shown in figure 2-1 seeks to integrate the most salient factors, or groups of factors, influencing the development of Soviet high-performance computing systems and their associated R&D facilities. It recognizes the considerable overlap between the factors that play a role in both organizational and technological development. Many of the elements in this framework have already been discussed. Here we reiterate some of this discussion, and discuss the remaining factors and some features of the framework as a whole.

2.6.3.1 Technology and Organizational Structure

The two units of analysis, the technological system and organizational structure, have been discussed in section 2.6.1. One characteristic which they share is momentum. In other words, their future state is strongly influenced by the current state. Once established, organizational structures can be difficult to change. When they do change, it is rare that they change drastically in one step. Rather, they evolve so that the structure tomorrow represents an incremental change in today's structure. Because it is generally more difficult and more expensive in terms of time, money, and energy to change a structure drastically, today's structure puts practical constraints on how much a structure can be changed, and how quickly.

A characteristic of technological change pointed out by [Nels77; Dosi82; Saha81b; Saha85] and others is that most technological innovations exhibit an incremental nature. Today's technology is generally an extention of yesterday's, rather than a radical departure. Gradual improvement is a dominant characteristic. A significant amount of innovation and improvement originates through 'learning-by-doing' and 'learning-by-using' [Saha85; Dosi88b]. Dosi also points out that it is well established that

> 1) in spite of significant variations with regard to specific innovations, it seems that the directions of technical change are often defined by the state-of-the-art of the technologies already in use; 2) quite often, it is the nature of the technologies themselves that determines the range within which products and processes can adjust to changing economic conditions; 3) it is generally the case that the probability of making technological advances in firms, organizations, and often countries is, among other things, a function of the technological levels already achieved by them [Dosi88b, 223].

Thus the technology which exists today has a considerable impact on the technology developed tomorrow, both suggesting and constraining new avenues of development. For

example, the need to maintain software compatibility from one computer generation to another constrains the path of hardware development. As engineers work on a particular design, their knowledge about a particular approach increases. They are more likely to pursue development paths which build on their expertise than move in directions in which they have limited knowledge.

High-performance computing systems demonstrate another way in which a technology influences the future development of the same technology. Computer-aided design and simulation systems running on current computer models are often used to develop the next generation systems. The complexity or thoroughness of the modeling depends on the capabilities of the software systems and the hardware on which they run.

2.6.3.2 Environment

The importance of the environment–that which lies outside the boundaries of an organization has been identified as important in both the technological innovation and organizational development literatures. It provides many of the resources and signals to developers which constitute the ‘‘selection criteria’’ shaping researchers’ decision-making. The construction of advanced technologies and an organization’s vitality depend on its ability to acquire the financial, material, and political support necessary to carry out its activities over the long term. As we have seen, studies on innovation in the Soviet Union and on Soviet computing have also placed emphasis on the role of the context within which innovations take place. The projects which we are studying all have a strong applied nature; the development goals include the construction of physical machines which are oriented to one degree or another to use beyond the boundaries of the R&D facility.

In our case study we view the environment not as a single entity which can be represented by a single metric or label, but as a set of factors, each having its own influence on our units of analysis. These include, but are not limited to, project sponsors and funding

organizations, customers, upstream suppliers, series production factories, the nature of the Soviet economic system (the economic macrostructure), legislation, and foreign export control policies. Two components of our conceptual framework, technological availability and organizational slack, contain elements which could easily be considered part of the environment. We single them out because of their importance to our understanding of developments in Soviet HPC and because they are not confined to the world external to an organization.

2.6.3.3 Technological Availability

Technological availability refers to the presence and accessibility of technology needed to carry out some desired line of development. Technology here refers to physical artifacts such as components, tools, and examples of promising developments (called technological guideposts by Sahal [Saha85]), processes, and technical know-how. The presence or absence of key technologies has a strong, direct impact on other technologies, impacting the perceptions of what is feasible, the development timetable, and the quality and characteristics of the final product.

Although the availability of specific technologies is not usually a sufficient condition for the development of a particular high-performance computer, it is often a necessary one. The lack of certain technologies forces efforts to compensate through design modification, the development of substitute technologies, or greater efforts to acquire technology by other means. Such efforts have strong impacts on the nature of the end product and the time needed to develop it.

Aspects of technological availability which have implications for organizational structure are the presence of know-how and examples of successful organizational structures. While in general certain types of technologies (i.e. telecommunications) can support certain organizational structures (i.e. geographically distributed), in our study know-

how and examples of success play a more significant role. Organizational leadership may choose to pursue a particular form of change based on their own experience, or examples of changes which they observe in other organizations.

2.6.3.4 Organizational Slack

Organizational slack refers to the amount of resources available to an organization [Meye82]. It can include financial and material resources, knowledge, people, or even items of barter value such as bricks. Clearly these resources are necessary to sustain the activities of an organization. The degree to which these resources are slack–available in quantities exceeding that needed for the organization to maintain is core activities–plays an important role when organizations undergo change. While an abundance of resources can allow an organization to ignore or respond slowly to problems, transformation can be greatly hindered when there is a lack [Hage80, 274-276]. Efforts such as retooling may not be carried out when there are inadequate resources constraining the options for change.

Organizational slack does not refer solely to resources contained within the organization. An organization may have a considerable in-house reserves of capital, equipment, and know-how. Its ability to go outside and acquire such elements as lines of credit or ready supplies of inputs is also part of its slack.

Two important components of organizational slack are the quantity of resources available, and the degree of discretion which individuals within an organization have over them. Both components have changed significantly in the Soviet Union in recent years. Reforms in legislation have given individuals within organizations throughout the economy much greater freedom to manage their own finances, to make decisions about how finances are to be allocated, and resources accumulated. Changes in the quantity of resources available is a secondary effect of *perestroika* resulting from a number of factors

including changing accounting practices, inflation, decreases in the military budget, and the general poor state of the economy.

2.6.3.5 Belief Systems

The theory of technological paradigms places great importance on the set of beliefs about ‘‘the problems that must be tackled, the tasks to be fulfilled, a pattern of inquiry, the material technology to be used, and the types of basic artifacts to be developed and improved’’ [Dosi88]. Individual projects are strongly driven by design philosophies which govern how design objectives are to be prioritized and met, and how tradeoffs are to be resolved. Changes in key components of a design philosophy can result in significant shifts in project design and construction. Under conditions of such dramatic change such as in the Soviet states, it would not be surprising to observe changes in the philosophies behind HPC development, which would manifest themselves in changes to the technologies.

Belief systems play an important role in the organizational development literature as well, particularly in regard to organizational transformation. The cultural metaphor in particular emphasises on the role of the ‘‘world-view’’ of individuals and groups within an organization, but the organization as a system metaphor also recognizes the role of the beliefs of key decision-makers shaping strategy and directing decisions about the organization’s structure, domain of activity, mission, etc. [Gree88; Hini88]. It is also possible, although by no means certain, that a changing environment can result in a shift in basic beliefs about an organization’s appropriate sphere of activity and principles of organizing, paving the way for alterations in both R&D programs and structure.

2.6.3.6 Strategy

In a study like ours, which emphasises the temporal aspects of technological and organizational development, strategy is a one element which helps us understand the courses of action taken by individuals or groups of individuals. It is a vehicle by which elements of a belief system are translated into concrete changes to technology or organizations, opportunities and constraints of the environment are reflected in the characteristics of the HPC system or the organization.

The word strategy is used in a number of different senses in the literature and, as Chaffee points out, often with little concensus. Mintzberg has isolated five definitions: strategy as a plan, pattern, position, perspective, and ploy [Mint88, 13-30]. As a plan, strategy is viewed as a preconceived course of action. Within the organization as a system perspective, strategy is more likely to be viewed as a pattern which emerges through the actions of individuals or organizations. In constrast to strategy as a plan, which focuses on intent, this definition examines actions which actually occur. As a position, strategy reflects the selection of a particular domain, or niche within an environment. As a perspective, strategy is a way of perceiving the world and one's place in it rather than a chosen position or action. It is integrally linked with belief systems. Most often found within political metaphors, strategy as a ploy involves the creation of a diversion, in order to attain other, higher-level goals.

These views of strategy can complement each other. Each highlights different elements which help us understand an organization's course of action and the path along which its technology develops. Most relevant to our study are strategy as a plan (intended strategy) and a pattern (realized strategy). Together, they help us understand what courses of action were intended, and which actually occured. A comparison of these can yield useful insights into the reality of technological development in the Soviet context.

Strategy as a ploy does not play a large role in our case studies. The two other types of strategy–strategy as a perspective and as a position–are, to a degree, incorporated under the heading of ‘‘belief systems.’’ They both reflect a deeper understanding of the nature of the organization, its role, and the appropriate way of conducting its business.

2.6.4 Constituent Cases

Our study is limited to HPC projects which resulted in the construction of at least a prototype and their successors. We do not discuss the dozens of projects which progressed no further than paper design or abstract theoretical conceptions. The latter are too numerous to be considered in depth, contribute little to our understanding of the development of advanced technologies within the broader socio-economic context, and have had little or no impact on the computing needs of the country.

Because the Soviet HPC sector consists of a modest number of projects, organizations, and individuals, we are able to discuss nearly all R&D facilities and projects which satisfy the above criterion. The core of the study will be in-depth case studies of selected lines of development at three R&D facilities complemented with a less thorough treatment of approximately a dozen others. The core studies will examine developments at the following institutes. The computer systems of interest developed at each are listed in hard brackets:

1. Institute of Precision Mechanics and Computer Technology (ITMVT), Moscow [El’brus family]
2. Scientific Research Institute of Control Computing Machines (NIIUVM), part of the Impul’s Scientific Production Association, Severodonetsk [PS-2x00]

3. Computer Center of the Siberian Department of the USSR Academy of Sciences (VTs SO AN SSSR)/Institute of Informatics Systems (ISI) Novosibirsk [MARS-M, MARS-T]

These organizations and their projects form a cross-section of the sector as a whole, and are useful for illustrating the key features of Soviet HPC R&D. They constitute two industrial R&D facilities (with ITMVT having dual subordination to the Ministry of the Radio Industry (Minradioprom) and the Academy of Sciences), and one purely academic facility (VTs SO AN SSSR/ISI). The computer systems developed within them reflect both applied industrial and academic research orientations. The El'brus and PS-2x00 are the two most successful families of Soviet HPC computers in the sense of numbers of units manufactured, and the El'brus line includes the most powerful to reach series production. ITMVT is the leading HPC R&D facility in the country. The machines include both those which push the boundaries of the capabilities of Soviet industry, and those which stay comfortably within its boundaries. These organizations are geographically distributed, lying in Moscow, the Ukraine, and central Siberia,.placing them at various distances, both physically and metaphorically, from the center of power. The organizations range in size, from tens to thousands of workers. Levels and percentages of support which come from the state budget vary significantly. The projects reflect a wide range of research approaches, from architectures reasonably well-researched in the West to radical, unproven experimental designs. The user communities range from non-existent to large (hundreds or thousands of users). In short, factors and trends which affect other organizations on Soviet HPC are likely to impact one or more of these. Such diversity within one sector enables us to observe changes under a variety of conditions, laying a rich foundation for analysis.

Besides those covered in our study, there are undoubtedly secret military projects which we know nothing about. Their absence will not invalidate the conclusions of the study. First, many of the systems discussed in our study were designed primarily for the military sector. By definition, our study leads to some conclusions about computers used in the military. Second, highly secret projects have a direct impact on only small portions of the military. They have little impact on the overall state of Soviet high-performance computing. To have a broader impact they must, necessarily, be publicized and made more generally accessible. Similarly, research results coming from such projects will have a broad impact only if they are incorporated into systems which gain greater visibility and use.

2.7 Data Sources and Collection, Analysis, and Validation

2.7.1 Sources

Much of the detailed information about the HPC systems is found in printed matter: books, open journals, and institute publications. The Mosaic Group at the University of Arizona has gathered such information for many years. While we have obtained much of this material through Soviet publication bookstores and direct subscriptions, some of the most informative material has been given to us during visits to Soviet institutes.

We also rely heavily on conversations with individuals intimately familiar with the HPC projects and institutes. They not only have provided technical information not available through publications, but also are the best sources of information on questions regarding organizational change, technological developments, and the broader development context. We have conducted on the order of a hundred semi-structure and open-ended interviews with individuals who work in, or are knowledgable about, these organizations and their projects.

We have made visited nearly all of the R&D facilities mentioned in our study and had multiple site visits to the institutes in our core studies over a period of several years. In each case we have been able to evaluate directly the state of the technology and the organization at regular intervals since 1989, and have been able to gather detailed information from sources in prior years. Multiple visits provide a set of longitudinal data tracking developments over time. Thanks to electronic mail, on-going communication with many individuals in this sector has become practical.

2.7.2 Analysis

The study analyzes multiple cases and embedded units of analysis. Each case and each unit of analysis must be examined individually, and as a part of a larger whole. In each case, we will seek to develop the ‘‘most effective story line’’ to understand what happened and why. For each project we will identify the principal factors shaping development of the technology and its organizational context. In chapter 8, we will provide a cross-cutting analysis of the individual cases, identifying similarities and differences between the individual cases and coming to broader conclusions about Soviet ability to conduct R&D in advanced technologies, technological innovation, and organizational development.

2.7.3 Validation

To the degree possible, we obtain multiple sources for data items. This includes confirming data using multiple written and oral sources, as well as using the same or different sources at multiple points in time. In the past, this method has not always ensured the validity of the data. Official sources were widely parroted. Under *perestroika,* however, individuals have been much more willing and able to speak freely. We have even had occasion to test real machines to get ‘‘live’’ performance data [Wolc91]. Finally, the core

cases have been reviewed by individuals who actively participated in the projects discussed. Other chapters also have been reviewed by individuals more broadly familiarity with Soviet HPC.

CHAPTER 3. AN OVERVIEW OF SOVIET HIGH-PERFORMANCE COMPUTING

3.1 Introduction

Soviet high-performance computing has a long and complex history. To set the stage for subsequent discussion, this chapter provides an overview of this sector from the inception of digital computing in the USSR through the present, highlighting the role of the Institute of Precision Mechanics and Computer Technology (ITMVT) in Moscow. We present the history and chronology of projects which reached at least the prototype stage, or for which prototypes are currently being contructed. This chapter raises a number of issues of technological development and HPC within the Soviet context which will be discussed later, and lays the groundwork for a comparison of a variety of different lines of HPC development.

3.2 HPC Efforts at ITMVT

3.2.1 Early Uniprocessors

The Father of Soviet computing, S. A. Lebedev, began work on the first Soviet digital, stored-memory computer in 1946 at the Institute of Electro-Technology in Kiev. Called the MESM (*malaya elektronnaya schetnaya mashina*-Small Electronic Calculating Machine), it was built under difficult post-War conditions outside of Kiev between 1947 and 1950, when the first programs were run on it. The MESM was formally completed and put into use in 1951 [Liso88, 20-21; Crow93]. Figure 3-1 presents a timeline of early uniprocessors built under Lebedev's direction.

In 1950, Lebedev was invited to come to Moscow to head a new laboratory which had been established for the development of electronic digital computers at the Institute

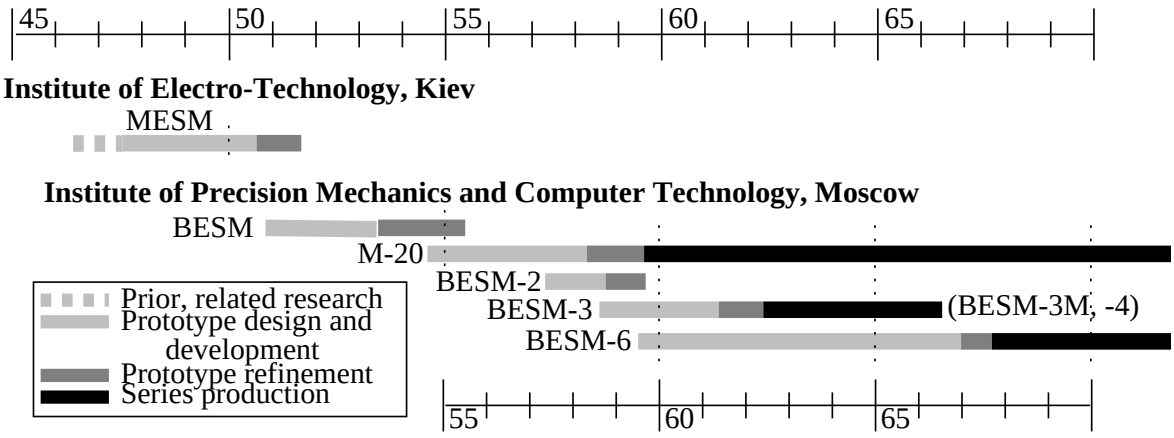


Figure 3-1 Early Soviet High-performance Uniprocessors

of Precision Mechanics and Computer Technology [Bard87; Bard88]. As the head of this laboratory and director of ITMVT (1953-1973) Lebedev established ITMVT as the premier developer of high-performance computing systems in the Soviet Union. ITMVT has maintained that role to this day. Lebedev cultivated a strong tradition of building fast machines with original architectures which were manufactured by domestic industry and used in real-world applications. He built complete systems, and pioneered advances in all aspects of computing, including components, power supplies, construction technologies, peripheral devices, software, and, of course, new architectures. Although software was a necessary part of any system, Lebedev and others felt that the mission of ITMVT was to build the fastest machines possible, even at the expense of software compatibility. The volume of software written for the early machines was small, and he felt that the users would always be ready to re-write existing code. Software compatibility did not become a major design goal until the 1970s, with the El'brus computers described below.

Lebedev began work on the BESM (*bystrodeystvuyushaya elektronnaya schetnaya mashina*-High-Speed Electronic Calculating Machine) in 1950. A three-address, floating-point machine with a 39-bit word length, the BESM was built using vacuum tubes and

mercury delay lines for operational memory. The machine had a main memory capacity of 1024 words and a processing rate of approximately 1,000 operations per second. It was completed in 1952, and accepted by a State commission in 1953 [Crow93].

During these years, Lebedev's group competed with others for equipment and components. In 1951, the USSR Council of Ministers decided to fund two competing projects with the goal of creating a world-class electronic digital computer. One was Lebedev's BESM project; the other, called the Strela, was built in the Ministry of Machine and Instrument Construction in Moscow. The latter had closer ties to industry (and the supply of components) and considerable political influence. This project made it difficult for Lebedev to acquire the quantities of electro-static cathode ray tubes he needed. Although Lebedev did get the CRTs necessary after considerable complaining to the Academy of Sciences and government officials, the experience further convinced him of the need to establish close relationships with industry. Such ties were achieved during the early 1960s when ITMVT and a number of applied research institutes of the Academy of Sciences were transferred to industrial ministries that had equipment and production facilities the Academy lacked [Lakh90, 39-40]. Lebedev promoted close contacts and active involvement by the factories from the earliest stages of research and development as a means of significantly shortening the development cycle [Bard87].

In 1954, Lebedev started a new project to build a computer using germanium semiconductor diodes. In 1955, he began working in close cooperation with a special-design bureau at the Moscow Calculating-Analytic Machines (SAM) Plant in Moscow for the development and series production of such a machine, called the M-20. Some problematic circuits led to strained relations between Lebedev and his SAM counterparts, and Lebedev re-focused his efforts on an an industrial version of the BESM, called the BESM-2. Both the M-20 and the BESM-2 were completed in 1958. The latter had an op-

erational memory of 2048 words and an average processing rate of 7-8,000 operations per second. The M-20 had a 45-bit word length and a nominal performance rate of 20,000 floating-point operations per second.

Lebedev subsequently developed the transistor-based BESM-3 and two slightly upgraded production versions, the BESM-3M and BESM-4. This three-address machine was based more directly on the M-20 than the BESM-2, having a 45-bit word length and using the M-20 instruction set. It had 4-8K words of memory and ran at 20,000 operations per second, like the M-20 [Grub77; Targ80]. The BESM-4 was completed in 1961 and was in series production from 1962-1966 [Gryz88, 93].

In 1959, Lebedev began preliminary work on a machine based entirely on a semiconductor component base. The first mock-up of the BESM-6, Lebedev's most enduring work, was completed in 1964. The BESM-6 was a single-address machine with 48-bit words and a performance of one million operations per second. A full prototype underwent factory testing in 1966, and the machine passed state testing in 1967[1] [Laut91, 11].

The BESM-6 had a profound impact on the Soviet scientific computing community. Between 1966 and approximately 1980 over 350 units were manufactured at the Moscow SAM Plant [Supe91b, 14]. For approximately two decades the machine was the computational workhorse of Soviet science, probably logging more hours of use per unit manufactured than any other serially produced computer in the world. It was beaten in the HPC race only by the 3 MIPS CDC 6600 (1964). Thousands of programmers worked on the

[1] This chapter presents a number of figures indicating machine chronologies. We consider the prototype development period to last from the time a decision was made to build a physical machine to successful completion of ''state testing.'' For state testing, a government commission consisting of representatives of principal users, prominent members of industry and academia, and government policy makers–usually under the chair of Academician A. A. Dorodnitsyn–examined the machine, ran tests, and verified that the prototype conformed to the technical statement of work established at the outset of development. Passing state testing signaled successful completion of the development phase, although did not rule out further modifications to the system.

machine which became the first Soviet computer with a full and rich set of systems and applications software. The machine was very reliable, by Soviet standards, and easily maintained. Over the years, existing systems were upgraded in the field by adding semi-conductor memory and enhanced peripherals. The BESM-6 incorporated many “firsts” for Soviet computing. These included virtual memory, pipeline processing, hardware memory protection, scratch-pad memory, and many others [Laut91, 8-10]. Many of the lead engineers of prominent HPC projects at ITMVT and elsewhere in later years were part of the BESM-6 development team.

3.2.2 ITMVT Computers of the Late 1960s and 1970s

3.2.2.1 BESM-10 and AS-6

Following the completion of the BESM-6, its development team divided to work on two different projects. The chief engineer of the BESM-6 project, V. A. Mel’nikov, worked on a system called the BESM-10, which was to be a 64-bit successor built using integrated circuits. A. A. Sokolov began work around 1969 on a system called the AS-6. Little has been written about this machine; A. A. Sokolov has a reputation for caring little for writing and publishing. The AS-6 was a multi-machine system designed for reliable real-time control of objects. Running at 5 MIPS, its strengths were in inter-computer interaction, such as high data transmission throughput and direct memory access by I/O subsystems. Some distributed operating system and network protocol ideas were pioneered in it. The AS-6 was built in relatively few years thanks to the experience of the development team, the use of the same (i.e. available) component and construction base as the BESM-6, and the urgency of finishing the machine in time for use in the mission control system for the Apollo-Soyuz space project in 1975. The system for that mission included a number of tightly-coupled AS-6 and BESM-6 computers. This machine was

not compatible with the BESM-6. Sokolov felt that requirements for software compatibility ‘‘tied ones hands’’ and that achieving high performance was more important. Between 1975 and approximately 1980 on the order of 15 such systems were manufactured.

3.2.2.2 El’brus

Simultaneously, V. S. Burtsev and others, including B. A. Babayan, began designing a general-purpose multiprocessor dubbed the El’brus. Burtsev had previously worked on special-purpose multiprocessors for the military at ITMVT, and was proposing a large multiprocessor system which offered not only increased performance, real-time capability, high reliability, and increased main and peripheral storage, but also ease of programming and inter-generational software compatibility. The emphasis on software was something of a departure from ITMVT tradition, but would serve this line of development well in the future.

At the time the designs for the BESM-10, AS-6, and El’brus were emerging, Soviet leadership was becoming increasing concerned about the ‘‘computer gap’’ with the West in high-performance computing. Decisions had been made in the late 1960s to develop a Soviet line of IBM System/360 compatible machines to address deficiencies in general-purpose mainframes for economic applications. The introduction of Control Data Corporation’s CDC 7600 in 1969 highlighted a serious lack in high-performance systems for scientific applications. The CDC 7600 was over an order of magnitude faster than the BESM-6. While the latter was in full series production, no machines were ready to succeed it. There were few high-performance alternatives to the AS-6, BESM-10, or El’brus.[2] Discussions of which systems to support coincided with the need to select a

[2] The M-10, a vector-oriented parallel system designed for image processing is described in section 7.6. It undoubtedly profited from policy makers’ sense of urgency. It too had to be designed and developed, however. Supporting development of this machine would not resolve the issue of which systems to develop at ITMVT.

successor to the aging S. A. Lebedev who would soon (in 1973) step down as director of ITMVT. The AS-6, incorporating the same technology as the BESM-6, could easily be built at the Moscow SAM Plant. With the large-scale efforts underway to tool up for the manufacture of ES mainframes, Minradioprom did not have the production resources to support both the BESM-10 and the El'brus lines, however. Ultimately, this issue was settled less on technological grounds than political ones. Burtsev was able to line up more high-level support than Mel'nikov in the Military-Industrial Commission (VPK) and Minradioprom itself. The El'brus was supported, and Burtsev was selected to succeed Lebedev in 1973.

The El'brus computers are discussed in depth in chapter 4. These 64-bit machines were designed for the most demanding military and civilian applications. They had to offer not only high performance, but high reliability. The principal customers also wanted to be able to develop real-time software quickly and effectively. El'brus designers reviewed a number of Western projects and felt that these requirements were best met in the Burroughs 700 Series. Many of the features of these machines were worked into the El'brus: a stack-based multiprocessor architecture, modular construction, multiprocessing, dynamic resource allocations, hardware tags, hardware oriented towards efficient compilation and execution of programs written in high-level languages, software compatibility between system generations. The El'brus architecture is not a copy of the Burroughs systems, however, as we shall see.

With the El'brus computers, ITMVT continued its tradition of driving development in the Soviet computer industry. High requirements forced the development of new generations of integrated circuits, power supplies, cabling, design tools, printed-circuit boards, connectors, etc. The El'brus computers pushed the boundaries of each of these technologies simultaneously. The development times suffered as a result. The technical and ad-

ministrative challenges were staggering. Technically, delays resulted from having to develop, debug, and incorporate so many new technologies simultaneously. Administratively, each relationship with a factory had to be negotiated through a long chain of Ministry and Party officials. The factories themselves were generally disinclined to upset current production schedules with the introduction of new technologies and exercised a non-trivial *de facto* influence over production. As we can see from figure 3-2, the El'brus-1 and -2 took over 10 and 15 years, respectively, to reach series production. Similarly long development periods characterize other industry driving projects.

During the latter half of the 1970s, several events worked concurrently to shape ITMVT's future developments. First, the BESM-6 had been in production for a decade. Hundreds of systems had been installed, and a large and active user community had grown up around it. Something had to be done to support this community in the future. Second, the first prototypes of the El'brus-1 were under construction. Although less powerful than the El'brus-2 which embodied the original aspirations of the designers, the El'brus-1 gave users hope that a new generation of systems would be available shortly. Third, the AS-6 had been completed. This system would not be in production long, however, because many potential customers, believing that El'brus computers would soon be available, preferred to wait for the new machines. ITMVT in fact promoted this view. Fourth, Cray Research Inc. introduced the Cray-1 in 1976. Its clear demonstration of the viability of the vector-pipelined approach made a strong impression on computer engineers in both the USA and the Soviet Union. Its peak performance of 160 Mflops further underscored the lag in high-performance computing between the two countries. Fifth, during the late 1970s and early 1980s the political climate between the Soviet Union and the West worsened considerably. Computers were becoming more and more important in defense systems, and support for technologies of strategic importance in general grew. In

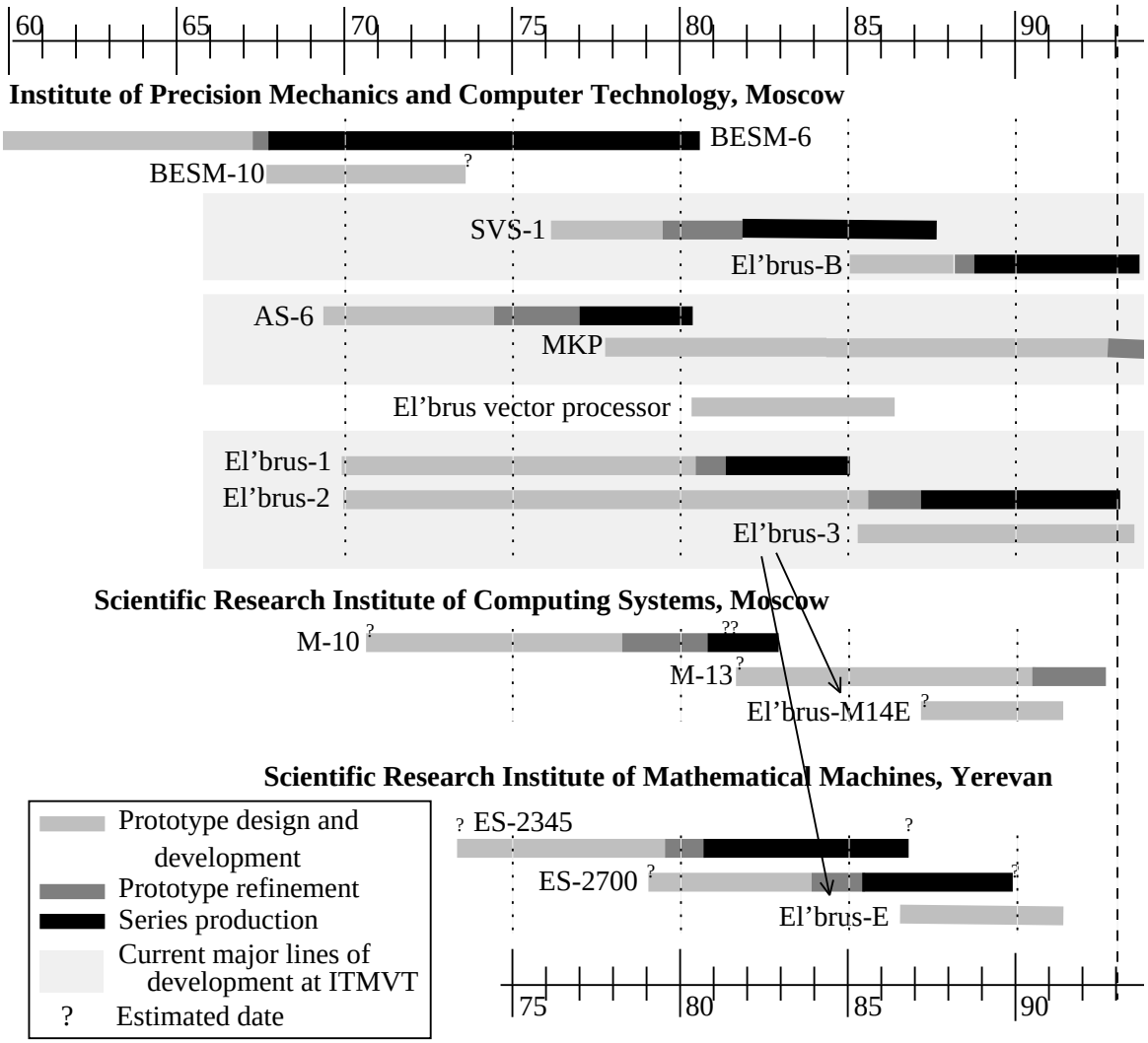


Figure 3-2 High-performance Computers at ITMVT and Related Institutes

the following sections we shall see how these factors shaped the direction of R&D at ITMVT.

3.2.2.3 SVS-1

To provide a migration path for BESM-6 users, engineers at ITMVT built a BESM-6 compatible processor called the SVS-1 which, using some of the same components and technologies as the El'brus computers, could be incorporated into an El'brus configura-

tion as a specialized processor. Like the BESM-6, it was a 48-bit machine. Its architecture was largely the same as the BESM-6's; small changes to improve performance were made to the adder, multiplier, and communications unit [Tyap82]. It was compatible with the BESM-6, although the BESM-6's operating system had to be modified to be able to function in the context of an El'brus configuration [Bala82].

One or two such processors could be incorporated into an El'brus configuration [Mvke80]. Configurations in which these were the only CPUs were called El'brus-1K2 and El'brus-1K4. SVS-1 based systems offered BESM-6 users improved performance and significantly better peripheral storage systems. The SVS-1 processor had an average performance of 2.5-3.5 MIPS. Because of the improved peripheral storage capability provided by the El'brus configurations, it could run 10-12 times faster than a BESM-6 on applications with large volumes of data [Mvke80].

Initiated around 1975 or 1976, the SVS-1 passed state testing in 1979 [Katk78; Tsan79; Mvke80; Supe91b, 13]. Between 60 and 100 units were manufactured between 1979 and 1987 [Supe91b, 13].

3.2.3 ITMVT Computers of the 1980s

3.2.3.1 El'brus Vector Processor

The Cray's vector-pipelined architecture inspired the development of a specialized vector-pipelined processor which could be incorporated into an El'brus configuration. Built using the El'brus-2 component base and construction, the processor employed a rather traditional vector-pipelined architecture with multiple, pipelined vector functional units and register files. In contrast to the Cray-1, which had three floating-point functional units (add, multiply, reciprocal), the El'brus-2 vector processor had two add, two

multiply, and one division floating-point units.[3] The El'brus vector processor also lacked scalar processing capabilities; this was to be provided by the native El'brus-2 processors. Like the Cray, the El'brus vector processor provided chaining between functional units to allow the output of one to be used directly as the input of another. The El'brus vector processor used a switch to provide chaining between any two functional units. Designed for a 40 nsec clock period, the vector processor would have had a theoretical peak performance of 125 Mflops [Burt85; Burt89].[4]

The vector processor was never completed. During the early 1980s conflicts arose between Burtsev and others within ITMVT and without. The details of the conflicts are not public, but they undoubtedly involved dissatisfaction with the progress of the El'brus program, differences over the future direction of El'brus development, and personal issues. Burtsev was removed as director in 1984 and replaced by G. G. Ryabov, formerly the head of ITMVT's computer-aided design division. Ryabov did not support the project, although he allowed it to continue for a couple of years with dwindling support. In 1986 the project was terminated and the division was disbanded [Mvs89, 44].

One of the basic issues about the best path of HPC development which divided Burtsev and others like Ryabov was the use of special-purpose vs. general-purpose processors. Burtsev felt that high-performance could best be achieved through the use of a general-purpose system with attached specialized processors. "[T]he solution to the problem of achieving maximum speed is best achieved in a multiprocessor complex with extensive use of specialized vector processors, alongside universal processors (oriented towards scalar computations)" [Burt89, 8]. Others, like Ryabov, held that any reasonable

[3] The El'brus vector processor also had a logic unit. Besides the floating-point units, the Cray-1 had three vector units (add, logic, shift), four scalar (add, logic, shift, POP/LZ) and two address (add, multiply). Since the Cray-1 could produce two floating-point results per 12.5 nsec clock period, the theoretical peak performance was 160 Mflops.

[4] Five results per clock period.

application would consist of a mix of scalar and vector operations and that general-purpose processors were the best means of achieving high performance on a broad set of applications. When Ryabov became director, the latter view prevailed. Work on the vector processor ended, while work on the Modular Pipeline Processor was accelerated. The use of specialized processors in the El'brus-3, a successor to the El'brus-2, also was rejected.

3.2.3.2 Modular Pipeline Processor

Following his work on the AS-6, A. A. Sokolov began working on a new computer called the Modular Pipeline Processor, discussed in section 7.2. On the one hand, this machine was influenced by Sokolov's previous work on the BESM-6 and AS-6 and by the vector-pipeline ideas of the Cray systems. The MKP implements pipelining ideas differently from the classical Cray-like architecture in that it integrates the vector and scalar operations more tightly into a single, general-purpose processor. On the other hand, the MKP was designed for a user community which could not necessarily afford systems as expensive as the El'brus. This factor made economy of hardware, ease of maintenance, and the ability to adapt the configuration to a variety of user needs major design and construction goals. The MKP is "modular" in the sense that a variable number of processors (theoretically with a variety of functional unit sets) can be combined in configurations with an assortment of other systems and peripheral devices.

Sokolov had worked on the MKP design since the late 1970s. Shortly after Ryabov became director, the project began to receive strong support. At this time, the technical characteristics of the next generation of ECL gate arrays were becoming known and the MKP was designed to achieve one Gflops in a dual configuration with a clock period of 10 nsec. The first full prototypes of this system were completed in 1990.

3.2.3.3 El’brus-B

In 1985, engineers began working on upgrading the SVS-1. In 1986, they changed plans and decided to develop a machine which, while compatible with the BESM-6 and SVS-1, had a new architecture and a new component base. The El’brus-B (also called the El’brus-KB) has three operating modes. The first mode provides full compatibility with the BESM-6 and SVS-1 processors. It uses a 48-bit word and 15-bit effective address and address registers. In the second mode, the word-length remains 48-bits, but the addressing is expanded to 27 bits. The third mode uses 64-bit operands and 27-bit effective addressing [Chai88, 25].

The El’brus-B is constructed using the same ECL chips used in the El’brus-2. By 1988 when the machine was first constructed, problems in the manufacture of these components had largely been ironed out; the El’brus-B is considered rather reliable. It runs at approximately five Mflops and has 64 Mbytes of main memory [Supe91b, 14; Usdi91, 30].

The El’brus-B passed state testing in 1988. On the order of 70 units have been manufactured at the Moscow SAM Plant since then [Supe91b, 13].

3.2.3.4 Expansion of the El’brus Family

As the El’brus-2 prototype neared completion, Babayan and others began work on the design of the next generation system. The new machines preserved many of the design principles of the El’brus-2: coarse-grain processors, modular construction, shared memory, multiple functional units, hardware tags, hardware support for high-level language constructs. One of the earlier El’brus design decisions which was to have a particularly profound influence on the new generation of systems was the use of a high-level programming language for all software, without exception. The lack of an assembler language made it possible to alter the underlying architecture significantly while maintaining

	Mico-El'brus	El'brus-E	El'brus-M14E	El'brus-3
Number of processors	1-10	2-8	2-8	1-16
Clock period (nsec)	125	40	25	10
Peak performance (processor)	8 MIPS	50 Mflops	160 Mflops	700 Mflops
Peak performance (configuration)	80 MIPS	400 Mflops	1280 Mflops	11 Gflops
Technology	CMOS	ECL	ECL	ECL
Architecture	RISC-like	VLIW	VLIW	VLIW
R&D facility	ITMVT	YERNIIMM	NIIVK	ITMVT

ITMVT Institute of Precision Mechanics and Computer Technology, Moscow
NIIVK Scientific Research Institute of Computing Systems, Moscow
YERNIIMM Scientific Research Institute of Mathematical Machines, Yerevan

Table 3-1 Target Characteristics of Recent El'brus Computers
Source: [Baba89, 879]

software compatibility. While the El'brus-2 has a stack-based architecture with ''as much control as possible'' given to the hardware, the El'brus-3 takes a very-long-instruction-word approach in which decisions about scheduling are made not by the hardware, but by the compiler. Existing code, including systems software, would have to be re-compiled, but not altered.

This feature made it possible during the early 1980s to consider expanding the El'brus line to embrace a broad spectrum of systems, from microprocessor to high-end supercomputer. Inspired by similar efforts by DEC and IBM, El'brus engineers worked on a series of upwardly compatible systems whose target characteristics are shown in table 3-1.

The El'brus-3 is discussed in chapter 4. The two mid-range El'brus computers were developed at the Yerevan Scientific Research Institute of Mathematical Machines and the

Scientific Research Institute of Computing Systems, Moscow. Both these projects were carried out in close cooperation with ITMVT. The VLIW machines are all compatible at the level of the architecture, but each implementation was carried out independently. At the time of this writing, construction of an El'brus-3 prototype processor is nearing completion. The other projects have been terminated.

3.3 The Proliferation of Soviet HPC Efforts (1978-1985)

During the late 1970s and early 1980s the number of HPC development projects within industry and academia increased significantly. Strong push and pull forces drove this trend. Except for the AS-6, no new high-performance systems had entered series production since the BESM-6 in 1966/7. Military users had a growing need for general-purpose and specialized high-performance computing systems for the new generation of weapons. For national security reasons, they often refused to rely on Western systems for critical applications. Pushing the development of new systems were many researchers within the Academy of Sciences, the Ministry of Higher Education, and industry who were dissatisfied with the policy of copying Western work which characterized the ES mainframe and SM minicomputer lines. By the end of the 1970s several largely theoretical or paper designs had progressed to the point where building a prototype was a possibility. Prominent academicians and members of the Academy of Sciences such as G. I. Marchuk, V. M. Glushkov, V. A. Mel'nikov and others wanted to build new machines with their own, often radical, architectural ideas.

3.3.1 Industrial Projects

3.3.1.1 Attached Array Processors

During the late 1970s and early 1980s the Soviet, Bulgarian, and East German computer industries began manufacturing attached array processors which could be used in ES mainframe configurations for computationally intensive tasks. In the West, a number of generations of attached array processors were developed during the late 1960s and early 1970s, the most successful of which was the AP-120B by Floating-Point Systems, Inc., introduced in 1976 [Hock88, 30-32]. Traditionally the primary applications have been image processing and the analysis of seismological applications, and the attached array processors initially were hardware systems designed for the execution of fast Fourier transform and other algorithms. Soviet, Bulgarian, and East German industry followed this trend and introduced a number of models described in section 7.12, beginning in 1979.

The attached array processors were not truly ‘‘high performance,’’ since the most widely used models (ES-2335, ES-2345, MAMO-1M, ES-2706) offered only 5-60 MIPS performance on 32- or 38-bit data. In the absence of other computers such as the El’brus, however, they offered significant performance improvements over the general-purpose mainframes available to the scientific community. It would be incorrect to say that such systems were developed because of delays in the El’brus program, but it is likely that because the latter were not available, the attached array processors proliferated to a greater degree than would have been the case otherwise. Hundreds of attached array processors were manufactured in the Soviet Union and Eastern Bloc countries, primarily in Bulgaria. By far, most of the latter were sent to the Soviet Union [Ivan88; Prat90]. Most of the AAP were used in geophysics applications [Tcha92, 33; Niko82; Prat90].

3.3.1.2 The PS- series

One of the only high-performance lines developed outside of ITMVT to reach series production was the PS-2000 (*perestraivayemaya struktura*–reconfigurable architecture) designed by researchers at the Institute of Control Problems (IPU) in Moscow and built at the Scientific-Research Institute of Control Computing Systems (NIIUVM) in Severodonetsk, Ukraine. Both institutes were part of the Ministry of Instrument-Building, Means of Automation, and Control Systems (Minpribor). Although not the most powerful, the PS-2000 was among the most successful Soviet HPC systems in terms of development cycle length and number of units installed.

The PS-2000 and its successors, the PS-2100 and PS-2300 are discussed at length in chapter 5. The PS-2000 achieved high performance through high levels of parallelism. It had a a single-instruction multiple-data (SIMD) architecture in which an instruction is executed simultaneously on multiple pieces of data by multiple processing elements. Part of the inspiration for such an architecture came from the ILLIAC IV computer developed at the University of Illinois during the late 1960s. With the ability to modify the communications links between processing elements and dynamically turn individual processing elements on or off, the system could be used successfully in a variety of applications with high degrees of data parallelism. The PS-2000 was developed primarily for use within the Ministry of Geology and Ministry of the Oil and Gas Industry for seismic exploration. With up to 64 processing elements and a theoretical peak performance of 200 MIPS on 24-bit data, the machine offered 1-2 orders of magnitude more computing power for applications which did not require great precision (>24 bits) than was generally available to these users.

This system had a remarkably short R&D cycle, as shown in figure 3-3. The first prototype was built within four years from the time financing was secured and a decision to

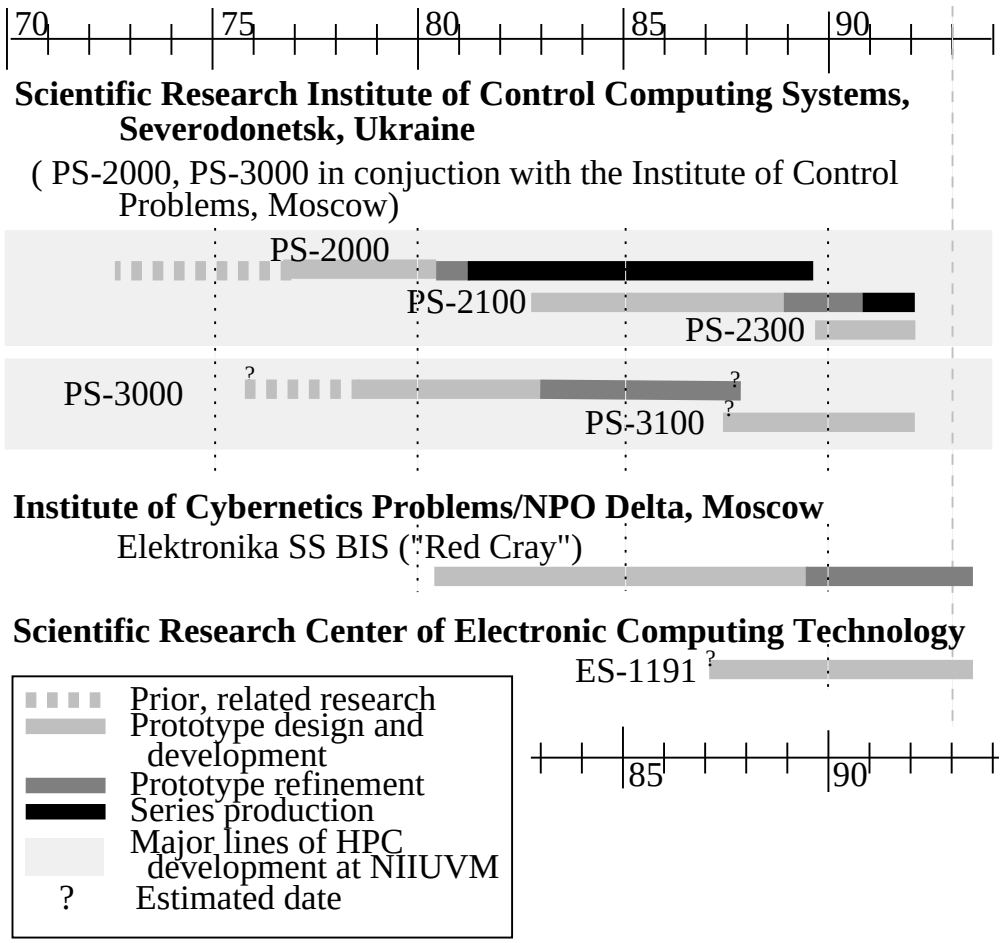


Figure 3-3 PS- and Other Industrial High-performance Computers

build the machine was made (1976). The machine entered series production only a year later and approximately 200 units were manufactured before production ended in 1989.

Several factors contributed to this success. The developers at NIIUVM had long been involved in developing systems for industry, and built the PS-2000 with a pragmatic eye for what could realistically be accomplished with the technologies at hand. Having many standard, identical pieces, the PS-2000 was also easy to manufacture, and relationships with the nearby factory were well established, thanks to administrative links in the Impul's NPO and prior development of control systems. The PS-2000 was not an industry

driver. To speed development, engineers almost exclusively used components already in series production. The project also profited from a demanding timetable and considerable high-level support. While the PS-2000 was under development, the Soviet leadership began a massive campaign to develop the nation's oil and gas resources. The PS-2000 was as well suited as any to address the computational needs of such a campaign, especially since during the late 1970s and 1980s American Presidents Carter and Reagan imposed technology embargoes, making it difficult or impossible for the oil and gas ministries to obtain and maintain Western computers [Mche81].

Thanks to stable target applications, requirements, and steady support for R&D from Minpribor and the Ministry of Geology, the PS-2100 is a very smooth and natural evolution of the PS-2000. The former incorporates up to ten times as many (640) processing elements arranged in modules of 64 PEs each, increased word-length (32- vs. 24-bit), more main and peripheral memory, etc. The theoretical peak performance of a full configuration is 1.5 GIPS. Primarily because of the need to develop a new generation of gate arrays for the processors, the PS-2100 took at least 50% longer to develop than the PS-2000. Approximately forty base modules (64 PEs each) were built before production ended in 1991.

The PS-3000 (described in section 7.5.1) was a parallel project also conducted jointly at IPU and NIIUVM. It was designed for use as the top level of complex hierarchical data processing and real-time control systems. Although it also has the PS- designation, the 32-bit PS-3000 has a very different architecture than the PS-2000. It consisted of two or four scalar processors, with each scalar processor associated with a vector processor. The machine incorporated two key features: a dynamic allocation of computing resources in which control units and processor fields were decoupled from each other (as opposed to the static coupling of traditional SIMD designs), and pipelining of processors. Because of

the limitations of the available hardware, however the true pipeline processing was not implemented. The vector processors consisted of a number of processing elements which in many respects mimic the operation of a true vector-pipelined processor. The peak performance of a full configuration was 20 MIPS.

Although it used an available component base, the PS-3000 had a much longer development time than the PS-2000. Possible contributing factors were the need to rework the preliminary design in order to implement it using the available component base, a desire on the part of systems developers to work on a new generation system rather than push the system through into production, and low demand for the machine. Only about 10 systems were manufactured.

3.3.1.3 Elektronika SSBIS

The Elektronika SSBIS was the most direct Soviet response to the introduction of the Cray-1 in 1976. In a sector in which original design was the rule, the Elektronika SSBIS is an exception. Work on a machine patterned after the Cray-1 began in 1980 by a team led by V. A. Mel'nikov, the chief-engineer of the BESM-6. He had left ITMVT in 1978 to work at the Del'ta Scientific Production Association in the Ministry of the Electronics Industry. That he, a disciple of the Lebedev school, should agree to implement a non-indigenous design speaks strongly of the emphasis policy makers at this time placed on initiating an effort to follow in Cray's footsteps [Sher92b, 1]. Nevertheless, as described in section 7.3, engineers did ultimately incorporate a number of distinctly non-Cray elements.

While the El'brus computers had some high-level characteristics in common with the Burroughs 700 Series computers, the underlying architecture has significant differences. The Elektronika SSBIS, on the other hand, was designed to be nearly compatible with the Cray at the level of assembler. The two machines are not binary compatible because de-

signers lacked detailed descriptions of the Cray construction and desired to improve the implementation of individual instructions. The Elektronika SSBIS also differs in the number of functional units, the number of instruction buffers, and other implementation specifics.

The Elektronika SSBIS exhibits the characteristically long development time of industry-driving Soviet large-scale, advanced computing projects. The machine took nearly ten years to build; the first prototype was completed in 1989. The machine was based not only on a new technological base, but also a new infrastructure. This was the first high-performance computer developed within the Ministry of the Electronics Industry, and although components were a primary product of the ministry, considerable time and effort were needed to establish relationships with the scores of additional factories and enterprises needed to develop and manufacture everything from CAD systems to printed circuit-boards, cables, etc.

3.3.2 Academic Projects

As in the West, Soviet researchers have for decades viewed parallelism as a promising path of achieving high computational performance. Given perennial problems with the performance and reliability of the component base and subsystems, it was widely held that parallel systems, in principle, offered a means by which performance comparable with Western machines could be achieved using the slower, less reliable components of Soviet industry. Parallel systems were also attractive to the research community because of the rich research opportunities. The most significant Soviet parallel systems developed by the academic research community, shown in figure 3-4, cover a very broad spectrum of architectural approaches.

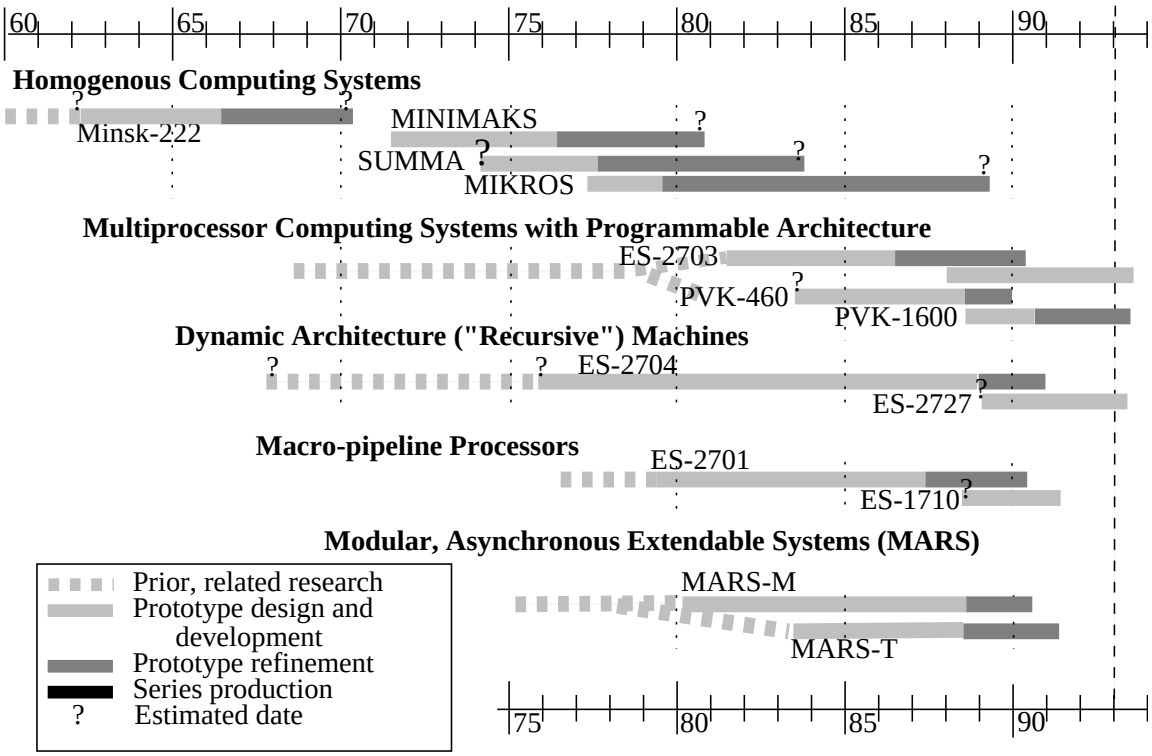


Figure 3-4 Academic Parallel Computing Systems

3.3.2.1 Homogeneous Computing Systems

Without question, the father of parallel computing in the Soviet Union was E. V. Yevreinov, who began research on parallel, multi-machine systems in 1960 at the Institute of Mathematics of the Siberian Department of the USSR Academy of Sciences. During the early 1960s he began working on combining individual computers into a single system to improve performance and reliability. He published his first work on this subject in 1962 [Yevr62].

Yevreinov pioneered the development of so-called homogeneous computing systems (OVS). This label covers a category of approaches to coarse-grain parallelism rather than a specific architecture. Tightly coupled parallel systems and geographically distributed

systems can all be homogeneous computing systems [Yevr81]. All OVS share three characteristics [Dimi82, 84]:

- parallel computation;
- a programmable, or reconfigurable structure.
- a homogeneous (single-type) of computing element;

The OVS are, fundamentally, a collection of computing modules which are linked together via a regular, tunable interconnect network. The computing module can be as simple as a processor plus memory, or as complex as a complete general-purpose computer with processor, memory, communications channels, external memory, I/O devices, etc. As a rule, the computing modules are all of a single type. This leads to a shorter development cycle, ease of manufacture, and ease of maintenance. It also facilitates the inclusion of additional computing modules and the development of systems software [Dimi82, 89-90]. A so-called elementary machine consists of a computational portion, the computational module, and a system unit which managed the interconnection and interaction with other computational modules.

All of the OVS developed by Yevreinov or his colleagues at other institutes were manufactured on the basis of computers already in series production.

The ability to program the interconnect system allows the OVS to be tuned to match the structure of a particular task. Bus, pipeline, matrix, and hierarchical configurations could be established by software. This capability also allowed the system to be reconfigured in the event of a hardware or software failure [Dimi82, 90].

Yevreinov was one of the first to explore issues of distributed processing, interconnect protocols, distributed operating systems, and software development for distributed-memory hardware platforms. Many of the prominent Soviet parallel computing projects carried out during the 1970s and 1980s trace their roots, directly or indirectly, back to

Yevreinov's work. A conference on homogeneous computing systems held in 1966 reportedly drew many of the individuals who would later lead parallel projects in programmable architecture systems, recursive architectures, reconfigurable systems, and others.

Yevreinov's first running system, called the Minsk-222, was completed in 1966 at the Institute of Mathematics in Novosibirsk [Dimi82, 81]. It consisted of two Minsk-22 and one Minsk-2 computers. The former was a 37-bit machine with 4 Kwords of memory running at 5000-6000 operations per second which entered production in 1964. The Minsk-222 was used designed for controlling scientific experiments [Dimi82, 81; Apok74, 214]. Another version of this system was completed by 1972. It consisted of 1-18 so-called elementary machines (EM) physically linked in a ring. Each elementary machine had a processing rate of up to 4000 operations/sec and 32K words of associated memory. The EMs are linked by seven control lines which, under the control of a lead-EM, govern the interaction of all EMs [Dimi82, 175-193].

Three other systems, some with several versions, were developed at the Institute of Mathematics during the 1970s and early 1980s: the MINIMAKS, SUMMA, and MIKROS.

Work on the MINIMAKS was started in 1971 and a prototype was completed in 1976 [Golo80]. Research using this machine was conducted into the 1980s, however [Yush81]. The design allowed for 2-64 elementary machines. The machine could accommodate any computer of the Aggregate System of Computer Technology (ASVT) minicomputers developed at the Impul's Scientific Production Association. This family of control computers was originally based on Hewlett-Packard minicomputers, but quickly evolved into an indigenous line. The M-6000, used in the MINIMAKS prototype entered series production in 1972, and had a performance of 200 KOPS. The MINIMAKS had two types of interconnect systems: one for data which linked each elementary machine with its four

nearest neighbors, and a control line which linked each elementary machine with two neighbors. The interconnect system employed the communications channels of the ASVT minicomputers. Tuning MINIMAKS structure took place through altering tuning registers which indicated the ‘‘active links’’ for a given elementary machine [Dimi82, 193-217].

Built during the late 1970s by V. G. Khoroshevskiy, a colleague of Yevreinov’s, the SUMMA consisted of 1-10 elementary machines, based on the Elektronika-100I minicomputer which had a processing rate of 30 KOPS [Kash79; Sedu79; Golo80; Khor82; Khom87]. Each elementary module was linked with three neighbors via half-duplex communications lines. Unlike MINIMAKS, it used only one type of communication line for both control and data transmission. The greater stability and reduced need for real-time response of the target SUMMA applications made this possible. Information was passed via a packet-switched approach using either address or name identification for routing [Dimi82, 217-229]. The first version of the SUMMA was completed in 1977.

During the late 1970s and early 1980s, the number of organizations directly involved with Yevreinov and others in OVS development expanded. This was in large part due to the fact that Yevreinov worked at a number of different institutes during this period. The MIKROS (*MIKRoprotsessornaya Odnorodnaya Sistema*–Microprocessor Homogeneous System) was developed by researchers at the Institute of Mathematics in Novosibirsk, the Novosibirsk Electro-Technical Institute, and the All-Union State Design-Construction Institute (VGPTI) of the State Statistical Agency in Moscow. It was based on the Elektronika-60 computer, a minicomputer software compatible with the Western LSI-11 minicomputer. In contrast to the OVS mentioned above, MIKROS had a much more flexible structure. While elementary machines in the MINIMAKS and SUMMA were connected to a fixed number of neighbors, elementary machines in MIKROS could be con-

nected with 1-12 others. The MIKROS could also be constructed as a geographically distributed system through adapters to telecommunications channels [Dimi82, 238-245]. This approach was used to build systems in other affiliated institutes, such as the Smolensk Branch of the VGPTI [Zuyk82]. The first MIKROS was completed in 1979, but work on it and others continued through the late 1980s. In 1989 a system with 24 elementary machines was being used as a platform for parallel systems software development [Mvs89, 6].

Other OVS were constructed at other organizations. Researchers at the Moscow Electro-Technical Institute for Communications built a universal communications computing system called the ‘‘Kollektiv,’’ designed for use in the control of communications systems [Dimi82, 232-235; Mamz84; Kudr84; Mamz89]. Also based on the Elektronika-60, this system used a ring-shaped trunck system channel containing 16 data and 13 control busses, making it possible to vary the number of computer modules (with full connectivity) from 2-16 [Yush81].[5] Based on the SUMMA architecture, the ME-80 was developed using Intel 8080 microprocessors at the Institute of High-Energy Physics in Dubna [Bald82]. The Statistika-1.0 OVS was developed at the All-Union Design-Engineering Institute of the State Statistical Association for use in an integrated system for information processing related to accounting, information services, planning, and management.[6] It also is based on the Elektronika-60. Its structure is basically hierarchical; Statistika-1.0 systems could be joined together in a hierarchical integrated system [Dimi82, 235-238].

Researchers in three other major lines of development–multiprocessor systems with programmable architecture, dynamic architecture (‘‘recursive’’) machines, and macro-

[5] As of late 1991, Yevreinov was reportedly employed here.
[6] By 1982, Yevreinov was working at this insitute [Ilyu82].

pipelined computers–view the homogeneous computing systems philosophy as an inspiration to their own projects. Each of these systems, to one degree or another, consists of a collection of identical processing elements with distributed memory and control. Although the PS-2000 is not considered an OVS because of its centralized control, the head of the laboratory (and later director) at the Institute of Control Problems where the PS-2000 was built, I. V. Prangishvili, had worked on OVS ideas during the late 1960s and early 1970s [Yevr74].

3.3.2.2 Multiprocessor Computing Systems with Programmable Architecture

The ‘‘multiprocessor computing systems with programmable architecture’’ developed at the Scientific Research Institute of Multiprocessor Computing Systems (NIIMVS) in Taganrog, Russia, under the direction of A. V. Kalyayev, were strongly inspired by the reconfigurable nature of Yevreinov’s homogeneous computing systems. This work is discussed in section 7.9. During the 1960s Kalyayev worked on building digital systems which modeled the functions of analog integrators. During the 1970s, he began working on more general-purpose systems based on so-called ‘‘macro-processors,’’ ‘‘macro-switches,’’ and ‘‘macro-memory’’ units. Underlying this work was the notion that the performance of a system is best when there is a good match between an algorithm’s control and data flows and the underlying hardware which executes it. The macro-switches provide a reconfigurable set of links which can be ‘‘tuned’’ dynamically to provide an interconnect system reflecting the nature of the problem being solved.

Kalyayev felt that a system should be user-oriented, i.e., should enable a user to operate the system in a language close to that used by practitioners in the applicataion domain. The MVS PA computers were designed primarily for mathematicians and designed so that the user would access a set of ‘‘macro-operations’’ (running on ‘‘macro-processors’’) which consisted of high-level mathematical operations (fast Fourier trans-

form, matrix multiplication, integration, etc.). During the 1980s, researchers at Kalyayev's institute built a number of systems incorporating these principles and designed and manufactured a number of processor, switching, and memory components to support their architectures.

3.3.2.3 Dynamic Architecture Machines

V. A. Torgashev and others worked on building dynamic (''recursive'') architecture systems, first at the Leningrad Institute of Aviation Instrument-Building, and later at the Leningrad Institute of Informatics and Automation. These ideas were first widely publicized at the International Federation of Information Processing IFIP '74 conference in Stockholm [Glus74]. These computers fall into a category of non-von Neumann architectures called data- or demand-driven. In data-driven (data flow) machines, the availability of operands triggers the execution of an operation. In demand-driven (reduction) machines, a request for a result prompts the execution (possibly recursively) of the operation which will generate it. In principle, such systems can ''seek out'' parallelism inherent in a program but not recognized by a programmer, improving performance. Torgashev's systems are discussed in section 7.7. In the dynamic architecture systems, a program is represented as a set of automata which represent data, operations, relations, references, or physical machine resources. Automata can be combined into more complex automata. A problem, represented by a dynamic automata network, its executed by applying transformations to the network until a final state is achieved which represents the solution. During the 1980s Torgashev developed a prototype system of which several units were built.

3.3.2.4 Macro-pipelined Machines

Researchers at the Institute of Cybernetics in Kiev worked on so-called ''macro-pipeline'' computers described in section 7.8. This approach incorporates a number of

coarse-grain processors which can operate in a systolic, pipelined fashion in which the data for an algorithm propagate across the processors, with each processor performing a significant amount of computation before sending the results to a neighboring processor. The programming language associated with such a machine, MAYaK, was designed to allow traditional, sequential programs to be converted to parallel systems. By packaging existing code with MAYaK instructions specifying inter-module communication, the program could be broken into pieces which could run in parallel and communicate with each other on multiple processors. A number of prototype systems were manufactured during the 1980s.

3.3.2.5 The ES-270x Systems

The basic research for the three systems just mentioned was conducted for years before full prototype systems were constructed. Carried out in academic institutes with little or no manufacturing capability, such systems could not be constructed without significant help from industry. Around 1980, individuals at Scientific Research Center for Electronic Computer Technology (NITsEVT) began an effort to provide research teams with ES mainframe technology and development resources to build prototype implementations of their ideas. Each of the projects had concentrated on demonstrating new ideas for constructing a computational engine. Researchers had designed the systems not as complete, industrial systems, but primarily to demonstrate new possibilities for carrying out parallel computation. NITsEVT agreed to provide both the technology to construct processor prototypes, as well as that which was lacking to enable the processors to perform useful work. As a result, the above systems were built on the basis of ES mainframe technology and were linked by standard ES channels to a mainframe host which provided needed I/O, systems management, and user interaction facilities.

Although the projects were independent of each other, they each received the designation ES-270x, indicating that they were a processor attached to an ES mainframe system: ES-2700 (attached-array processor), ES-2701 (macro-pipelined system), ES-2702 (symbolic processor at the Institute of Applied Mathematics, Moscow), ES-2703 (multiprocessor computing system with programmable architecture), ES-2704 (dynamic architecture machine), ES-2705 (analog parallel system, Riga Polytechnical Institute). Another system fitting into this naming scheme, the ES-2706, was a Bulgarian attached array processor and not directly supported by NITsEVT. NITsEVT coordinated efforts to push proposals through policy-making and funding organizations. During the early 1980s a series of resolutions of the USSR Council of Ministers provided the necessary support for the various development stages of the ES-270x machines. Such efforts were in keeping with some of the stated goals of the 11th Five Year Plan [Uprs81; Myas82].

NITsEVT committed to supporting these projects through the prototype stage. Production of the systems depended on finding production facilities willing to manufacture the systems, and gathering enough political support to overcome any resistance by factories to introducing new products. Both depended on the existence of customers who would purchase the machines once they were manufactured; having adequate financing for R&D was not sufficient. Factories were also interested in the ease with which machines could be manufactured using the technology available.

Efforts to arrange for series productions of the machines were ultimately unsuccessful. Although some (single digits) of potential customers were found for some of the machines, the market was not large enough to make it worth factories' while to manufacture them. From the users' perspective, there were fundamental problems with each of the systems. Lacking their own I/O facilities and systems management capabilities, the systems had been built out of necessity as attachments to standard ES mainframe hosts.

These configurations were generally not able to supply data to the processors at a rate that would support the claimed performance levels. Plans were made for giving the processors direct access to main and peripheral storage of the host computers, but the prototypes which were constructed did not have such features. Second, with the exception of the ES-2701, the systems required significant re-coding in non-standard languages. Third, although manufactured with NITsEVT support, the projects varied in the degree to which they conformed to accepted construction practices at the factories. The greater the number of different kinds of circuits and modules, the greater the factory's difficulty in manufacturing the system. Reportedly, the ES-2704 had a relatively small number of different types of modules, while the others were more complex, and therefore less desirable from the perspective of a factory.

3.3.2.6 MARS

The Modular, Asynchronous, Extendable Systems (MARS) were developed at the Computing Center of the Siberian Department of the USSR Academy of Sciences. The conception for these systems were developed by G. I. Marchuk and V. Ye. Kotov and published in 1978. Formerly the director of the Computing Center, Marchuk at that time was President of the Siberian Department. Later, as the chairman of the State Committee of Science and Technology, he was able to provide considerable high-level support for the implementation of the MARS ideas.

In their analysis of computing trends during the 1970s, Kotov and Marchuk identified as key the broadening of the sphere of application of computer technology, the compound or systemic nature of the problems to be solved, the transition from traditional von Neumann machines to computing systems with a variety of configurations, capabilities, and purposes, the miniaturization of the component base, and the trend towards higher-level man-machine interfaces.

The key architectural principles of the MARS are parallelism, decentralized information processing and data flow, asynchronous interaction of devices and processes, a hierarchical structure, specialized systems components and hardware implementation of complex data processing functions, modularity, reconfigurability, and self-identification of data and processes.

Like the academic systems described above, the implementation of the MARS conceptions required the assistance of industry. While the ES-270x systems were supported by NITsEVT, MARS was supported by ITMVT, thanks in part to the close relationship between Marchuk and V. S. Burtsev, ITMVT director from 1973-1984. Marchuk was also able to secure funding through the GKNT.

Two systems embodying MARS ideas to varying degrees were developed between 1980 and 1990. The first of these, called the Mini-MARS (later, the MARS-M) was a shared-memory heterogeneous multiprocessor incorporating a variety of advanced architectural ideas including decoupled architecture, very-long-instruction-word (VLIW), and virtual heterogeneous multiprocessors. The second, called MARS-T, was a multiprocessor with computational elements patterned after the transputer and the stack-based Lilith processor developed by Niklaus Wirth (author of Pascal and Modula-2).

Between 1985 and 1988, these projects and others were supported through the so-called START project. In 1981 the Japanese had launched the Fifth Generation Project, which placed artificial intelligence at the core of a program oriented towards the development of a new generation of computers. Soviet researchers and policy makers began discussing the need to launch a Soviet response. Several prominent AI researchers joined with Kotov to push an AI/hardware/software program through policy making channels, and found it useful to cast their work as a response to the Japanese efforts. Thanks in large part to the help of G. I. Marchuk, these efforts were successful and the three-year

START program was created. The START program, and the MARS computers more generally are the subject of chapter 6.

Prototypes of the MARS-M and MARS-T were built by the conclusion of the START project in 1988. In spite of the high levels of government support, the high-profile nature of the project, and favorable organizational conditions, the projects still suffered from the administrative gap between industry and the Academy of Sciences, and the nature of supplies and operations at industrial factories through the mid-1980s. During the late 1980s, support for MARS research and demand for such computers dwindled. Work on hardware development ended by 1991.

3.4 1985-present

Soviet HPC developments during the latter half of the 1980s took place within the context of a growing gap in computing power and availability between the East and the West; established HPC projects initiated during the 1970s and early 1980s; and major, disruptive efforts to reform Soviet society and the economy. Technological advance in the West during the early 1980s was relentless, with firms like Cray and Control Data Corporation introducing new models every 2-4 years with double or triple the performance of their predecessors.

3.4.1 Policy-making Developments

From the perspective of high-performance computing, key policy-making events were the creation in 1983 of the Department of Computer Technology, Informatics, and Automation (OIVTA) of the USSR Academy of Sciences, and the Comprehensive Program of Scientific-Technical Progress of Member Countries of the Council of Mutual Economic Assistance to the Year 2000 (Program to the Year 2000), adopted in December, 1985 [Sama85; Prav851219]. The latter called for broad cooperative efforts in five basic cate-

gories: ‘‘electronification’’ of the national economy, comprehensive automation of manufacturing, nuclear energy, bio-technology, and development of new technology for the production and processing of new materials. The goals of ‘‘electronification’’ included the development of supercomputers having speeds of more than one billion operations per second, large numbers of personal computers and software, the development of a unified systems for transmitting digital information, the development of new satellite communications and television systems, and the development of new electronic devices and fast, reliable integrated circuits [Prav851219]. The GKNT was given over-all responsibility for organizing and coordinating under this program [Byal90, 19].

The Academy of Sciences had lost much influence in establishing computing policy during the late 1960s and early 1970s when the ES and SM programs were implemented. With the creation of the OIVTA the Academy sought to re-assert its role as a consultative body in the field of computing and consolidate and coordinate informatics research within the Academy by providing an official forum in which issues related to computer technology could be discussed and policy formulated [Mikh84].

The OIVTA played a leading role in crafting the Program to the Year 2000 [Guse83; Yasm85]. Several individuals prominent in Soviet HPC were members of the OIVTA: G. G. Ryabov, B. A. Babayan, V. S. Mikhalevich (director of the Institute of Cybernetics in Kiev), V. P. Ivannikov, V. Ye. Kotov, and A. V. Kalyayev [Alek84; Izv841224]. The Institute of Cybernetics Problems was created as part of the OIVTA and became home to the Elektronika SSBIS project. One of the main goals of the new department was the creation of supercomputers and, thanks to OIVTA efforts, the Program to the Year 2000 included plans for the ‘‘development of new supercomputers having speed of more than one billion operations per second...’’ and a machine with a performance of 10 billion operations per second by 1990 [Prav851219; Veli85; Sama85; Veli87; Marc87].

Several Soviet high-performance computing projects received financing under the umbrella of this program from a variety of sources such as the State Committee on Science and Technology, the Academy of Sciences, and individual Ministries and enterprises. Among the projects targeting the one billion-plus operations per second goal were the MKP and El'brus-3 of ITMVT, the ES-1191 of NITsEVT, the PVK-1600 and others at NIIMVS, ES-2727 successor to the ES-2704, and the ES-1710 successor to the ES-2701 [Przh89, 36-37]. The MARS (START) program, with its emphasis on new computer architecture and artificial intelligence also benefited from the Program to the Year 2000's emphasis [Koto87].

The Program to the Year 2000 was designed to be an international effort, and some lip-service was paid to cooperation with other countries in HPC R&D. For example, [Ivan87] reports participation by Hungarian and Polish specialists in the creation of high-level artificial intelligence languages and power supplies for the El'brus-3 and MKP. In practice, however, whatever international contact there may have been in these and other high-performance projects, they were at best minimal and did not have a significant impact on the technology. Soviet high-performance projects remained almost exclusively Soviet.

3.4.2 Developments in High-Performance Computing Systems

Although it provided a vehicle for channeling development funds to the high-performance computing sector, the Program to the Year 2000 did not qualitatively change the landscape of Soviet high-performance computing. The technological developments in high-performance computing during the latter half of the 1980s are best characterized as a logical extention of lines of development begun during the 1970s and early 1980s in which basic design philosophies were preserved.

The ES-1191 (figure 3-3) and the loosely-coupled array processor configurations were the only ‘‘new entries’’ in the Soviet HPC sector during the latter half of the 1980s. Other projects had either been initiated previously, or were evolutionary extensions to previous projects. A number of projects, including the ES-1068.17 and ‘‘Sibir’’’ systems described in section 7.12.6, consisted of a mainframe host with a number of attached array processors. The ES-1191 represents the first direct efforts of the Scientific-Research Center of Electronic Computing Technology (NITsEVT) to develop a high-performance computing system. NITsEVT had assisted with the ES-270x projects, but until the mid-1980s had concentrated its own development efforts on building high-end mainframe systems.

The ES-1191 was perhaps inspired by IBM’s efforts to enter the supercomputer market through the introduction in 1985 of its 3090 VF, a 3090 mainframe with one or more integrated vector processors. The ES-1191 also integrates vector processors into a system based on a general-purpose mainframe, but the design differs considerably from the IBM machines. Seeking not only to break the 1 Gflops performance barrier, but also provide high average processing rates, preserve the systems software, and clearly separate the computationally intensive tasks from the routine system management tasks, engineers designed a machine with a computational subsystem including both scalar and vector resources. The scalar processors treat the vector processors as a shared resource. This machine is described further in section 7.4. The ES-1191, originally scheduled for completion in 1989 experienced delays and reductions in financing. Nominal work continues on the system, but the prospects for completion are very poor.

In spite of the inability to secure production facilities, R&D continued into the early 1990s for successors to the original ES-270x systems. Support for the MKP, El’brus-3, and Elektronika-SSBIS remained constant, if not necessarily adequate. The latter had

production facilities available, but the relationships with the myriad of upstream factories became more tenuous as economic relationships were transformed from a centralized command mode to horizontal links based on mutual benefit. The system of state orders *(goszakaz)* which replaced the centralized Plan as an agent of centralized control of production helped maintain these links, but the state orders grew increasing unable to compensate for basic production shortfalls, discontinuities in the supply chains, and growing local autonomy. The deteriorating financial state of nearly all Soviet organizations, their *khozraschet*-related tendency to manage their money more tightly, and the highly uncertain government budget allocations devastated the market for HPC.

As work progressed during the late 1980s, the Soviet economy deteriorated, the infrastructure necessary for building new machines began to fracture, and the market evaporated. In 1989 the Eastern Bloc dissolved and the Program to the Year 2000 became irrelevant. The system of Five Year Plans ended in 1990 and large-scale ‘‘goal-directed programs’’ for the advancement of one branch of the economy were largely phased out as tools of economic management. Trade relationships with the Eastern European countries were severely disrupted with the conversion to trade based on world prices at the beginning of 1991 [Izv910720; Newm91; Seme92]. In particular, the supply of attached array processors from Bulgaria and the former East Germany ended as prices started being computed in hard-currency terms, and Soviet users lacked, or were unwilling to pay, the necessary amount of money. Financial support for high-performance computing became more fragmented, dependent on the agendas and abilities of individual government funding organizations and HPC customers.

3.4.3 Soviet Computing Associations

In response to the growing crisis, developers, manufacturers, and users of high-performance computers formed two associations to promote their own interests. Leaders

at ITMVT, the Scientific-Research Computing Center at Moscow State University, and other organizations spearheaded the creation of the ‘‘Supercomputer Association of Users, Developers, and Manufacturers of High-Performance Systems,’’ officially organized in January, 1991. This association, described at greater length in section 4.8.3, was created to draw together the various players in the HPC sector to encourage closer cooperation, and serve as a mouthpiece and lobbying force for the HPC sector to help secure government support. The Association held its first conference on the ‘‘Problems of the Development of High-Performance Computing Systems’’ in October, 1991 to coordinate efforts, and discuss issues related to the basic survival of the industry. Although the association has been so strapped for funds that its activities in 1992 were minimal, its efforts in 1991 did play a role in securing basic funding for all prominent on-going HPC projects during 1992.

In 1990, the Soviet Transputer Association was formed [Usdi91, 30].[7] The name of the association is almost a misnomer, since it includes not only organizations and individuals using Western transputers, but also Soviet organizations working on distributed memory multiprocessor systems. Most of the parallel projects discussed above which trace their inspiration back to Yevreinov’s homogeneous computing systems (OVS) fall into this category: Torgashev’s dynamic architecture machines, the macro-pipeline systems at the Institute of Cybernetics, Kiev, the multiprocessor computing systems with programmable architecture in Taganrog, Khoroshevskiy’s work on homogeneous computing systems in Novosibirsk, and others. Each of these groups is a member of the Russian Transputer Association, and each has begun working to port their systems software to a transputer-based hardware platform [Niim90, 7-8]. Individuals at the Scientific Re-

[7] Following the breakup of the Soviet Union, the association was renamed the Russian Transputer Association.

search Institute ‘‘Kvant’’ in Zelenograd built a three-chip transputer-like processor [Niik92; Korn92]. Researchers at the Institute of Applied Mathematics and elsewhere use transputer platforms to model other computer architectures, develop parallel algorithms, and perform complex data processing tasks [Zabr91].

In spite of the broad differences in computer architectures, the multiprocessor, distributed memory approaches taken by these groups are not altogether incompatible with the basic ideas of transputer construction. In fact, the transputers simply provide a hardware platform of basic processors together with integrated communications facilities. With the appropriate systems software, the basic multiprocessor hardware can be used to model a variety of other architectures. While the resulting system might not be as efficient as a customized hardware system, transputers provide basic facilities that free researchers who want to conduct research in parallel systems, algorithms, and applications from having to expend much time and resources on building the hardware itself. In the Soviet context, as trade (legal and illegal) with the West in computer components grows and the prospects for enlisting computer factories in the development of new hardware systems were worsening, transputers can provide an alternative to many Soviet scientists.

The Russian Transputer Association reportedly has over two hundred members. It was formed on the initiative of V. K. Levin, the director of NII Kvant, in part in response to a government program to promote transputer-related research within the Soviet Union [Mvs89, 58]. The association does not finance research, but works to facilitate contacts with the Western transputer community and the flow of information among individuals involved in Soviet transputer-related research. Several international conferences organized by the RTA have been held in Russia and RTA members have traveled abroad to transputer conferences. The association also provided expert advise to policy makers who had to evaluate and fund research proposals in computing.

CHAPTER 4. INSTITUTE OF PRECISION MECHANICS AND COMPUTER TECHNOLOGY

4.1 Introduction

In this chapter we examine developments at the Institute of Precision Mechanics and Computer Technology (ITMVT), primarily within the division devoted to the El'brus multiprocessors. ITMVT has been without question the most active and prominent R&D facility for Soviet high-performance computing for four decades. Pursuing the goal of developing the fastest machines possible, ITMVT has played a decisive role in shaping the high-performance computing landscape, and the broader computing industry as well.

The El'brus computers have been the most powerful systems manufactured in the Soviet Union and have been principal tools of that country's most demanding computer users. The third generation of this family is now nearing completion. In tracing the development of this series, we learn much about the factors–technical, social, political, and economic–which have shaped their evolution in the Soviet context. These systems give us considerable insight into the process of developing large-scale advanced technologies within the Soviet system. We will examine the impact of the reform-related changes have had on the technology and the organizational context within which they were developed, and discuss the prospects for large-scale development in the future.

4.2 History of ITMVT Research

The Institute of Precision Mechanics and Computer Technology was established in 1948 in Moscow under the USSR Academy of Sciences through the merger of the Department of Precision Mechanics of the Institute of Machine Studies, the Electric Simulation Laboratory of the Institute of Energy, and the Department of Approximate Calcula-

tions of the Institute of Mathematics [Golo88, 24; Itmv90; Crow93]. The first director, from 1948-1950 was N. G. Bruyevich.

The Institute's role in Soviet digital computing began in 1950 when academician M. A. Lavrent'yev became director and established a laboratory for the development of electronic digital computers [Bard87; Bard88]. He invited S. A. Lebedev, the father of Soviet digital computing, to Moscow as its head. Lebedev had built the Soviet Union's first electronic, digital, stored-program computer, the MESM, in Kiev between 1947-1951 at the Institute of Electrical Engineering. He had had the strong support of Lavrent'yev who was the vice-president of the Ukrainian Academy of Sciences from 1945-1948 [Itmv90; Crow93].

We discussed the contributions of S. A. Lebedev in chapter 3. Throughout the 1960s and 1970s he established ITMVT as the leading R&D facility for Soviet high-performance computing and cultivated a strong tradition of building fast machines with original architecture for use in real-world applications. He pioneered advances in all aspects of computing, and initiated a tradition of working as closely as possible with industrial design bureaus to shorten development cycles and ease the transition to series production.

Besides these computers, ITMVT workers built a number of special-purpose systems, primarily for the military, about which almost nothing has been written in the open literature.[1]

During the early 1970s, following the introduction of the CDC 7600, Soviet leadership became increasingly concerned about the ''computer gap'' with the West in the area of high-performance computing. Heated discussions on the topic were held at the highest levels of the Military-Industrial Commission (VPK), the Academy of Sciences, and the

[1] These include the 5E92B (1964), 5E51 (1967), 5E65 (1968), 5E67 (1970), 5E26 (1974) [Laza91b].

ministries involved in computing. These discussions coincided with those about who should succeed S. A. Lebedev as the director of ITMVT, who stepped down in 1973 and died in 1974 [Glus78b; Bard87]. The leading candidates for the position were V. A. Mel'nikov, the chief engineer of the successful BESM-6, and V. S. Burtsev, the chief engineer for some real-time special-purpose machines for the military. Mel'nikov was proposing the BESM-10, a general-purpose multiprocessor successor to the BESM-6 built with integrated circuits. Burtsev proposed the El'brus, named after the highest mountain in Europe. This family is discussed at length below.

Both machines could not be supported. Financing could have been allocated easily by government decree, but manufacturing capacity was more difficult to create. At that time Minradioprom did not have sufficient facilities to manufacture both. The Moscow SAM Plant was busy with mass production of the BESM-6 and anticipated production of the AS-6. A number of other major Minradioprom factories, such as those in Minsk, Penza and Kazan', were being converted to the manufacture of ES mainframes. The Zagorsk Electro-Mechanical Factory (ZEMZ) was to be the primary facility for the next generation ITMVT machine, but it did not have the capacity to carry out two major projects at once.

The discussions and maneuvering for support deeply divided ITMVT, the user communities for ITMVT machines, and the Soviet industrial leadership. Burtsev was selected to replace Lebedev because he proved more successful in establishing a base of support. Mel'nikov had the support of a large portion of the BESM-6 user community, in the Academy of Sciences and among certain groups of military users such as nuclear weapons designers. This included the Father of the Soviet space program, Academician M. V. Kel'dysh who was the head of the Institute of Applied Mathematics. These users had an investment in code which ran well on the BESM-6 that they wanted to preserve.

Furthermore, many potential users were alarmed at the complexity of the machine Burtsev was proposing. Burtsev, on the other hand, was able to enlist the support of the influential Academician G. I. Marchuk as well as large segments of the military user community, such as the rocket-builders who were most interested in real-time applications. Burtsev also had the all-important support of the Minister and Deputy-Minister of Minradioprom, V. D. Kalmykov and V. S. Semenikhin [Grea83; Kalm72].

When Lebedev resigned his position in 1973, a year before his death, Burtsev was appointed director, ensuring that the El'brus program would live. Shortly thereafter, Mel'nikov left ITMVT and founded a new organization, the Delta Scientific Production Association in the Ministry of the Electronics Industry, taking a number of the BESM-6 systems programmers like V. P. Ivannikov with him. Other BESM-6 engineers, mainly those who were working on the AS-6, remained at ITMVT.

4.3 El'brus-1 and El'brus-2

4.3.1 Requirements

To a large extent, the requirements which most strongly shaped the El'brus design were dictated by military applications. Most of the customers for ITMVT machines, and virtually all customers for the El'brus machines, were in the military-industrial sector.[2] Their applications included controlling space missions, operating anti-ballistic missile and real-time radar installations, and running atomic energy stations. The El'brus were also designed for large-scale, computationally intensive scientific applications including weapons design. These applications demanded above all high performance, and high reli-

[2] In 1991, G. G. Ryabov stated that over 80% of the customers for [all] ITMVT machines (not just the El'brus) were in the military industrial complex [Laza91b]. No figures are available for the early 1970s, but the percentage was undoubtedly quite high then as well.

ability. The latter was necessary both in real-time applications where system failure could cost lives, and in scientific computation where a system that crashes midway through a lengthy computation is not useful. Reliability requirements were to have a particularly strong influence on the design, since designers had to incorporate many fault-tolerant features into the architecture to compensate for unreliable components provided by the Ministry of the Electronics Industry (Minelektronprom).

The principal customers for the El'brus also had a strong need to be able to develop real-time software quickly and effectively. Ease and efficiency of programming became a third requirement which powerfully shaped the El'brus hardware and software design. Finanlly, the El'brus was also to serve at the heart of large-scale, distributed data processing centers. Extensive I/O and data transmission facilities were important.

In a 1975 article, Burtsev summed up the major trends in computer development which were incorporated into the El'brus [Burt75]:

- high reliability;
- ease of software development;
- inter-generational software compatibility;
- increased main and peripheral storage;
- the ability to link geographically distributed equipment into a central data processing center.

In particular, he noted the trend in computer languages away from physical addresses, making software development easier. At the systems level this leads to the concept of virtual memory. Within processors, this principle can be reflected in not using explicit addresses for registers but dynamic allocation. As we shall see, this concept has significant consequences for the El'brus.

4.3.2 Design Antecedents

The ideas for how to design a machine to meet these requirements came from a variety of sources. An early ITMVT machine called the 5E92B, developed in 1964, was able to detect and correct all single-bit errors in hardware. It established a precedent for placing significant control in hardware [Laza91]. The principle of hardware control was taken to new heights in the El'brus-1 and -2. Not only were fault-tolerant features incorporated into the hardware, but also support for operating systems and high-level languages, and advanced instruction scheduling features as well.

Although some of the other ideas implemented in the El'brus–modular architecture, machine support for high-level languages, tagged architecture, etc.–had been implemented to some degree in earlier Soviet machines, one of the strongest influences on the thinking of the El'brus designers came from the writing of J. K. Iliffe and the Burroughs 500 and 700 Systems which incorporated many of Iliffe's ideas.[3]

In [Ilif68], Iliffe defines a computer system from a programmer's point of view. Rather than view a system as a linear store of both instructions and data (classical von Neumann model), Iliffe sought to define a non-linear system structure which reflected the structure of programs in a multiprogramming environment. He proposed using a tree structure for program and data segments and a hierarchy of processes. Such an arrangement would support dynamic control over data structures and processes, and support interacting, parallel processes [Ilif68, 1-15]. To further simplify programming and enhance the ability of the system itself to monitor processes, Iliffe proposed incorporating a means of indicating, at the hardware level, the type of individual data elements [Ilif68, 11-12]. If the machine could distinguish between addresses and instructions, and dynamically interpret operands, the number of instruction/data type combinations specified in the in-

[3] Iliffe developed and helped implement many of his ideas before the publication of [Ilif68].

struction set could be reduced significantly, and programmers could be insulated from local optimization of the address and instruction codes [Ilif68, 33]. To implement data typing in hardware, Iliffe proposed using hardware tags [Ilif68, 34-35].

In 1961, Burroughs introduced the first commercial computer using a tree structure, the B 5000 [Ilif68, 25; Orga73, vii]. This was one of the first machines to incorporate a push-down stack for operands. A stack readily lends itself to representing the structure of block-structured, procedure-oriented languages–of the Algol family, for example–which are characterized by nested blocks that define the scope of an algorithm's variables and identifiers, and allocation and deallocation of dynamic resources. A stack is a useful structure for representing the execution state of structured programs. A program block's data and instructions can be pushed onto the stack as the block is entered, and popped off when control exits the block. The stack reflects the context of the active block.

The B 5000 was followed by the B 5500 and B 6500 in 1969 and, in 1970, Burroughs introduced the B 5700, B 6700, and B 7700 mainframes. The latter systems refined and expanded the ideas pioneered in the B 5000. Key design goals for the Burroughs machines, as for the El'brus, were reliability, high speed, and ease of programming. Some of the key design features used to achieve these goals were:

- a modular structure consisting of multiple CPU, memory, I/O, and data transmission modules which were treated as shared resources in a single, integrated system;
- multiprocessing and multiprogramming as a normal mode of operation;
- dynamic allocation of system resources, including CPUs, memory, I/O and data transmission processors;
- hardware oriented towards efficient compilation and execution of programs written in high-level languages;

- stack-based complex instruction set computer (CISC) central processing units;
- hardware tags;
- software compatibility between machine generations.

Each of these principles and many implementation details were adopted by the El'brus developers. In implementing them, El'brus designers were guided by a basic philosophy that the machine should have an integrated hardware/software design, as had been advocated by Iliffe. Consequently, they felt that as much control as possible should be implemented in hardware to simplify programming of systems and applications software, and all programming without exception should be done in a high-level language. During the design phase Burtsev, Babayan and their coworkers examined several Western systems (including the CDC 6600, the MU5 project at Manchester University, and the Multics operating system) to see which would best support such an integrated design. They felt that the Burroughs machines offered the best opportunity.

The philosophy of an integrated design was held to a non-trivial degree by the Burroughs designers, but as we shall see, the El'brus designers carried them to new levels. In particular, the El'brus designers decided to implement a single language, later called El'-76, in place of multiple languages oriented towards separate programming functions such as systems programming, applications programming, job control, etc. Another basic difference was the implementation of a more advanced form of virtual memory, described below. These decisions led to the development of an instruction set, programming language, and compilers which different significantly from that of the Burroughs machines.

Two core design principles of the El'brus which were hardly implemented in the Burroughs machines were achieving high performance through:

- multiple functional units;

- dynamic instructions scheduling performed by hardware.

4.3.3 Burroughs/El'brus Comparison

In this section we describe the architecture of the El'brus computers and compare and contrast it with that of the Burroughs B 6700.[4] A detailed description of the B 6700 and El'brus is beyond the scope of this study; interested readers can find more information in [Orga73; Burr72; Zamo85; Baba90]. We focus on some of the principal features of the machines, highlighting points of similarity and difference on both the conceptual and implementation levels. Such an analysis will reveal the degree to which the Soviets adopted not only specific architectural features, but also important guiding principles which were to shape the development of this family in future years, even after the Burroughs architecture had been largely discarded. We will also be able to determine how advanced the Soviet work was, relative to the world-wide state-of-the-art.

While the El'brus and Burroughs machines have considerable similarities to one another, they have both quantitative and qualitative differences. Quantitative differences are those in which the essential nature of an architectural feature is the same in two systems, but the degree of implementation (number of units, volume, size, etc.) differs. Quantitative changes in one part of a system frequently necessitate qualitative changes in another part of the system. There are a number of examples of such patterns in the El'brus vis-à-vis the Burroughs machines. The qualitative differences in a number of cases reflect indigenous innovations and Soviet contributions to the computing field.

[4] In this section we limit our discussion to the El'brus-1 and El'brus-2 which have essentially the same architectures. Unless otherwise stated, comments referring to the El'brus computers in this section refer to both.

4.3.3.1 System Organization

Table 4-1 provides a comparison of the features of the Burroughs 6700 and El'brus computers which differ quantitatively. Like the Burroughs machine, the El'brus computers consisted of multiple central processing units linked by a crossbar switch to shared main memory modules. Independent I/O processors have direct access to the memory modules as well, and relieve the CPUs of much I/O overhead. Data Transmission Processors linked to the I/O processors are also independent digital computers responsible for interfacing with a wide variety of peripheral devices employing a wide array of telecommunications line disciplines. The architecture of the El'brus computers is shown in Figure 4-1.

4.3.3.2 CPU

The central processing units of the B 6700 and the El'brus computers are stack-based, using a zero address CISC instruction set and reverse Polish notation. By using a stack-based architecture and a number of other features, Burroughs and El'brus designers developed machines whose hardware reflected the structure of the software sufficiently well that most, if not all, code for the system could be written in a high-level language.

The object code of a a compiled program consists of a set of segments. A segment generally corresponds to a single procedure or block in the source code. When program execution is started, two portions of memory are allocated: one for the stack and another for the segment dictionary which is used to reference the multiple program segments. Each entry in the segment dictionary points to a single segment indicating whether or not that segment is located in main memory.

The stack structure consists of procedure activation records. Each record contains memory allocated for a procedure's variables and descriptors, pointing to a data structure

	El'brus-1	El'brus-2	B 6700
Word Length (bits)	64+8	64+8	48+4
Clock Period (nsec)	260	47	200
CPUs	1-10	1-10	1-3
Memory Modules	4-32	4-32	1-32
Size of Module	256 Kbytes	4.5 Mbytes	96 Kbytes (16K words) 384 Kbytes (64K words)
Min main memory	1 Mbyte	18 Mbytes	384 Kbytes
Max main memory	8 Mbytes	144 Mbytes	6 Mbytes
Max memory exchange rate per processor	20.7 Mbytes/s	180 Mbytes/s	183 Mbytes/s
I/O Processors	1-4	1-4	1-3
Max peripherals per I/O processor	256	256	128
Max peripherals in configuration	1024	1024	256
Max throughput per I/O processor	3.6 Mbytes/s	30 Mbytes/s	1.67 Mbytes/s
Data Transmission Processors	1-16	1-16	1-12
Max communications lines serviced in system	2560	2560	2048
Performance (nsec, (cycles))			
Addition (single precision)	520 (2)	141 (3)	200 (1)
Multiplication (s. precision)	1300 (5)	235 (5)	2000 (10)
Division (single precision)		1081 (23)	10800 (54)

Table 4-1 Comparison of El'brus and B 6700 System Characteristics
Sources:[Dpro71; Burr72; Dpro77; Golo80; Mvke80; Timo81; Zamo85; Baba90]

such as an array. Memory portions for code segments and arrays are allocated by the operating system on demand.

Descriptors provide a level of addressing indirection which facilitates access to the same data or code structure by multiple tasks, re-enterability of code, and an economical use of stack storage. Both the Burroughs and El'brus computers were designed as multi-

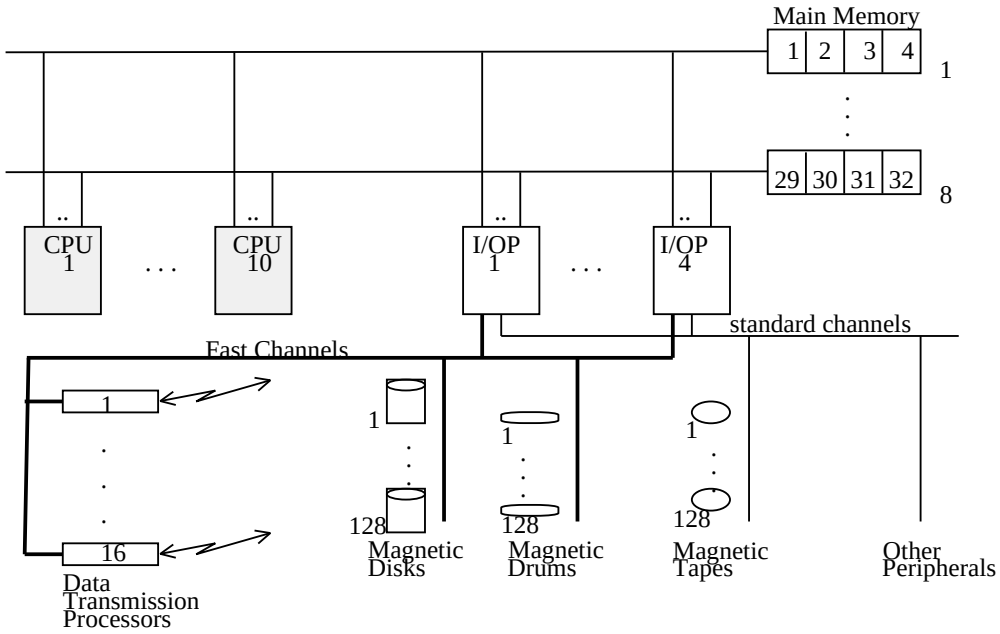


Figure 4-1 El’brus Structure
Source: [Baba90, 46]

processing, multitasking systems. Separating the code portion from the data portion makes it possible for multiple tasks to access and execute identical code, but each with its own data. Each task therefore has its own data stack and segment dictionary stack, but code can easily be shared among tasks.

The B 6700 and El’brus machines use very similar stack disciplines. The above description applies to both. A more detailed examination reveals that both use comparable special-purpose registers and control words to manage the stack. For example, each system has base-of-stack, stack limit, and top-of-stack registers. Mark stack control words placed at the start of the addressing space for each entered procedure are linked together to provide a dynamic history of procedure entry. Thirty-two display registers point to the mark stack control words, collectively defining the blocks whose addressing spaces are

global to the procedure (block) currently being entered. A return control word points to the calling procedure, indicating a return address when a block is exited.

While the Burroughs and El'brus computers have very similar approaches in their treatment of tasks and stacks, they differ a great deal in CPU construction and the underlying dynamics of execution. The B 6700 CPU consists of a 48-bit adder (arithmetic unit), an address processing unit, seven functional controllers (program, arithmetic, string, stack adjust, interrupt, transfer, and memory), and register sets. The latter include four 51-bit data registers which hold the top two stack elements (single or double precision), one current program instruction word register, one scratch register, and 48 20-bit registers which include the 32 display registers plus eight base and eight index registers [Burr72, 5-1:3]. In addition to these functional resources, the CPU has ten so-called Operator Family Controllers which group related operators into families to minimize the logic required in the processor. The families include arithmetic operators, logical operators, subroutine operators, and others [Burr72, 5-1].

The Program Controller controls the program flow. It controls the transfer of an instruction word to the Current Program Instruction Word register, decodes the syllable to be executed, and selects the appropriate Operator Family Controller to execute the instruction. A key feature is that instructions are executed sequentially in the order dictated by the compiler. There can be no over-lapped execution of arithmetic instructions because the CPU has only one adder.

In designing their CPU, the El'brus developers sought to increase performance by increasing the amount of parallelism within the CPU. The El'brus machines have ten functional units: an adder, a multiplier, a divider, a logic unit, a decimal-coded conversion unit, an operand call unit, an operand write unit, a string processing unit, a subroutine execution unit, and an indexation unit. There is some functional overlap between these

units and the controllers of the B 6700, but important differences exist. In particular, the El'brus has four times as many arithmetic units. Multiple arithmetic units had been implemented in other machines–notably the CDC 6600 in 1964 which had eight arithmetic functional units–but not in a stack-based architecture.

Stack-based architectures are not well suited to feed multiple, parallel functional units. Zero address operations assume that the necessary operands are located at the top of the stack. Clearly, only one zero address operation at a time can access operands correctly, essentially precluding the simultaneous operation of multiple functional units. In the B 6700, the stack-based approach worked well because instructions were executed sequentially; in practice, this meant that while one controller was executing an instruction the others remained idle.

The challenge for the El'brus designers was how to achieve efficient, parallel operation of the functional units within the context of a stack-based architecture. The key features of the solution to this problem were a) an internal, register-based (non-stack based) representation of the top of the stack, and b) dynamic, sequential/parallel scheduling of instructions [Baba90, 63-64, 73-75].

A so-called instruction block decodes instructions sequentially, in the order specified in the object code generated by the compiler, and places them in an instruction buffer. Rather than feeding these instructions directly into the functional units, however, the control unit converts the instructions from a zero address representation to an internal, addressed, register format. The instructions now incorporate explicit references to registers which will contain the necessary operands. While the B 6700 used two registers to store the two top elements of the stack, the El'brus computers have a 32-word buffer for the top 32 stack elements. The conversion from zero address to register-based instructions permits access to other than the top elements in the stack.

The control unit issues the instructions to the appropriate functional units at a maximum rate of two instructions per clock cycle.

A unique feature of the El'brus is that the instructions can be issued to the functional units before all the necessary operands are available. Hardware controls indicate the availability of operands, and the functional units simply wait until all operands are available before executing instructions. In effect, execution takes place in a data-flow manner with the exact order of execution depending on the order in which operands become available.

To the compiler, the El'brus looks in every way like a stack-based machine, even though the underlying implementation is not a stack.

The El'brus designers placed considerable value on ease of software development and placed a great deal of control in the hardware. In particular, all scheduling of functional units and all decisions about parallel execution of instructions are handled by the hardware. In contemporary terminology, this is called a superscalar approach to instruction-level parallelism. A distinguishing feature of a superscalar processor is that it is presented with a sequential program and tries to execute as many instructions as possible in parallel, with all scheduling decisions being handled in hardware [Fish91, 1236-1237]. Although some of the underlying ideas have been developed since the 1960s (the IBM System/360 Model 91, for example), superscalar approaches are currently very much at the forefront of RISC processor development.

A further difference between the Burroughs and El'brus approaches is the treatment of arrays. In the B 6700 all elements of arrays are accessed (indirectly) by indexing through an array descriptor. Because of this extra memory access, an extra memory cycle is often unavoidable [Orga73, 85]. There is no special hardware support for vectors [Orga73, 91].

In the El’brus, the hardware is designed to detect operations on arrays (vectors) and provide pre-fetching of array elements into local cache memory [Pent82, 57; Baba90, 56]. In the index unit there is associative memory which stores the address of the current element together with the step in memory. Only the first element must be accessed through the array descriptor; all others can be accessed directly. The associative memory can store information on up to six arrays, and the element address computation in a loop takes only one clock cycle [Baba90, 56]. Array elements for up to five loop iterations can be fetched in advance [Baba90, 14]. For this reason, implementation of vector operation in the El’brus is considerably more efficient than that in the B 6700.

The cache memory in each processor plays an important role not only in how the each El’brus CPU handles operations on arrays (vectors), but also in the multiprocessor operation of the system as a whole. The B 6700 had no cache, only a local set of special-purpose registers. In the El’brus machines, the cache consists of four distinct sections (size given for El’brus-2):

- instruction buffer (512 words) for storing instructions executed by the program. Subsequently, if instructions are executed multiple times, access time is shortened;
- stack buffer (256 words) for holding the most active (topmost) portion of the stack, which otherwise is stored in main memory;
- array buffer (256 words) for storing array elements which are processed in loops;
- associative memory for globals (1024 words) for data other than that stored in the other buffers. This includes global program variables, data descriptors, and local procedure data which does not fit in the stack buffer [Kriu80, 66-67; Baba90, 13-14].

This cache organization made it possible to incorporate a relatively large number of processors effectively into a shared-memory configuration. Cache memory basically contains copies of data or instructions stored in main memory, and in multiprocessor systems it is possible that multiple processors can each have a local copy of the same data element simultaneously. Some means are needed to make sure that all copies are identical. As the number of processors accessing main memory concurrently increases, the overhead required to maintain cache coherency can become prohibitive, limiting the number of processors which can be used effectively to a small number. In part for this reason, configurations of IBM multiprocessors such as the IBM 3033 and the IBM 3084 incorporated only up to four processors. The IBM dual-processor 3033 (introduced in 1978) used a simple store-through design in which data changed in the cache is immediately changed in main storage. The 3084 model (introduced in 1982) employed a more advanced, store-in, cache coherence scheme in which transfers to main memory could be delayed until cache elements were to be overwritten, or another processor accessed the corresponding data elements in main memory [Pras89, 217-218].

Cache coherence in the El'brus was maintained through the use of the segmented cache together with the notion of a ''critical section'' in a program. Portions of a program which access resources (data, files, peripherals) which are shared by multiple processors must be written by the programmer as critical sections in a manner which regulates simultaneous access. In the case of the El'brus, the programmer uses semaphores to synchronize access [Baba90, 17, 113-119]. A segmented cache and the use of critical sections make it possible to limit significantly the amount of overhead in achieving coherence. First, on the average, critical sections constitute only about 1% of the execution time of El'brus programs [Baba90, 17]. In other words, in the remaining 99% of the time, a given data element will not exist in more than one cache simultaneously and cache co-

herence is not a problem. Instructions in the instruction buffer are static, so copies in multiple caches will remain identical. In the El'brus, the array buffer will, as a rule, be empty during operations in critical sections. The only data elements for which incoherence is a serious problem are those contained in the associative memory for globals. Therefore, measures to reconcile the cache of multiple processors are taken only when absolutely necessary. This is one of the reasons the El'brus configuration can accommodate up to 10 processors.

Cache memory was not a new concept when the El'brus was designed. The IBM System/360 model 195 (announced in 1969) had 32 Kbytes of cache, for example. The El'brus represent one of the earlier examples of a segmented cache, however. Three of the buffer types used in the El'brus (stack buffer, instruction buffer, and associative memory buffer) were implemented on a more limited scale in the B 7700 (introduced in 1976), but the El'brus-1 design and much of the construction would have been completed by the time information about the B 7700 became available in Russia [Dpro77, 11c]. The El'brus cache coherence solution is significant because it is one of the earliest uses of a mechanism that would support a relatively large number (10) of processors in a shared memory configuration. The El'brus was one of the first general-purpose shared-memory system in the world with this number of processors to reach series production.[5]

4.3.3.3 Tags

Like the Burroughs machines, the El'brus use hardware tags to enable hardware to identify specific types of data and instructions. The Burroughs machines used three-bit tags to identify single/double precision operands, data descriptors, and a number of con-

[5] Others which share this honor were some bus-connected minisupercomputers with multiple processors sharing multiple memory units which appeared during the early-mid 1980s. For example, the Sequent Balance 8000, with up to 12 processing elements was first delivered in 1984 [Hock88, 46]. This pre-dates the El'brus-2, but is later than the El'brus-1.

trol words. In their zeal to place as much control in hardware as possible, El'brus designers developed an elaborate set of hardware tags. Using six-bit tags, they were able to distinguish between half/single/double precision operands, whole/real numbers, empty/full words, labels (including such specialized labels as ''privileged label without block of external interrupts'' and ''normal label without block of address information write''), semaphores, control words, and others [Zamo85, 129].

One major purpose of the tags was simplification of programming. If the functional units could distinguish between real and integer operands, they could be designed to adapt themselves to computation on either. There would be no need for separate scalar and floating-point units. The El'brus could in effect implement dynamic typing of data which is useful in, for example, building operating systems in which the precise type of the operands may not be known ahead of time. This capability was one of the reasons why the entire El'brus operating system could be written in a high-level language.[6]

Another purpose of the tags was error detection. Hardware could detect such errors as attempted arithmetic operations on control words, for example. Tags could also be used for memory protection, for restricting writes to specific types of data [Baba90, 11, 54-55].

Tags were not an invention of the El'brus designers. They had been a key element in Iliffe's Basic Machine [Ilif68] and the Burroughs machines which it inspired. The El'brus pushed these ideas to a new level of detail and complexity.

---

[6] Other hardware features played a role as well. One of the main problems was being able to use high-level languages for specifc parts of the system software, such as memory allocation and process switching. To do this, code transparency and object code predictability are required. The El'brus uses special very-high-level hardware to accomplish this, e.g., process switching can be programmed as a sequence of assignment statements performing clearly defined actions on special hardware registers. The operating system mechanisms were defined first, and the hardware design and instruction set were tailored accordingly.

4.3.3.4 Memory

As in the Burroughs machines, El'brus main memory is shared among all processors and is modular in design. The B 6700 main memory was initially implemented as 1-64 memory modules of 16K words (48+4 bits) each for a maximum memory size of 1024K words. Later versions of this machine incorporated a mix of 16K and 64K word modules, although the maximum memory size remained 1024K words [Dpro71, 11b; Burr72, 1-8; Dpro77, 11b]. Data could be interleaved across modules.

The El'brus contained significantly more memory. The El'brus-1 contains 4-32 modules of 256 Kbytes each [Zamo85, 145], while the El'brus-2 contains modules of 2M words (4.5 Mbytes) for a total of 16M words (144 Mbytes) [Baba90, 47,78]. The El'brus memory is organized as a hierarchy, with a memory section (stored in a single cabinet) consisting of four memory modules; each memory module consists of up to 32 blocks, each containing 16K words. Interleaving is possible at multiple levels: between sections, between modules within a section, and within the individual modules. Up to four words can be read from each memory module in one cycle. The maximum throughput per section in the El'brus-2 is 450 Mbytes/s, although the maximum data exchange rate with each processor is 180 Mbytes/s [Baba90, 78].

The memory management schemes of the B 6700 and El'brus are, at the general level, very similar. Both employ segmentation. Memory is organized into variable length segments which reflected the logical divisions of a program determined by the compiler. Corresponding to the logical division of a program, segments can be coded independently, given different levels of protection, and shared among processes.

In the B 6700, segments were moved between main and virtual storage as complete segments [Orga73, 90]. Arrays were an exception. They could be stored in main memory in groups of 256 words each, bounded on both ends by memory link words [Burr72, 5-7].

The El'brus treats program segments differently from data and array segments. The former are treated just as in the B 6700 and moved in and out of main memory in one piece. Data and arrays of constants, on the other hand, are organized into pages of 512 words each [Zamo85, 122]. This approach is similar to that of the B 6700, except the paging principle is applied more broadly, to data, as well as to arrays. In both cases, this allows the processor to handle data sets which exceed the amount of physical memory available to a process. However, the El'brus approach uses memory more efficiently and allows faster process swapping.[7]

The El'brus employ a more contemporary type of virtual memory. In the Burroughs computers, addressing was limited to 20 bits, or 2^{20} words, the maximum size of physical memory in both the B 6700 and B 7700. Segments were moved between main and secondary storage and their presence in main memory was indicated by a ''presence bit'' in their descriptor which remained in main memory during the run of the process. There was no concept of a true virtual memory space which was larger than the total amount of physical memory; the descriptors contained only physical addresses [Burr72, 1-8; Orga73, 17].

The El'brus machines use a similar, 20-bit addressing scheme for program segments, but 32-bit addressing was used for segments of data and arrays of constants. This provided a virtual memory space of 2^{32} bytes (4 Gigabytes). These segments were moved between virtual and physical storage using a paging mechanism which used paging tables stored in an associative page memory unit to convert between virtual and physical addresses. Virtual addresses consist of a page number and an offset within the page [Zamo85, 119, 122; Baba90, 66]

[7] In the B 6700 process swapping required scanning the process stack for all descriptors to arrays used by the current process.

4.3.4 Reliability

Building computers for a variety of applications, including real-time, both the Burroughs and El'brus engineers placed high priority on system reliability. This was particularly true for the El'brus engineers who had to struggle with a notoriously unreliable component base. The modular design of both machines provided redundancy of all system resources, and the ability for each unit (CPU, memory module, I/O processor, etc.) to operate independently of the others. Within this modular framework, hardware and software mechanisms to increase reliability were implemented on a number of levels.

Both the Burroughs and El'brus computers incorporated fail-soft features in which systems routines regularly check systems modules and, when a module fails, automatically remove the troublesome device from operation and reassign its functions to other modules without operator intervention [Dpro77, 11e; Zamo85, 117; Baba90, 97]. In the El'brus computers there are actually two levels of recovery–''soft restart'' and ''hard restart''. In the first, processes are interrupted and restarted on processors; in the second, a unit is logically removed from the configuration [Burt87, 20].

To improve the reliability of individual modules, the El'brus computers use Hamming code error detection and correction in memory modules and re-read or re-write to memory [Burt87, 19; Baba90, 80]. Automatic instruction retry (up to 16 times) is used in the CPU. The B 6700 did not have the latter feature, which was introduced in the B 7700 [Dpro77, 11m]. In spite of such features, the reliability of individual modules remained low in comparison with machines such as the Cray which had a mean time between failure of hundreds or thousands of hours. An official analysis of the El'brus-2 reliability made at the time of state testing of the El'brus-2 in 1985 gives the figures shown in Table 4-2.

Device	Mean Time Between Failures (hours)	Mean Time to Repair (hours)
CPU	92	0.6
Main Memory	1263	0.29
I/O Processor	565	0.3

Table 4-2 El’brus-2 Reliability Characteristics
Sources:[Burt87, 18]

It is significant to note, however, that although the mean time to failure is low, especially for the CPUs, the mean time to repair is also well under an hour. As a rule, therefore, an El’brus configuration under constant surveillance by trained technicians could be kept running for long periods of time. As modules failed, they would be switched out by the system itself, quickly repaired by technicians, and switched back into operation. El’brus operation was, as a result, very labor intensive. El’brus computers at the most important installations (such as at the All-Union Scientific Research Institute of Experimental Physics (VNIIEF) in Arzamas-16) did have skilled on-site technicians who could keep the machine running nearly all the time. Installations which did not have a skilled maintenance crew could not expect to keep the system up for long periods of time.

If a CPU failed, a process was restarted from a checkpoint on another processor. The practical impact of this was that jobs which had long execution times between checkpoints (i.e. if extensive computations were carried out on a very large data set), the danger that the process would fail in the middle and have to be restarted was very real. According to one user, ‘‘running jobs on the machine was nerve-wracking.’’

One of the main reasons for the lack of reliability were the multi-chip modules (described below) which were used in the El’brus-2 processor from 1985-1989. When gate

arrays replaced multi-chip modules in the El'brus-2 construction in 1989, reliability improved dramatically. According to some reports, the reliability of the CPU increased to 240-500 hours mean time between failures.

4.3.5 Performance

The performance figures given for the El'brus computers are nearly always the same: In full configurations, 12-15 MIPS for the El'brus-1, and 120-125 MIPS for the El'brus-2 [Tass78; Golo80; Mvke80; Ivan87; Baba90, 47]. These figures represent El'brus performance on a Gibson-3 mix (a mix of a variety of instructions used to measure performance on IBM and ES mainframe computers), rather than the theoretical peak performance [Baba90, 50]. According to V. S. Burtsev, the theoretical peak performance figures were never emphasized (or published in Russia, to our knowledge) because he personally did not believe in using peak performance figures to advertise his machines. Besides a personal reluctance to use ''unrealistic'' theoretical peak performance figures (an opinion voiced by many in the West), Burtsev was using a means of quantifying performance traditional for the ES series (and for IBM machines). The real competition to the El'brus computers came from the high-end ES mainframes which were the only real alternative for organizations seeking to acquire significant general-purpose computing power. Seeking a basis for comparison, policy makers perhaps preferred to have performance of the two families of machines based on the same kinds of tests. Both a single-processor El'brus-2 and an ES-1066 have official performances of 12.5 MIPS on a Gibson-3 mix [Dani84; Ecot85; Vdnk86]. In a direct test on a large physics problem, however, the single-processor El'brus-2 ran 2.5 times faster than the ES-1066 on 32-bit operands, and 2.8 times faster on 64-bit operands [Baba90, 15]. The theoretical peak performance (TPP) of the El'brus-2 was calculated independently and published in [Doro92]. Taking into account the number of clock cycles to compute results in each of the functional units which

could perform floating-point operations, the authors computed a theoretical peak performance of 9.4 Mflops per processor, or 94 Mflops for a 10-processor configuration Doro92[Doro92, 5].

A few other performance reports (not independently verified) have been published. In 1988, S. V. Kalin ran the 24 Livermore Fortan Kernels on a single El'brus-2 CPU and measured a harmonic mean[8] of 2.7 Mflops [Baba90, 50]. In comparison, Pfeiffer, et al. report a harmonic mean of 15.26 Mflops for the LFK on a single-processor Cray X-MP with a clock cycle of 9.5 nsec and a theoretical peak performance of 210 Mflops [Pfei90, 140].

These figures appear to point to a strength of the El'brus design: average performance on a variety of applications. Although the peak performance of the Cray X-MP processor is over 20 times that of the El'brus-2 CPU, the harmonic mean is only 5.7 times greater, a ratio only slightly larger than the ratio of the clock periods of the two machines. The El'brus-2 performance on nicely vectorizable problems is significantly lower than that of the Cray X-MP, but it performs rather well, relative to its clock period, when a variety of programs–not all of them vectorizable–are run.

In retrospect, however, the designers of the El'brus underestimated the importance of vector-pipelining for achieving high performance on vectorizable problems. If an El'brus processor were designed on vector-pipeline principles, producing two results (i.e., add and multiply as in a Cray) it would have had a theoretical peak performance of 42.5

[8] The harmonic mean (unweighted in this case) is computed as

$$\frac{1}{\left(\frac{1}{I}\right)\sum\frac{1}{R_i}}$$

where I = number of programs, and R_i = execution rate of program i. The harmonic mean, taking into account the proportion of a task completed at a given rate, gives a ''truer'' indication of average performance than an arithmetic mean [Worl84, 124-125].

Mflops. In a 10-processor configuration, the TPP would be 425 Mflops. A possible reason for the lack of vector-pipelining is that the basic design work on the machine was carried out before the impact of the Cray-1 introduction on supercomputer design was felt. It is also possible that the relatively good performance of the El'brus-2 on mixed tasks made the issue less pressing.

There are two basic bottlenecks in El'brus-2 performance. The first is the lack of pipelining in the functional units so that each functional unit uses 3+ cycles to generate a result. Second, the instruction issue mechanism is only able to issue two or fewer operations per cycle. Given the rate at which the functional units operate, this is not necessarily a bottleneck in real applications. It could be, if the functional units were pipelined.

4.3.6 Differences Between the El'brus-1 and El'brus-2

The El'brus-1 and El'brus-2 are virtually identical in their basic design. They differ by an order of magnitude in their Gibson-3 (not theoretical peak) performance, however. The primary reason for this was the difference in the component technology. First, the El'brus-1 uses TTL technology, while the El'brus-2 uses ECL components. The clock period could therefore be decreased by a factor of more than five. This alone does not explain the factor of ten difference in performance. Performance also depended on the rate at which data could be moved from main memory to the CPUs. While the El'brus-1 used ferrite core memory, the El'brus-2 uses semiconductor memory [Golo80; Mvke80; Baba90, 78]. As a result, the data throughput from main memory to each processor was nine times greater in the El'brus-2, making it a more balanced system.

Another difference was the treatment of arrays. As has been mentioned, the El'brus-2 has a hardware mechanism to prefetch array elements, enabling it to perform considerably better on vector computations than the El'brus-1.

As a result of these and other changes, the instruction sets of the El'brus-1 and El'brus-2 were slightly different. In computers where much systems software is coded in assembler, differences in the instruction set typically mean that machines are not software compatible. Porting code from one machine to another requires not only recompilation, but modification of the source code. The decision to code all El'brus software, including the operating system, in a high-level language meant that all software could be moved from one platform to another simply by recompiling code. No modifications were necessary. This feature, the lack of assembly-level code to ''fix'' the instruction set was to give the El'brus designers considerable freedom of movement in later years. It proved beneficial in the transition from the El'brus-1 to the El'brus-2, and would in later years make it possible to expand the El'brus line into a variety of architectural approaches (mentioned below, and in chapter 3) which bore little resemblance to stack-based machines.

4.4 The El'brus in the Soviet context

4.4.1 The Long Road from Conception to Production

The El'brus project was begun around 1970. Most of the design work was done between 1970 and 1973, when the draft design *(ekskiznyy proyekt)* was completed [Pent82, 10]. During these years, designers had access to information on the B 6700, but only about the instruction set and the system structure block diagrams. During 1975-1976 they obtained more detailed functional descriptions of the Burroughs which led to some modifications of the design of the hardware and programming language. In 1977, a B 6700 was sold to the oil industry and ITMVT designers were able to examine it in detail. Because it had never been a tradition at ITMVT to develop functional duplicates of Western machines (à la the ES program) and much information about the B 6700 hardware was lacking at crucial design stages, the El'brus hardware was designed from scratch.

The first El'brus-1 prototype became operational around the end of 1977, and the first parts of the operating system were run on it at the end of 1978 [Golo80; Golo86]. It underwent state testing in 1979, and was accepted by a state commission in 1980 [Golo86, 87].

El'brus software was the topic of a conference held in Novosibirsk in 1976, and in 1978 the machine was touted in a prominent article in Pravda [Baba77; Burt78].

The El'brus-2 was developed in parallel with the El'brus-1, but on a very different component base. The draft design was completed in 1978. The first El'brus-2 prototypes were running in 1984 and a two-processor version underwent state testing in 1985. In February, 1986 a 10-processor unit was brought on-line. Full series production began in 1987 and continued through the end of 1992.

The El'brus computers had a long and painful birth. Shortly after the Pravda announcement, the Estonian Academy of Sciences announced that it would receive an El'brus-1 by the end of 1980 to be used in a collective-use computer center to service a number of Academy institutes [Vyrk78]. The Institute of Cybernetics was to house the machine and develop a time-sharing system to give other institutes access [Vyrk79]. These estimates proved to be wildly optimistic. By the end of 1981, some, but not all, portions of the El'brus-1 had arrived in Tallinn [Favo81; Gudi82]. By 1982, the complete machine was expected by 1984 [Aben82]. A dual-processor machine was finally running in 1986, thanks to the purchase of a cooling station from the Finns. Once installed, it was equipped only with about a dozen disk drives and drums (with a total capacity of about 70+ Mbytes (!)), had low reliability (especially when multiple user jobs were running) and was used rather infrequently.

While there were rumors that many other customers for the El'brus-1 had their orders canceled without explanation [Harv83] and that the El'brus program as a whole was in

trouble, the Estonian machine was perhaps an extreme case. Primary El'brus customers with good maintenance support report using the El'brus-1 with some success. The Estonian experience did nothing to help the reputation of the El'brus program, however. According to G. G. Ryabov, this machine was shipped to Tallinn ''practically untested'' and without the equipment and support necessary to get it running quickly. This bad experience, coupled with the long delays in getting the machine into production and the general lack of involvement of the broader Academic computing community gave the program a bad reputation. Although the machine was praised publicly, in closed discussions the machine was criticized by the Academic and industrial communities. Growing criticism over the progress being made, complicated by personality conflicts within ITMVT and disagreements over the future course of the program, brought matters to a head in 1984. Burtsev was removed as director of ITMVT and replaced by G. G. Ryabov.

Burtsev's removal did not necessarily dampen the criticism of the machine. Only during the late 1980s were significant number of El'brus-2 processors manufactured. Between 1985 and 1989, approximately 100 El'brus-2 processors were reportedly manufactured. Of these, half were used in five 10-processor installations. By 1992, over 200 processors had been manufactured.

Even then the machine, which had been developed principally for military customers, was inaccessible to large segments of the traditional (BESM-6) user community. Nearly all installations were highly restricted. These included weapons designers at VNIIEF in the closed city of Arzamas-16, Mission Control in Moscow, rocket designers at the Energiya Scientific-Production Association, and others [Laza91].

In 1989 some efforts were made to remedy this situation through the creation of the Collective Use Computing Center of the USSR Academy of Sciences, housed in the new

Presidium of the Academy of Sciences building in Moscow, which currently contains an eight-processor El'brus-2 configuration [Veli89]. This was, however, too little, too late.

In comparison to the BESM-6, which was very reliable by Soviet standards and well understood by those who used it, the El'brus was considered very unreliable and hard to maintain, a fact which has been acknowledged by the designers themselves. This was especially true for those users who did not have strong maintenance support. Only in 1989 when gate arrays replaced the unreliable multi-chip modules did the machine become acceptably reliable.

4.4.2 The Role of the El'brus in the Soviet Computer Industry

Approximately fifteen years elapsed from the start of El'brus-2 design work to its introduction into series production. Modest numbers of El'brus-1 computers were built in the interim, but the real goal, from the start, was the ECL-based El'brus-2. We observe that very long development times are a common occurrence in Soviet high-performance computing, but why is this so? In the case of the El'brus, many of the delaying factors can be traced to the nature of the technology, the structure of the infrastructure supporting research and development, the quality of inputs from, in particular, Minelektronprom, and, to a lesser degree, the relationship between ITMVT and the El'brus factories.

The El'brus computers were a driving force for the entire Soviet computer industry. The mission of ITMVT, both held by ITMVT engineers and imposed by policy makers in Minradioprom and the VPK, was to build the fastest machine possible given the available technology. Development involved a balancing act. One the one hand, a working machine had to be produced, so the capabilities of the supporting industries had to be taken into close account. On the other hand, high requirements were needed to force the rest of the industry to raise their technological level. Once production of a new technology (such

as components) was assimilated, it was often incorporated into other computers, such as the ES mainframes.

Building the ‘‘fastest machine possible’’ generally implied ‘‘at the world level.’’ Building machines at the ‘‘world level’’ required the development of new components, cables, power supplies, cooling systems, printed-circuit boards, connectors, computer-aided design tools, manufacturing facilities, etc. for each generation. The El’brus computers pushed the boundaries of many technologies simultaneously. Of the over one-hundred million rubles spent annually on high-performance projects spearheaded by ITMVT, only 25-30% stayed at ITMVT. The rest went to fund development of supporting technologies in other institutes, some of which were in other ministries. All together, the El’brus computers involved hundreds of enterprises which manufactured everything from cabinets to glass bulbs to printed-circuit boards to components. In most cases, new products had to be developed and manufactured.

It is perhaps the case that without a specific, high-profile project like the El’brus there would have been little impetus for the supporting industries to improve their technologies. Pushing many technological boundaries simultaneously had negative effects on the time needed to develop the end product, however. First, a great amount of time and effort was needed to get each factory to assimilate production of new items. Second, so many parts of the computer were little more than prototypes themselves that the debugging process was greatly extended.

Getting each of the factories to produce what was needed was ‘‘an absolute nightmare.’’ First, dealing with each factory involved a long bureaucratic trail. The director of the ITMVT, Burtsev and later Ryabov, would have to negotiate at each level of the economic management structure, from the factory up through ministerial section heads to the minister himself and, on many occasions, to the Central Committee of the Communist

Party. The greater the ‘‘administrative distance’’ between ITMVT and a particular factory, the more people were involved in this chain, the longer the negotiations took, and the weaker the feedback and accountability between ITMVT and the factory. Particularly time consuming was the interaction with entities in other ministries, in Minelektronprom, in particular, but even within Minradioprom negotiating the hierarchy was problematic. At each level one had to deal with individuals who had a monopoly position, and their own interests [Burt92, 9].

Second, although the factories were subordinated to the ministries, they did have considerable *de facto* influence over production. Factories were typically heavily loaded with orders and often used this as an excuse for not completing a given order on time. Under these conditions, factories tended to favor manufacture of simpler products already assimilated into production. Re-tooling for a new product cost much time, effort, and required the termination of some existing production, often at the expense of missing other production targets. Because each factory also had to depend on many other suppliers, assimilation of new production was very stressful. If the factory would not get any more money for a new product than for an old product for which there were still orders, it had little incentive to go through the trouble. Furthermore, since the Plan indicators often specified number of items to manufacture, quality was a secondary concern [Berl76]. The low quality of parts was a continuing thorn in the side of the El’brus designers; although they expended a great amount of effort and hardware designing the machine to be reliable, they were never able to completely compensate for the low reliability of the components.

For each new product, ITMVT faced factories disinclined to assimilate new technologies. Overcoming this resistance involved working the higher levels of bureaucracy to bring sufficient pressure on the factories that they would develop and incorporate El’brus

parts into production. The monopolistic nature of the economy further complicated matters. A very high percentage of the parts of an El'brus are single-source; only one factory manufactures them. El'brus designers and manufacturers had little choice in the selection of specific factories for specific products. If a factory was slow in producing a given item, there were no alternatives to waiting, or putting pressure on the factory to move more quickly.

In principle, the Military-Industrial Commission (VPK), which had inter-ministerial oversight over computing, should have been in a position to facilitate the interaction. In practice, according to some intimately involved with policy making, the VPK would at one time push for speeded development of one system, then later for another. The El'brus had to compete for attention and, when pressure was applied to speed up development of something else, El'brus-related efforts languished.

4.4.3 The El'brus Component Base

While each new item which had to be assimilated into production and manufactured played a role in lengthening the development time, components were the most problematic. From the beginning of the El'brus program around 1970, the plans were to build the El'brus using ECL 100 series (later called the IS-100) components. These were functional duplicates of Motorola's MECL 10K chips. The first experimental units became available to designers in 1972-1973, but series production was far enough in the future that designers had to change their plans. The El'brus-1, therefore, was built using low TTL chips–the Logika-2 and 133 series, reportedly functional duplicates of Texas Instruments chips [Shaw85]–rather than ECL [Grub80; Mvke80].

In 1972, ITMVT initiated the development of a multi-chip module technology and pushed Minelektronprom to work on implementation as well. The modules consisted of 8-10 IS-100 chips on a single substrate. Theoretically, this would make it possible to in-

crease the speed of connections to individual chips. The chips themselves were manufactured by Minelektronprom, but ITMVT established a facility to place multiple chips on a single carrier. This was later done at the Minelektronprom plant. The first units became available in 1976, and development of the El'brus-2 processor began. Approximately half its logic was implemented as individual IS-100 series chips and the other half as multi-chip modules. This unit was built during 1979-1980 and debugging began at ITMVT around 1981.[9]

Both the IS-100 and multi-chip modules had reliability problems which compounded debugging difficulties and compromised the over-all reliability of the El'brus systems. By one estimate, 1-1.5 years were added to the debugging stage solely because powering a processor on and off had a tendency to break down the chips. Another source states that the system debugging was stretched out 2-3 times because of the poor components. On the order of five multi-chips modules reportedly were replaced per day at the beginning. Resistor chips used in the initial models were also quite unreliable. Several tens of El'brus-2 computers were built using the multi-chip modules. Given their low reliability, however, designers tried to switch to gate arrays as quickly as possible, eventually phasing out the multi-chip modules.

ECL gate arrays with 100-200 gates per chip were developed in 1983-1984 for the El'brus-2 and the first El'brus-2 processor using these (experimental) chips was completed in 1986. It took several years for Minelektronprom to assimilate mass production of the chips at a satisfactory level, however. Several El'brus-2 computers were manufactured at ZEMZ during 1985, but they were unworkable. Reportedly, the ceramic frames of the gate arrays would crack because the cooling system did not adequately dissipate the heat the chips generated. Efforts to improve the design of the gate arrays stretched

[9] Another informed source states that the first multi-chip modules became available in 1977-1978.

out development time by a year at least. Delivery of both the first experimental El'brus-2 (1984-1986) and second workable El'brus-2 based on gate arrays (1986-1987) were both impacted by the delay. Only in 1989 did the gate arrays replace the multi-chip modules in series produced El'brus-2 processors.

ITMVT served as a driving force for the electronics industry in more ways than one. First, development of the IS-100, multi-chip modules, and gate arrays was initiated by ITMVT for its high performance computers. Once manufacturing had been assimilated, the chips were used more broadly in the computer industry, in the ES mainframes in particular. For example, the ES-1035, ES-1036, ES-1061, ES-1065, ES-1066 and a number of parallel processors based on ES technology used ECL chips called the IS-500 or K500 series which were essentially the same as the IS-100 [Anto81; K86; Lomo87; Torg88]. The main difference, reportedly, is that the IS-100 have through-hole pinouts, while the IS-500 are surface mount. The I300B gate arrays, developed for the MKP and El'brus-3 (described below), have been used in a re-engineering of the ES-1066, called the ES-1087. ITMVT also used the components for its line of BESM-6 compatible systems.

Second, to get the needed chips, ITMVT felt it necessary to push Minelektronprom by actively participating in the design of the chips themselves. A rather close working relationship was established between ITMVT and engineers at Minelektronprom design facilities during the mid-1980s, especially after Ryabov became director of ITMVT [Kova87, 2]. ITMVT had a division devoted to the development of CAD systems for chip and printed-circuit board design, and between 1984 and 1988 ITMVT even developed a special-purpose system for testing chips for Minelektronprom. ITMVT engineers would take the basic technical parameters of the chips and design them using a CAD system which, while developed at ITMVT, was used by Minelektronprom. Simulation tests were also carried out at ITMVT. During this time there would be frequent interaction between

ITMVT and Minelektronprom. Close cooperation and exchange of technology was necessary to ensure that the design could be manufactured by the factory. The design, together with a set of tests developed at ITMVT, would be sent to Minelektronprom which would manufacture a chip tested jointly by ITMVT and Minelektronprom. ITMVT had to approve the first chip, which would then be put into series production.

The fact that ITMVT and chip designers in Minelektronprom had good relationships did not mean that it was easy to get the chip manufacturing factories (a different group of decision-makers) to produce the chips, for reasons mentioned above. Minelektronprom was notorious for manufacturing a few prototype chips which were approved by the state commissions, but then have great difficulties manufacturing large quantities of reliable chips. One El'brus factory representative has commented that the battle to get Minelektronprom to manufacture all the chips they needed was an annual affair, often reaching the level of the ministers who would sign an agreement among themselves indicating the amount of chips individual factories would receive.

Thus, the development of computers drove the development of the component base rather than vice-versa, as has become the predominant practice in the West, particularly during the 1980s and 1990s.

The complexity of the El'brus architecture compounded the problems with the component base. Early in the design phase many expressed concern that the machine was too complex for the available technology. These fears were realized. The complexity of the logic required much specialized hardware and levels of IC integration which placed great demands on the component base. Whether the architecture was too complex or the component base too weak is a largely academic question; the fact is that there was a mismatch between the architecture and what could be supported by the component base which aggravated the reliability problems.

This lesson was not lost on ITMVT developers, however. The El'brus-3 and MKP appear to have designs which, in relation to the capabilities of the components, are relatively less complex than the El'brus-2 [Baba89b, 130; Supe91, 17]. Whether they are simple enough for the components at hand is still an open question, however.

4.4.4 Peripheral Storage

One area in which ITMVT has not been an industry driver is external storage. The El'brus machines have been equipped with magnetic drums designed for the BESM-6, but the magnetic disks and tapes have nearly always been those designed for the ES mainframes, using the IBM-compatible data channels. The El'brus-1 could accommodate up to 32 disks and 32 drums per I/O processor, with any four of them operating concurrently (on the four channels making up the fast channel block) [Golo80; Kriu80; Mvke80]. The maximum exchange rate with peripheral storage was 4 Mbytes/s per I/O processor. The ES-5056 magnetic disks had a storage capacity of 7.25 Mbytes each [Mvke80; Es76; Kezl86]. As disks with greater capacity were developed for the ES mainframes, they were incorporated into El'brus configurations. Although there is every reason to suspect that the most important El'brus users received the best drives Soviet and Eastern European (chiefly Bulgarian) industry had to offer, disk drives advanced slowly. The largest capacity disks manufactured in series production by Eastern Bloc industry, the 317.5 Mbyte ES-5063, did not become available until 1984-1986 or later [Dani84; Dani86]. With a data transfer rate of 1.198 Mbytes/s, the disks underutilized the El'brus-2 fast data channels which each had a transmission rate of 4 Mbytes/s [Baba90, 90]. Following the breakdown of trade relations with Eastern Bloc countries in 1991 (with Bulgaria in particular) shortages of disks became a severe problem for many users. As late as 1991 there were El'brus installations with (significant numbers of) 100 Mbyte disks only.

Although ITMVT did reportedly on occasion try to press Minradioprom for the development of higher-capacity disks, its efforts were unsuccessful.

4.4.5 Relationship with Factories

ITMVT has had long-standing relationships with the small number of factories manufacturing its machines. The Moscow Calculating-Analytic Machines (SAM) Plant manufactured the general-purpose systems designed for mass-use. These include the BESM-6 and related machines such as the AS-6, the SVS-1, and El'brus-B. The MKP, developed by many of the same individuals who worked on the BESM-6 and AS-6 has also been constructed here. The El'brus computers were manufactured primarily at the Zagorsk Electro-Mechanical Factory (ZEMZ) outside of Moscow, with the I/O processors, telecommunications processors, and some disk drives manufactured at SAM [Supe91b, 15]. Some El'brus subsystems such as the memory modules were manufactured at the Penza Computational Electronic Machines Plant (VEM). During the 1980s a factory in Tashkent was also retooled to manufacture El'brus machines.

ITMVT worked closely with the factories, especially with their associated design bureaus. In general, re-tooling a factory was a very slow process for reasons mentioned above. By establishing a close working relationship early on in a given project, ITMVT could improve communication with the factory and help ensure that the machines were being constructed in a manner which suited the plant's capabilities.

ITMVT and its few primary production factories were dependent on each other. ITMVT depended on the factories' manufacturing facilities. The factories in turn depended on ITMVT for the development of new technologies: the machines, CAD systems, documentation standards, etc. By design, the factory and ITMVT worked with much the same technology. It was therefore possible, when necessary, for the factory to make small changes–redesign a PCB, for example–without the involvement of ITMVT.

The factory was also free to use the technology for development and production of products which had little to do with ITMVT. At the same time, the factory had little hope of advancing technologically without ITMVT's help. One exception to this symmetry, however, was printed-circuit board manufacturing. Around 1988, ITMVT had been able to acquire technology to manufacture 20-layer PCBs through a joint venture with the Swiss firm Rode, Inc. The same technology was to have been installed at the factories, but government hard-currency allocations were reduced before this could be done. Reportedly, the SAM plant has recently acquired equipment capable of manufacturing 20-layer PCBs (those used in the MKP and the El'brus-3) through a joint effort involving a German firm.

As soon as the work on concrete logic circuits began, ITMVT incorporated specialists from the factory design bureau into the development process. They served as liaisons to the series production factory. Interaction between ITMVT and the design bureau engineers took place nearly on a daily basis. The latter were intimately involved in prototype construction and the development of the series production documentation. Two basic tasks were accomplished concurrently: construction of the prototype, and creation of new production technology for the series production plant. The latter was nearly always necessary, since ITMVT machines were continually pushing existing technological boundaries.

Although a close working relationship with the design bureau greatly facilitated getting machines into production, it did not smooth the ITMVT-factory relationship completely. The design bureaus and the series production plants viewed a new machine differently. The engineers involved in prototype construction were usually enthusiastic about the work which they considered very interesting and personally challenging. For the series production plant, however, the introduction of new projects was a process filled

with uncertainty and delay. Before *perestroika,* the factories had more than enough orders to occupy existing capacity; assimilating new projects tied up such capacity and reduced the total output of the plant.

The technology itself complicated the relationship. The El'brus computers were difficult to build. They were also a moving target. As prototypes were built, the design and construction were frequently changed. Even after the prototypes were completed, modifications (such as the use of new kinds of chips) were made, requiring changes to the series production facilities. Furthermore, these changes had to be incorporated into machines which had already been manufactured, as well as to subsequent units. Although ITMVT directors were usually able to get additional funding allocated to the principal customers to cover this cost, the series production factories found such changes undesirable.

4.4.6 ITMVT Structure

ITMVT had a traditional institute-division-subdivision-laboratory structure. The basic hierarchy and the corresponding titles, pay levels, and administrative responsibilities were well defined by regulations in effect for applied-science institutes throughout the Soviet Union. The specific structure of ITMVT reflected the technical tasks associated with building ITMVT systems. Divisions were devoted to specific machines, and the structures within a division were devoted to the constituent hardware and software tasks, etc. Thanks to the long-term and stable nature of systems development, ITMVT's structure remained rather constant throughout the 1970s and 1980s.

It was difficult for people to be transferred from one part of the organization to another, but communication throughout the institute was quite fluid. One of the traditions established when S. A. Lebedev was director was that the various teams should have the freedom to interact with each other and share ideas. Lebedev created a collegial work en-

vironment. The hierarchy rather strictly determined which individuals could write articles about the machines, however.

Prior to 1985, ITMVT incorporated of a number of technical divisions. Two of these were devoted to the high-performance computing systems. The first, headed by A. A. Sokolov, worked on the MKP. This is the same division and, to a large extent, the same team that worked on the BESM-6 and the AS-6. This division also had a subdivision working on BESM-6 compatible systems called the SVS-1 and the El'brus-B. The El'brus computers were developed in a second division. This division consisted of hardware and software subdivisions.

ITMVT's research strategy was to develop two lines of high-performance systems concurrently. This practice originated in the 1960s when one team developed general-purpose systems (BESM-6 and its predecessors) and another developed special-purpose systems for the military. When the El'brus program was initiated, both teams worked on general-purpose systems. The strategy served two basic purposes. First, it increased the chances that one system or another would be put into series production. Second, it created some in-house competition for resources and recognition between teams which encouraged greater productivity and creativity.

Prior to 1985, Babayan was formally the head of the El'brus software subdivision, although in fact he was in charge of both subdivisions.[10] Functionally, the software subdivision had two components: software development tools, and operating systems. Compil-

[10]There were a number of instances in the El'brus division in which according to the formal structure (titles) two or more individuals were at peer levels (heads of laboratories, heads of subdivisions, etc.). In practice, however, one was the superior of the others. Such situations arose because of the regulations regarding wages, which specified how much money an individual with a given title could earn. An individual could not be given a raise beyond that specified in the regulations without giving him a new title. So, to increase an individual's wages, he would be officially made the head of an organizational unit at the same level as his immediate boss; unofficially, however, he remained subordinate to his boss. In other cases, a single subdivision could be broken into two pieces, effectively reducing the influence of the original division head. The reasons for such a change could just as easily be political as technical.

ers and other tools were incorporated into a subdivision separate from the operating systems subdivision because programming-language related features played an important part in the development of systems software. The Novosibirsk subsidiary of ITMVT which developed conventional language compilers like FORTRAN was actually subordinated to the software development tools subdivision, however.

A third division worked on small-scale special-purpose and embedded systems for the military.

A fourth division, under Ye. A Krivosheyev, worked on a specialized vector-pipelined processor for El'brus-2 configurations [Burt85].

Two other divisions supported the systems divisions. G. G. Ryabov headed the division of computer-aided design (CAD) systems; F. P. Galetskiy headed the multilayer PCBs and subassemblies division. The latter had worked on the multi-chip modules. Another division worked on main memory systems.

The main memory and PCBs divisions reflected V. S. Burtsev's efforts to compensate for an insufficiently responsive Ministry of the Electronics Industry. He hoped that by creating divisions to work on the design and development of chips and printed-circuit boards that he could provide his systems developers with components earlier in the systems life-cycle, and get them into production in Minelektronprom more quickly.

When G. G. Ryabov became director, he made a number of changes to the organizational structure. First, B. A. Babayan was made the formal head of the El'brus division. Second, Galetskiy's PCB technology division was strengthened through increases in staff and equipment. Third, although he did not terminate the division working on the vector processor (discussed in section 3.2.3.1), he did not support its work and eventually the division dissolved and the vector processor project ended. Fourth, he transformed the memory and electronic components divisions into small units which were to serve as a

bridge between ITMVT and Minelektronprom, hoping that they would lead to improved relationships between ITMVT and Minelektronprom. As we have described, these changes did improve the interaction between ITMVT and some of the design groups within Minelektronprom, but they still did not result in the timely delivery of quality components by the electronics ministry.

In addition to the technical divisions, ITMVT had two administrative councils. One council oversaw dissertation defenses and the awarding of Candidate of Science and Doctor of Science degrees, and made recommendations of appointments to scientific positions. The scientific-technical council, consisting of the director, his deputies, the division heads, and a Party representative, advised the directorate (which consisted of the director and his deputies) on technical matters. Several specialized committees such as the architecture committee, the software committee, etc. were subordinate to this council. This council could make recommendations, but the real decisions were made by the directorate, and the division heads.

4.5 El’brus-3

4.5.1 El’brus-3 Origins

As the first El’brus-2 processors were being completed in 1984-1985, the El’brus team under B. A. Babayan began preliminary work on the next generation of machines, and in 1985 ITMVT received a state order for the design and construction of a machine with a theoretical peak performance of 10 Gflops. The basic requirements for the El’brus-3 remained the same as those for the El’brus-1, and -2. High-performance on both scientific and general-purpose computation, reliability, and software compatibility with earlier El’brus models were particularly important.

There were a number of problems with the El'brus-1, -2 design which made it unsuitable as a basis for a machine with the performance called for in the state order. In addition to the architectural features limiting performance discussed in section 4.3.5, much information about instruction and data dependencies that can be determined from the source code was not available to the dynamic scheduler at run-time. The scheduler could only look ahead, in the best case, 32 instructions (the total number of buffer stations holding instructions and operands/operand addresses at each functional unit). This was frequently insufficient, especially in the case of conditional control transfers. Also, the dynamic scheduling made diagnostics very difficult. It was impossible to determine statically the exact order in which instructions were executed. The changes in scheduling from one run to another produced performance figures that also varied. Babayan says that it was very uncomfortable for him when demonstrating the system's performance before government commissions not to be able to duplicate precisely the performance claims! For these reasons, designers decided use pipelined functional units and look at static instruction scheduling.

To gain more control over execution and use more of the data and instruction dependency information contained in a program, developers began considering the possibility of giving the translator knowledge of, and control over, each execution cycle. In other words, the compiler would have sufficient knowledge of instruction execution times, memory access times, and transmission delays that it could schedule execution at a very fine-grain level. Developers were initially not sure that such a machine could be built, but were inspired by the experience of Floating-Point Systems, Inc. (FPS), which has built attached-array processors in which programmers are given very low-level control over the scheduling of each functional unit.

Thanks in large part to the design efforts and persistence of ITMVT, development of I300B ECL gate arrays with 1500 gates/chip[11] had progressed far enough in the Ministry of the Electronics (Minelektronprom) by 1984-1985 that computer engineers could begin designing machines which would incorporate them. The chips, with a minimum and average gate delay of 400 and 800-900 picoseconds, would support the design of a processor with a clock period of 10 nsec.

The first draft design of the El'brus-3 was a pure, traditional vector-pipeline machine. Even with a 10 nsec clock period, a conventional vector-pipeline processor producing two results per clock period would have a theoretical peak performance of only 200 Mflops. To achieve 10 Gflops, 50 such processors would have to be combined into a single system! Greater numbers of pipes could be incorporated into a single processor, but the types of problems which could effectively use a greater number of pipes is limited; it was a strong requirement that the El'brus-3 have high performance not only on scientific, but also on general-purpose applications. For this reason a pure vector-pipeline approach was rejected [Doro92, 16-17].

To satisfy the El'brus-3 requirements and achieve greater control over execution, designers adopted a very-long-instruction-word (VLIW) architecture. VLIW is a one form of instruction-level parallelism [Fish91]. Others are superscalar and data flow. The three vary in how dependencies between instructions are specified and which part of the system (programmer or compiler vs. hardware) specifies the dependencies and makes scheduling decisions. VLIW places this burden on the compiler, while a superscalar approach places it on the hardware (as in the El'brus-2). In VLIW terminology, basic units of computation such as addition, memory load, and branch are called operations. These corre-

[11] In comparison, the early Cray X-MP (prototyped in 1982) used 16-gate array circuits. The Cray Y-MP (1987) uses 2,500-gate array circuits, and the new C90 (1991) uses 10,000-gate array circuits. All these gate arrays are built using ECL technology.

spond to instructions in traditional sequential architectures. A VLIW instruction is a set of operations that are to be executed simultaneously. The compiler schedules a program by forming instructions out of operations that can be executed simultaneously.

Although they abandoned the complex, zero-address instructions for stack processing, and dynamic instruction scheduling of the El'brus-2, designers continued to adhere to several other El'brus design principles. Coarse-grain multiprocessing, modularity, shared-memory, hardware support for high-level languages and the operating system, multiple functional units, and hardware data tags were all incorporated into the El'brus-3. The design requirements which had given rise to these principles in the first place were still in effect. Furthermore, the requirements that software be portable between generations of El'brus computers was no longer just a good philosophy; it was a practical necessity since a considerable amount of systems and applications software now existed for these machines. The early decision to code all El'brus software in a high-level language gave designers the freedom to alter the architecture of individual processors radically. At the same time, the procedure-oriented nature of the El'-76 programming language and process-scheduling on the El'brus-2 made it necessary to preserve a procedure-orientation in the El'brus-3. As we shall see, this requirement was a principal cause of the difference between the El'brus and Western VLIW approaches.

Following a brief description of the El'brus-3 architecture, we will compare the El'brus-3 with two Western commercial VLIW machines, the Cydra 5 from Cydrome, Inc. and the Trace computers from Multiflow, Inc.

4.5.2 System Organization

To reach the goal of 10 Gflops, Babayan and others designed a tightly-coupled multiprocessor, shown in Figure 4-2 and described in more detail in [Baba90; Doro92], consisting of 16 processors, each with seven pipelined functional units (five arithmetic)

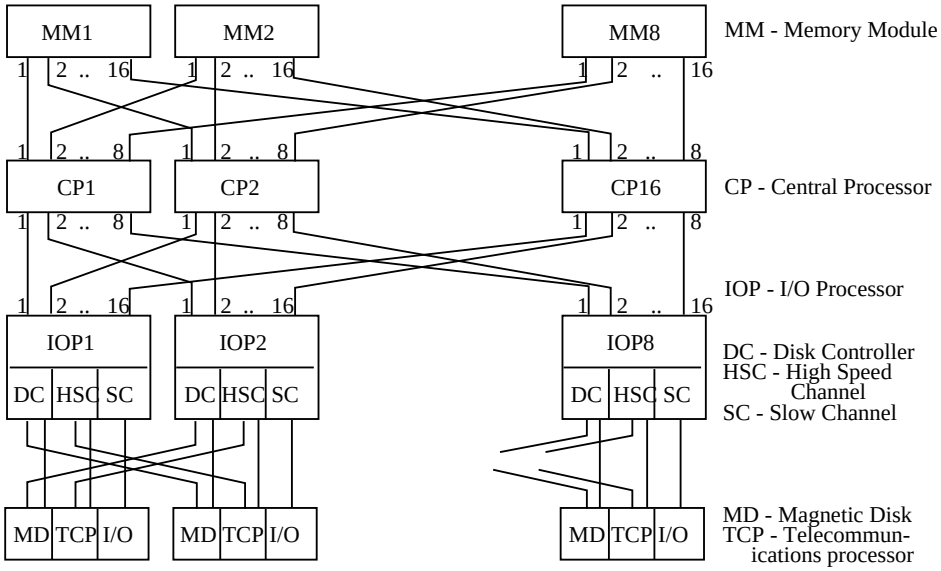


Figure 4-2 El'brus-3 Structure
Source:[Baba90, 38]

which could generate a result in each 10 nsec clock period.[12] The results of gate-level simulation performed after wire routing showed that a 12.5 nsec clock period was needed to allow the necessary data transfer at each stage, forcing designers in 1990 to change the clock period [Doro92, 7; Supe91b, 20]. Each processor therefore has a theoretical peak performance of 560 Mflops, or 8.96 Gflops for the full configuration [Doro92, 7]. The configuration also contains eight shared main memory sections, eight I/O processors, 16 telecommunications processors, magnetic disks, and other I/O devices. There are 18 Mbytes (2-megawords) of local memory in each CPU, interleaved on 32 banks. Each main memory section has a 288 Mbyte (32-megawords) capacity, with a cycle time of

[12]Each of the two logic units could perform a compare operation on floating-point operands.

400 nsec. The 16-ported main memory section is interleaved on 128 banks with a cycle time of 35 clock periods.

Data are transferred between the CPUs buffers, local and main memories, and the I/O processors via an I/O buffer in each CPU and input and output crossbar switches. Each of the eight I/O processors has a 200 Mbytes/s data transfer rate to and from the CPUs for an aggregate bandwidth of 1600 Mbytes/s.

4.5.3 CPU

The CPU consists of nine functional units: two adders, two multipliers, one divider, two load/store units, and two logic units. To reduce the complexity of the interconnect structure, the divider and one of the logical units are each combined with a load/store unit to form two compound units. Thus, there are seven functional units interconnected by a full 15x16 crossbar switch. All functional units are pipelined. The CPU also includes an Instruction Unit, an Indexation Unit, a Cache/Memory Management Unit, Local Processor Memory, and a Buffer Memory. The latter includes a 1024-word stack buffer and a 512-word array element buffer [Baba89b, 125]. The El'brus-3 employs the same cache-coherence mechanisms as the El'brus-2 [Baba89b, 125-126].

Operands for the functional units can come from

- seven functional unit outputs;
- seven synchronous history result buffers;
- a multiported stack buffer;
- a multiported array element buffer;
- literals from the Instruction Unit.

4.5.4 Influences on El'brus-3 Design

The El'brus-3 designers were strongly influenced by Western work on horizontal architectures. In 1981, FPS introduced the AP-120B attached-array processor. This system contained two floating-point arithmetic units which received operands via multiple data paths [Hock88, 209-224]. Instructions, issued every clock period, had fields which controlled the operation of all units in the computer during that clock period. These units included the two arithmetic units, an ALU for computing addresses, loop counts and indices, I/O ports, memory banks, etc. Programming the AP-120B required a very detailed understanding of the low-level timing and operations of the machine; as a result, very few people (chiefly those at FPS) programmed the system directly. Most users limited themselves to using the subroutine library provided by FPS. In short, given the technology and construction, the machine provided good performance, but it was quite inflexible and difficult to program. In general, an attached-array processor is more difficult to use and less efficient than a stand-alone machine because the programmer must manage two machines and explicitly move data back and forth between them. The data transfer path is slow relative to the processing speed of the attached processor, making it suitable only for problems in which the ratio of computation to data is high.

During the early 1980s Joseph Fisher worked on compilers for horizontal architectures which incorporated trace scheduling [Fish84]. Trace scheduling is a global compaction technique originally developed for generating long instructions of microcode from a sequential source (horizontal microcode). When using trace scheduling, the compiler ''guesses'' at the runtime control flow of a program so that sequences of code can be executed in advance, in parallel. Code can be reorganized when scheduling the Trace. To preserve the correct state of the machine to the external world if the compiler has made an incorrect guess and must backtrack, additional code (''compensation code'') is in-

serted to recover the proper state. Obviously, the system works best when the compiler makes correct guesses. An advantage of Fisher's compilers was that they performed global optimization of a program. The compiler analyzed the entire program and in principle had the freedom to rearrange code throughout the program, not preserving the integrity of the program's basic blocks, at least not at the scheduling level. Much of Fisher's work was incorporated into the Trace series of VLIW computers developed by Multiflow, Inc. during the late 1980s.

The work of FPS and Fisher in particular influenced El'brus designers. The latter did not seek to copy the Western work, but studied it thoroughly to understand the strengths and weaknesses. In particular, they did not like the difficulty of programming and inflexibility of the FPS attached-array processors, or the way that Fisher's compensation code greatly increased the size of the executable (and slowed down execution when an incorrect ‘‘guess’’ was made). The most important influence of the Western work was that it demonstrated that a VLIW approach was possible, giving El'brus designers the confidence to proceed.

A third body of Western VLIW work resulted in the construction of the Cydra 5 departmental supercomputer at Cydrome, Inc. [Rau89]. As we shall see, there is significant overlap between many architectural features of this machine and the El'brus-3. However, the El'brus-3 had been nearly completely designed before 1989, when El'brus engineers first learned of the Cydra 5 [Rau89]. The El'brus engineers made many of the same design solutions, but independently of the Cydra work. This is an excellent example of the parallel evolution of technology.

4.5.5 Comparison of El'brus-3 with Western VLIW Machines

4.5.5.1 Scheduling

Like the Trace and Cydra 5, the El'brus-3 performs a global analysis of a source program. Like the Trace, it analyzes one procedure or module at a time, looking for the critical path of computation. The compiler builds data dependency, control dependency, and procedure call graphs and tries to discover hidden dependencies between addresses to be computed at run time.

In spite of its advantages in achieving greater optimization, however, a global scheduling approach could not be used by the El'brus designers. Because the El'brus compiler must use the same code for all calls to a given procedure, it uses a procedure-oriented static scheduling (PSS) model in which code motion is restricted to within procedure boundaries. Consequently, the El'brus-3 designers used methods of reducing control dependencies in conditional branches and procedure calls that differ from those used in the Trace and Cydra 5.

The fundamental reasons for the PSS lie in the El'brus design goals and development environment, which differ from those of the Trace and Cydra 5. In contrast to the American computers, the El'brus-3, like the El'brus-1, and -2, was developed almost exclusively for military applications. The two primary requirements were high performance and portability of software. Cost, either in rubles or volume of hardware, was not a strong constraint.

Under these circumstances designers decided to duplicate a set of functional units, to implement an expensive full crossbar switch and distributed register memory for intermediate results, to use multiported register files, and so on. The availability of a sufficient number of functional units made it possible to execute multiple operations from alterna-

tive basic blocks in parallel, rather than ‘‘guessing’’ at the correct path, as in the Trace. The El’brus-3 compiler does not select the most likely path and does not use compensation code to restore program correctness if some predictions were wrong. Instead, multiple possible paths are executed simultaneously. To branch means to take the results produced by the true path. Further details of scheduling of branches are discussed in [Doro92, 15-16]. There it is shown that the El’brus-3 and Cydra 5 treatments of memory access operations in alternative execution paths are equivalent (although slightly different in implementation).

To preserve the portability of software, the El’brus-3 had to support El’-76, the base language of all models of El’brus computers. A procedure is a key object in the language. It is both a building block of the program and a basic computation unit. Designers were obliged to preserve the ability to build a program from separately scheduled procedures without any code duplication. Code motion is therefore limited by procedure boundaries and the same procedure code is used in all calls to a procedure. This contrasts with the in-line procedure substitution used in the Trace and Cydra 5.

4.5.5.2 Loops

One of the goals of the El’brus-3 design was to achieve vector supercomputer performance on scientific applications while providing high performance on a broader set of applications. The Cydra 5 designers had the same goal, building their machine to handle all the workload of a typical department, not just the numerical tasks [Rau89, 12-13]. Designing a machine which could process loops with recurrences and conditional branches was key. Such loops are typically not vectorizable on traditional vector-pipeline machines and therefore do not utilize the computing potential of those machines well.

As is described in [Doro92, 16-19], designers of both machines independently implemented very similar solutions. The approach of both machines is based on the ability to

store and access multiple loop iteration contexts during loop processing. For each loop iteration a set of registers, called an iteration frame, is dynamically allocated within the circular array element buffer. All variables to be processed during a loop iteration are placed within its iteration frame. Such variables can be both array or vector elements and scalars. They are accessed by means of base loop registers, each of which holds an iteration pointer and the iteration size. Any array element buffer is the sum of an instruction's displacement and the pointer.

Fast random access to vector elements has two primary advantages. On the one hand, the use of such a register file makes it possible to attain vector processing performance comparable to that of a vector supercomputer. In comparison with the latter, the El'brus-3 has additional overhead costs from the adding of the displacement to the loop base register.

On the other hand, random access to vector elements is much more flexible than in the case of vector registers. First, the size of an iteration frame and the location of any loop variable within the frame are known at compile time. Second, multiple iteration frames can be accessed during one clock period. As a result, any loop variable within the frames placed in the array element buffer can be accessed. The iteration frame approach is superior to vector registers in processing loops with recurrences.

In addition to providing mechanisms to support operations on multiple iterations of one loop, the El'brus designers implemented hardware mechanisms to help handle nested loops. In contrast, no architectural support for loops exists in the Trace, where loop unrolling in software is used. In the Cydra 5, hardware support similar to that of the El'brus-3 is provided for allocating registers in the array element buffer for each iteration, storing an iteration frame history, simultaneously accessing multiple iteration frames, and look-

ahead loading of data for the next iteration. It does not have hardware to handle nested loops.

4.5.5.3 Instructions

El'brus-3 instructions consist of a set of operations and an instruction format. Operations can be both half-single (32-bit data and 4-bit tag) and single precision (64-bit data and 8-bit tag). There are also operations to work with any word of double-precision (128-bit) data. The El'brus-3 approach to instruction encoding is the same as for the Trace: a packed variable-length memory representation of unpacked fixed-length machine instructions. The packed El'brus-3 instruction can consist of one to four 72-bit words. Locations of some operations within the instruction are fixed and others are variable. The unpacked instruction is 504 bits wide; it has a fixed location for each operation.[13] The Cydra 5 also uses an instruction format in which six operations to the six functional units could be initiated by one instruction. The MultiOp format consists of seven partitions, one for each operation and an additional one to control the instruction unit and other miscellaneous operations. It is 256 bits long [Rau89, 24]. The Cydra 5 has an additional format, the UniOp, which allows only a single operation to be initiated per instruction, making it possible to fit multiple UniOp instructions in each 256 bit instruction word. This was done to handle portions of code with little parallelism more effectively.

4.5.5.4 Synchronization and Exception Handling

Thanks to the shared-memory, multiprocessor nature of the El'brus configurations, there are some cases in which the static schedule of an El'brus computation can be altered at run time: overlapped execution for two independently scheduled procedures,

[13] Only 456 bits are used, however. [Baba89b, 123] states that the instruction word was 320 bits long. The size of the word was changed in 1989, after that article was written.

communication with asynchronous memory, and exception handling. The first case is specific to the El'brus, arising from the need to support separate compilation and execution of procedures, potentially using the same executable image; the others are traditional challenges for any VLIW architecture. In contrast to the Trace and Cydra 5, the El'brus-3 features the use of common, shared memory that is scheduled non-statically. The designers had no choice, because static scheduling of memory is not possible for multiprocessors with shared memory. The specific solutions to these problems are given in [Doro92, 21-22].

4.5.6 Performance

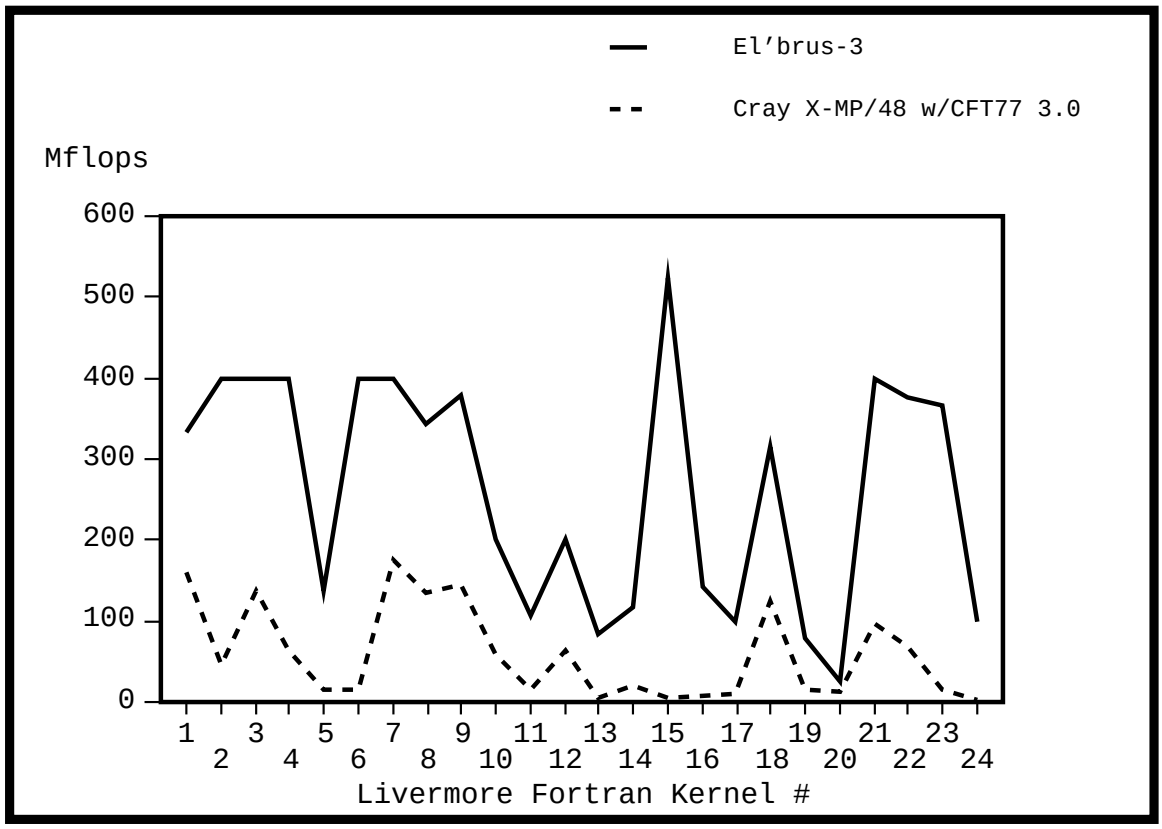
Since a fully operational El'brus-3 processor has not been built at the time of this writing, it is impossible to state what the actual performance will be like. A simulation of a (10 nsec) single-processor El'brus-3 running the Livermore FORTRAN Kernels was run on an El'brus-2.[14]

Figure 4-3 compares these results with real Cray X-MP/48 (single processor configuration) performance on the same benchmarks.

Although the meaningfulness of the simulation results can be debated,[15] they seem to indicate that the El'brus-3 is able to execute considerably more operations per clock period than the Cray X-MP, and, relative to the theoretical peak performance, has a more

[14] Jack Dongarra, the well-known author of the widely-used LINPACK benchmarks has remarked, ''I never believe simulations.''

[15] In particular, why should the performance on Loop 15 be the highest of all? It is vectorizable (although the Cray's CFT 77 compiler failed to take advantage of this), but not more vectorizable than other Loops.



	El’brus-3	**Cray X-MP/48**
Harmonic Mean	151.06	15.26
Maximum	523	175
Minimum	26	3.11
CPUs (in this configuration)	1	1
Clock Period (nsec)	10	9.5
Theoretical Peak Performance, Mflops (one CPU)	700	210

Figure 4-3 El’brus-3, Cray X-MP/48 Performance on LFK
Sources:[Pfei90]

consistent performance across the spectrum of applications (as measured by the harmonic mean) [Supe91b, 19].

4.5.7 Status

A mock-up of a multiplier was completed in 1990. In 1991, the manufacturing documentation was completed and several prototype processors were under construction at the Zagorsk Electro-Mechanical Factory outside of Moscow [Supe91b, 20]. Assembly suffered some delays while engineers waited for the necessary hardware from supplier factories. The El'brus-3, at the time of this writing, was undergoing hardware debugging. Financing remained stable, although marginally adequate, through 1992.

4.6 El'brus Microprocessors

During the early-mid 1980s as the El'brus-2 project was winding down, ITMVT engineers made important decisions about the future directions of the work. At this time they were strongly influenced by research strategies at IBM and DEC to provide a family of software compatible computers, covering a broad range of performances. At low end, DEC was pursuing the MicroVAX and VAXstation systems and IBM, the Micro370 desktop machines.

The El'brus approach was well suited for the creation of such a family. We have already discussed how the lack of assembler language programming enable ITMVT engineers to design a high-end, El'brus-2 compatible system with a radically different architecture. For the same reasons, engineers could build El'brus-2 compatible microproces-

sors having very different architectures.[16] Two lines of development were pursued, one based on the El'brus-2 architecture, and one based on the El'brus-3 VLIW approach.

4.6.1 El'-90

The 32-bit El'-90 microprocessor project was begun in 1986 by a team headed by V. M. Pentkovskiy (the author of the El'-76 language). The technical statement of work was created in 1987. There were few commercial RISC processors at this time, but El'-90 designers carefully studied the available literature and those microprocessors which had been marketed, such as Fairchild's Clipper, introduced in 1985.

The El'-90 architecture reflects a combination of RISC and El'brus-2 ideas. The uniqueness of the microprocessor arises from the combination of basic RISC concepts such as simple control and a simple instruction set with features traditionally part of large-scale, mainframe systems: multiple functional units, multi-level memory hierarchy (including sufficient cache), and multiprocessing support [Pent90]. During the mid-1980s, few microprocessor designers in the world had clear ideas about the best way to proceed. For this reason, many of the design decisions were indigenous.

Because providing full hardware control like the El'brus-2 would have required very extensive and complex logic in the integrated circuit, the El'-90 designers chose a RISC approach here, simplifying hardware control and placing more of the burden on the software. They used a simplified instruction set, the majority of which could be executed in

[16] An interesting historical parallel is the development of the Alpha AXP 64-bit microprocessor at DEC. Like the El'brus microprocessors, the Alpha AXP was a radical architectural departure from an established line, the VAX, and was designed for use in multiprocessor configurations. To port the Open VMS operating system and other VAX-based user applications which had significant assembler-level code, Alpha engineers had to use a sophisticated set of ''priveleged architecture library'' software routines to hide low-level hardware dependent functions, and develop a binary translator to run existing VAX binary images. This required much greater sophistication and effort than porting El'brus software to a new platform [Site93; Kron93; Thac93].

one cycle, multiple, highly-pipelined functional units, a wide control word visible to the compiler, multi-level memory hierarchy including large multi-port multi-window register files, and special instructions to support tightly-coupled multiprocessor systems. The El'-90 was also the first microprocessor implemented with a full-scaled tagged architecture to support dynamic, hardware typing, and ease of programming and debugging. Specifics about the construction of the cache–the cache's miss ratio, the write-to-memory policies, line replacement policies, etc.–were made with the goal of supporting a 10-processor configuration. Naturally, the experience of the El'brus-2 played an important role in these determinations. More detail of the El'-90 architecture can be found in [Pent90]. The ''El'brus Micro'', a multiprocessor designed to be based on the El'-90 is described in [Cher86].

The scale of the project (half a million transistors) required more extensive contacts between ITMVT and Minelektronprom than had been needed before. Thanks to G. G. Ryabov's efforts, a temporary technical collective (VNTK) was created to bridge the gap between ITMVT divisions and between ITMVT and Minelektronprom. This organization form had become possible during the mid 1980s and is discussed at greater length in chapter 6. Ultimately, the VNTK was transformed into a bona fide ITMVT division. The issue was made more urgent by the fact that the El'-90 was to be constructed using CMOS technology, rather than ECL which had been traditional at ITMVT. ITMVT engineers lacked the tools and experience to develop such technology alone. The VNTK incorporated both ITMVT engineers–V. Fel'dman and A. Zaitsev–and industry specialists, principally S. Kovalenko from Minelektronprom, in the same organizational unit. The first El'-90 prototypes were built in 1990. Because the electronics factory participating in the project, the Scientific-Research Institute of Precision Technology (NIITT) was unable

to achieve reliable production at the 1.5 micron level, no working chips were ever produced, although chips executing parts of the instruction set were developed.

The El’-91S is the successor to the El’-90. To a large extent, it preserves the design and technical decisions incorporated into the El’-90. This project did not pass the design stage.

4.6.2 El’-95

In 1989-1990 when the El’brus-3 architecture had been rather fully developed, B. A. Babayan initiated a second microprocessor project based on the VLIW architecture. The original intent was to implement the El’brus-3 architecture and structure in several chips. This proved to be very difficult given the microelectronics technology available. To simplify the implementation, the notion of hardware tags was rejected and the number of functional units decreased. The subsequent design contained three arithmetic channels, a 12-port register file, variable length instruction words, loop processing techniques, and static instruction scheduling plus speculative execution of branch alternatives.

During 1990, ITMVT began very actively looking for potential Western partners for their work. Ryabov and Babayan spoke with many American, European, and Asian companies during the early 1990s, including Hewlett-Packard, Siemens, Hyundai and others. Their work on the El’-95 microprocessor in particular impressed Bill Joy of Sun Microsystems, with whom they met in November, 1990 [Mark92]. Sun gave ITMVT the SPARC instruction set in 1991 and Babayan’s team worked on the design of a microprocessor incorporating the El’brus(-3) ideas in a manner compatible with the SPARC processors. Representatives of the two companies met a number of times during 1991 and early 1992, and in March, 1992 signed a contract for the design of a microprocessor and the development of a compiler for it [Mark92].

4.7 A Period of Change

4.7.1 Demand for ITMVT Computers

The El'brus computers were very large and very expensive, both to purchase and to operate. They cost millions of rubles,[17] needed extensive and complex cooling systems, and required a significant staff of operators to keep the system running. Although many (primarily military) users had been willing to pay large sums to achieve this computing power, not all potential users could afford these systems. This was particularly true during the early 1990s. Volume production of reasonably reliable El'brus-2 processors was finally achieved, just as the reforms introduced under *perestroika* started to take effect. As a result of the Law on State Enterprises (Associations) and the associated *khozraschet* accounting principles in particular, individual enterprises and institutes were granted significantly greater flexibility and responsibility for their own finances and became more careful with their expenditures. As government spending and investment in the military and scientific research and development were scaled back, organizations had less money to spend, and became less able to purchase expensive hardware. For example, the Institute of Applied Mathematics, which had received the first series-production BESM-6 and developed much systems software for that machine, decided to purchase a transputer-based machine rather than an El'brus-2, largely for financial reasons. Uncertainty about the future aggravated a worsening economic situation; unsure of what the future would hold, potential customers became reluctant to make large purchases.

[17] As price controls were lifted and negotiated prices became more common the late 1980s and 1990s, inflation rose to very high levels. Prices during these years are difficult to compare without specific information about price levels throughout the economy. The El'brus computers cost millions of rubles before inflation rose to significant levels.

Although the reforms gave enterprises the opportunity to purchase products directly from each other, nearly 100% of El'brus-2 processors were manufactured under state orders. As government spending decreased, economic uncertainty increased, and the financial state of enterprises and institutes worsened, orders for the El'brus-2 dropped to nearly zero by the end of 1992. Some individual processors were being manufactured for existing customers who wanted to upgrade a configuration or replace a processor, but no new installations were being created. A contributing factor, although probably not the dominant one, was the fact that the El'brus-2 had become obsolete in the eyes of many.

The new machines, the El'brus-3 and MKP, also had few customers, largely for the same reasons. Since these machines were still under development, there were few organizations outside the set of primary customers who had funded the development who wished to spend millions of rubles on unproven technology.

Ryabov has tried to find foreign customers for the machines among the countries of South-East Asia, but without success [Laza91b]. In short, by 1992, a viable market for the large machines had ceased to exist.

4.7.2 Relationships with Suppliers

By 1990, Minelektronprom enterprises felt acutely the need to manufacture products for which there was a market. The weak demand for new ITMVT machines such as the MKP and the El'brus-3 and the declining demand for the El'brus-2 meant that the demand for the chips used in these machines was also slack. By 1991 Minelektronprom had already assimilated production of the I300B gate arrays so factories would have welcomed orders for them, if volumes were high enough. This was not the case, and economic considerations were pushing Minelektronprom factories in the direction of simple, consumer products such as ICs for watches and electronic games which do not require state-of-the-art manufacturing technology and which could potentially be sold for hard

currency on the world market. At times, such production was being pursued at the expense of the few orders from ITMVT. It was also becoming difficult for ITMVT to get chips which required development work from Minelektronprom, even when money was available.

The situation worsened in 1992. As orders for El'brus-2 processors declined, the loss of a single order had a significant impact on the volume of chips that need to be ordered. When orders are sporatic, the chip manufacturers have to shut down production for a time, then start it up again. As a result, the chips which are manufactured after a lull are often defective.

The USSR disintegrated into individual states in December, 1991, and the political divisions did have some effect on ITMVT activities. The factory in Tashkent ceased production of El'brus-2s, but this reportedly was, for ITMVT, not an unwelcome event, since the factory had required much hand-holding. Specialized power supplies were also manufactured in Uzbekistan, and an alternative producer had to be found. Some contacts for the El'brus-3 were manufactured only in Yerevan, Armenia. When supplies of these were cut of as a result of the turmoil there, El'brus designers had to try to find replacements. Other plants have not been able to manufacture parts with the necessary heat-tolerance, however [Gigl921217, 3].

The political divisions have had much less of an effect on ITMVT operations than economic factors, however, since most of the suppliers are in Russia. As relationships became based on economics rather than administration, enterprises became less willing to manufacture complex goods with a limited market. It became more difficult to acquire not only chips, but power-supplies, cables, connectors, etc. In some cases, suppliers were themselves unable to get the necessary inputs. In either case, production of many El'brus

parts became sporatic, at best, and El’brus development and manufacture suffered correspondingly.

The underlying reality is that computers like the El’brus require an extensive infrastructure of up-stream industries. Failure by a few enterprises to supply the necessary parts can stall the entire production. Given the low level of redundancy among these industries, finding replacements is very difficult. In recent years, this infrastructure became more and more unreliable.

4.7.3 Relationships with the Factories

Traditionally, ITMVT and the factories were ‘‘glued together’’ by their mutual interdependence and the centralized planning mechanisms which stated that ITMVT computers were to be manufactured at specific factories for specific customers. Unlike entities in a scientific-production association, ITMVT and its factories were not linked administratively, except at higher levels within Minradioprom.

As individual enterprises were given greater freedom and responsibility for their own activities, the relationship between ITMVT and the principal factories gradually changed. The changes were not abrupt. ITMVT and the factory were still dependent on each other and both desired to continue to work closely together. Financing was still available for the factories, via ITMVT. State orders continued for the El’brus-2 and other systems through 1992. Funding for the El’brus-3 and MKP development continued–largely because the principal customers were reluctant to lose the large amounts they had already invested in these projects–although in real terms it declined, because of the high inflation levels.

Such funding was proving inadequate to support the over-all operations of the factories, however. ZEMZ and the Moscow SAM Plant no longer enjoyed the guaranteed El'brus market of earlier years.

Re-tooling for the manufacture of new machines involved considerable resources, and a market had to exist to recover those expenditures. While ITMVT directors had been able to get funding to cover the factory's cost over-runs in the past when El'brus-2 machine designs were changed, customers were now less tolerant. They began demanding that the factory cover such costs itself. Given the lack of a market, and the cost associated with high-performance computers, the series production factories were reluctant to establish a production line for them. The factories had to consider alternatives to large ITMVT computers.

Both factories started manufacturing goods for which there was a greater demand, including the assembly of personal computers, the manufacture of audio equipment, chemical products, watches, etc. The factories were also forced to become more attentive to the specific needs of customers. They began asking customers for their specific requirements and building configurations to meet those needs.

4.7.4 Relationship with the Ministry

Traditionally, the ministry played a very important role in the the life of ITMVT. It had a dominant voice in creating the Plan for ITMVT and the associated factories. All financing passed through the ministry's hands, even though it came from a customer who commissioned the R&D. The ministry also selected or removed directors. In short, the ministry played a dominant administrative and financial role.

In the years following the Law on State Enterprises (Associations), the influence of the ministry waned. ITMVT was able to deal more directly with customers, and financing

for R&D began to come directly from them. The Law on State Enterprises (Associations) gave employees of enterprises the right to elect their own leadership. One leading individual at ITMVT stated in 1991, ‘‘Now I don’t know what the role of the ministry is.’’

4.8 The Response to Change

4.8.1 Changes in Structure

The five technical divisions have remained largely intact since the start of *perestroika.* Each of them is fundamental to ITMVT’s viability in high-performance computing (HPC). As long as the ITMVT’s mission is to build the fastest machines possible, it is felt that a structured, integrated set of divisions oriented around the major projects and supporting technologies is essential.

However, the pressures to provide sufficient wages to keep ITMVT employees from leaving the institute forced some experimentation with alternative organizational forms. These new forms have not taken the place of the existing structures, but are add-ons. The 1987 Law on State Enterprises (Associations) made it possible to create cooperatives under the auspices of enterprises [Prav870701]. These small organizations had few restriction on wage levels, and could negotiate prices for the services performed. In practice, the Law on Cooperatives provided a mechanism whereby some of the restrictions on wages and financing in state enterprises could be skirted. Rather than have work done by its employees in their capacity as ITMVT employees, ITMVT could contract out work to a cooperative at a price which would allow a relatively high wage for the cooperative’s employees. The end result was that the same people would do the same work, but via a legal mechanism which allowed them to be paid more for it. The cooperative also was a mechanism by which an enterprise could convert accounting rubles (beznalichnyy) into cash (nalichnyy). Such a cooperative was created at ITMVT around 1989. Research

work, or work under the institute's Plan could not be carried out in a cooperative, but much of the supporting work, or work done under contract for specific customers (installing new disks, upgrading a computer center, etc.) could.

As new laws on organizations were passed, ITMVT gradually took advantage of them. Shortly after the 1990 Law on Small Enterprises was passed, ITMVT created a single small enterprise for essentially the same reasons as the cooperative. The number of workers at ITMVT dropped from on the order of 2500 in the late 1980s to approximately 1700 by the end of 1992, mostly because of people who accepted an early retirement offer. This also eased the burden of wages.

During the late 1980s, ITMVT made use of an older law allowing the creation of temporary scientific-technical collectives [Ntr85]. This 1983 law made it easier to form temporary structures which incorporated people from multiple industries, organizations, and divisions within a given organization. They were another effort to reduce the large gap which often existed among enterprises involved in various industrial branches. Changes in legislation made such arrangements possible, but economic and technical factors made them desirable. We have mentioned the VNTK created to support work on the El'-90. As customers became more careful with how they spent their money, ITMVT felt the need to work more productively. Such arrangements reportedly improved the efficiency of the teams. Another VNTK was formed to work on parts of the MKP project 1989. After the Law on Small Enterprises was passed, some collectives were established as small enterprises.

The introduction of new organizational forms within ITMVT could help increase wages of some individuals, but could not solve a basic difficulty with financing: the creation of large, complex systems required large amounts of financial and material support. The activities of collectives and small enterprises cannot generate nearly enough money

to support large-scale projects. ITMVT and the factories are administratively (and financially) distinct, so ITMVT receives no money from the sale of computers by the factories. In other words, the sale of current generation machines generates no income which can be used to fund development of the next generation. Traditionally, funding for large-scale R&D has came from a few principal customers who, in turn, had been allocated funds to support development by their ministries. As *khozraschet* took effect and the economic situation of these (military) customers worsened, they became reluctant and/or unable to spend the large amounts required to build new high-end machines. ITMVT had few options besides relying on government support.

The new structures, by nature small-scale and rather independent, were not well suited for the large-scale, integrated work required for large-scale systems. They were created to work on small, largely independent tasks which were funded independently of each other. Large-scale systems require considerable long-term funding from a single, or small number of sources. By the end of 1992, ITMVT had at least two types of organization. Approximately ten small enterprises had been formed, employing a few hundred people. They have their own back accounts, the right to set their own pay scales, enter into contracts independently of ITMVT leadership, and make their own personnel decisions. They all exist within the framework of ITMVT, which gives them name recognition, technical assistance, etc. The largest of these is the SPARC Center, which forms the basis for a contract between ITMVT and Sun Microsystems [Mark92]. The SPARC Center drew in individuals from a number of ITMVT divisions. Most of the workers had been part of the El'brus-3 project in Babayan's division. Babayan was also able to attract individuals from the CAD division who had worked on tools for the El'brus-3, and programmers from the division working on military embedded systems.

The core functions of ITMVT–the capability necessary to develop large-scale systems–retained the traditional integrated division structure. These functions were not a part of a small enterprise and remained under the leadership of ITMVT. Ryabov saw to it that within each division, the essential teams of engineers had enough funding to keep them together, and projects to keep them active.

In March, 1992, the SPARC Center signed a contract with Sun Microsystems to work on developing a microprocessor and the associated compiler [Mark92]. The following September, the SPARC Center signed a second agreement with SunPro, a subsidiary of Sun Microsystems, for software development for existing Sun products. This contract involved 33 people located in Moscow, at the (former) ITMVT subsidiary in Novosibirsk, and the Computer Center of the former Leningrad State University [Mark92b].

4.8.2 Changes in Technology

There have been few changes to the high-performance computing technology which can be directly attributed to the reform efforts. The technical characteristics of the MKP and the El'brus-3 reflect the requirements and design goals established when the projects were initiated. To our knowledge, there have been no changes to the El'brus-3 design, architecture, or construction which reflect attempts to make the system more viable under current conditions.

There are three main reasons for this. First, the El'brus-3 is close enough to completion that to alter its architecture, design, or construction significantly would seriously compromise it. Initiated just before the start of *perestroika,* the El'brus-3 was designed to be a very large, very powerful system, the most powerful system under development in the Soviet Union. Cost was not a strong constraint, as we have mentioned. By the time the effects of the deteriorating economy and strained development infrastructure were felt acutely in the early 1990s, the El'brus-3 had progressed to the final stages of devel-

opment. Changes at this point would have serious costs in time, effort, and money for ITMVT, its principal customers, ZEMZ, and upstream industries. Second, the principal customers still needed the most powerful computers they could obtain, and had already funded the project for half a decade. Changing the direction of the project at this point would jeopardize much of that investment. As a result, they continued to support the project. Third, ITMVT leadership felt that not following through on the El'brus-3 (and the MKP) would compromise the institute's position and capabilities in the high-performance computing sector. In Ryabov's words [Ryab91, 5]:

> As for the larger machines, there have not been any sharp turns. We must absolutely carry through the work to the creation of a prototype. This is not simply connected with economics; it is necessary for maintaining the level of development. To be fully convinced of one's calculations, of one's achievements. As concerns the El'brus-3, it is simply a matter of implementing this long-instruction word architecture, of obtaining a working instrument. It is difficult to overestimate the feedback loop. You can't do everything on paper.

4.8.3 Preserving Capability

The high-performance computing industry depends on the triad of developers, manufacturers and supporting industries, and users. The industry cannot survive unless each of these is active. Since ITMVT had been a leader and industry driver for decades, its challenge was not only preserving itself, but the entire industry of users, manufacturers, and supporting industries as well.

We have already mentioned that Ryabov's strategy for preserving in-house capability at ITMVT included securing sufficient financing for the El'brus-3 and MKP to keep the development teams intact, and preserving the division-oriented structure necessary for the development of large systems. In addition, he sought to find tasks which, while not nec-

essarily internationally competitive, would enable engineers to continue practicing their professional skills until funding for better projects became available. Such tasks were necessary for those involved in microelectronics and CAD development in particular.

To the extent possible, Ryabov worked to ‘‘secure an income’’ for the Minelektronprom factories developing El’brus chips. This included funding development out of discretionary ITMVT funds, trying to acquire government funding for development, and finding customers for machines built with those chips. To encourage the factories to continue production of the large machines, ITMVT was compelled to subsidize construction of the production technology and find customers for them, i.e. do much of the factories’ ‘‘marketing’’ [Supe91, 18].

To support the user community and expand the user base, Ryabov searched for ways to grant access to users who could not afford to purchase the machines, and include users outside the circle of traditional primary customers in the development process. One mechanism he plans to use is the creation of a supercomputing center located at Moscow State University. Since Soviet computer users are less able than before to purchase large, expensive systems themselves, a supercomputer center would allow multiple organizations to pool their resources and acquire a machine to use jointly. Ryabov is spearheading such an effort together with V. M. Repin, director of the Scientific-Research Computing Center at Moscow State University.

The centers would play an important role in ‘‘breaking in’’ new models, such as the MKP. The two or three years following state testing of the initial prototype have always involved considerable debugging and ‘‘modernization,’’ improvement of systems software and occasional upgrading of hardware elements. In the past, this has always been done at the installations of the primary customers, those who paid for the R&D. Ryabov is eager to place some of the first units into the hands of universities and centers such as

that just described to promote much broader involvement in the ‘‘breaking in,’’ and the more rapid development and perfecting of software. In an effort to avoid the criticism that accompanied the introduction of the El’brus line, ‘‘[w]e will try to see to it that each stage is controlled by our users.’’

Ryabov’s conviction (shared by many others) that all parts of the high-performance computing sector had to work together to survive was a principal factor in the creation of the Supercomputer Association of Users, Developers, and Manufacturers of High-Performance Systems on January 11, 1991. As indicated by its name, the association was formed by organizations representing a wide spectrum of high-performance computing activities. Members include ITMVT, the Moscow SAM Plant, the Kurchatov Institute of Atomic Energy, Moscow State University, several space-related research institutes, and integrated-circuit manufacturers from Zelenograd [Usdi91, 30]. In all, twenty-six organizations contributed to the start-up fund of the association, but considerably more have participated in the association’s activities [Supe91, 20].

Many of the participants in the Supercomputer Association were involved in a user-group for the BESM-6 which existed from 1968 to 1976 [Supe91b, 36]. This group, consisting of users and developers, held formal conferences every year and a half, and regular seminars. Like the BESM-6 users’ group, the Supercomputer Association was designed to draw together individuals from the entire spectrum of the HPC community to exchange information, expertise, and opinions [Supe91b, 36]. Among other purposes, such as the coordination and support of promising small-scale projects, the Association serves as a mouthpiece of the HPC community. Given the state of the nation’s economy, the sector had little hope of long-term viability if the government did not support it. The Supercomputer Association became one of the chief lobbying groups [Supe91, 6].

In October, 1991, the Supercomputer Association organized the conference ‘‘Problems of the Development of High-Performance Computing Systems,’’ held at a resort complex in Nepetsino, outside of Moscow. It drew approximately 210 individuals from 63 organizations [Supe91b, 34]. The conference sought to provide a forum in which four basic issues could be discussed: 1) the state of computer technology in the USSR; 2) the large problems–technical, economic, social, and political–facing the industry; 3) issues relevant to the user community; 4) issues regarding the future activities of the Supercomputer Association itself.

Many presentations and talks were given, but the underlying theme that ‘‘united we stand; divided we fall’’ was prevalent. No longer was it safe to assume that the government would sustain the sector. There were those within the government who argued (and continue to argue) that the questions of providing food, housing, and consumer goods to the populace and averting a breakdown of the social fabric are more important than the support of high-performance computing. The HPC community needed to make an equally compelling case that the future of science in the former Soviet Union depends to no small degree on the continuity of the scientific community through these troubled times, and that high-performance computing was not just one such branch of science, but was an important enabling technology for the rest. Many presentations, including a dubbed broadcast of an episode of ‘‘Adam Smith’s Money World’’ dealing with the Western supercomputing industry were used to stress this point. In short, the conference was designed to convince the HPC community to work together, and the government officials present that HPC deserved on-going support. The conference also gave ITMVT the opportunity to make a sales pitch for its own machines.

Was the conference successful? As G. G. Ryabov said late in 1992, ‘‘[t]he principal achievement of the conference is that the projects of ITMVT have not died’’ [Ryab92, 1].

The conference was probably not solely responsible for this fact, but did give ITMVT and other organizations a forum for making their case. HPC continues to be supported, albeit at a level which is hardly adequate for future development.

As 1992 drew to a close, HPC capability still existed, thanks to the persistence of ITMVT, although it was hanging by a thread. Although the factories had shifted much of their production to consumer goods, they still maintained, at a minimum level, the capability to manufacture high-performance computers.

4.9 Discussion

Since 1950 when S. A. Lebedev moved from Kiev to Moscow, ITMVT has been the unquestioned leader in Soviet high-performance computer research and development. Over the course of more than four decades engineers here have built over a half-dozen generations of high-performance computing systems for industrial use, amassing in the process considerable experience in all facets of computer development. In the process, the institute has been a driving force for the Soviet computing industry as a whole, sometimes pioneering the development of new technology, and always providing a high-profile project to serve as the focal point for technological advance in upstream industries.

The El'brus computers have been the most powerful built by Soviet industry, designed and developed for the most demanding users in the country. While reflecting the influence of foreign work, they are indigenous products which embody a host of original design decisions. Some, such as the particular dynamic scheduling and cache segmentation mechanisms are original to the El'brus. Others, like the use of hardware tags, were not original in the context of world-wide development, but were quantitative or qualitative variations of existing ideas. The evolution of the El'brus computers has been long and difficult. The R&D cycles regularly have stretched across ten years or more. As a

result, design decisions which were quite advanced at the time they were conceived, often were less so by the time the machines entered series production when measured against the international state-of-the-art.

There has been a great deal of continuity between the generations of this system, even during the turbulent *perestroika* years. Basic features of the design have remained qualitatively unchanged from one generation to another, although quantitatively the difference between El'brus generations is often greater than between consecutive models in Western supercomputer lines. Nevertheless, there are some highly significant qualitative differences which make the El'brus line a useful case for studying technological advance.

In this section we summarize the main factors which have shaped the complex evolution of the El'brus computers and the organization within which they were developed in light of the conceptual framework discussed in chapter 2. We highlight the changes in these factors since the start of the *perestroika* reforms and analyze their impact. We will set the stage for an examination of technological paradigms and trajectories in this context, and identify possible contributions of the El'brus story to contingency theories. In conclusion, we discuss the prospects for these machines and ITMVT more generally.

4.9.1 The Technology

In table 4-3 we summarize some of the key factors influencing the evolution of the El'brus computers. Items in bold font indicate changes since 1985 from items in the preceeding line. Within this extensive set are both elements of stability and continuity, and drastic change.

The El'brus computers evolved within the context of a set of guiding principles which, with some exceptions, has remained quite consistent over the last two and a half decades in particular. Each principle has been shaped by the development context, strate-

Environment	Technology
Directive form of economic management **Increasingly market driven** Monopolistic infrastructure Extensive set up upstream industries Contrary incentive structure for enterprises **incentives based on market signals** Requirements of principal users: high performance on broad spectrum of tasks, real-time capability, high reliability, ease of programming, upward compatibility between generations Relatively close links with supporting industry, based on mutual interest and administrative ties **weakening links with supporting industry, based on mutual interest and persuation** Invasive regulations regarding organizational structure and operation **much greater autonomy to institute and sub-institute organizational structures** Strong market **very weak market**	Modular, Procedure-oriented, Multiprocessing, shared memory, Hardware tags, Segmented cache, etc. Stack-based architecture **Very-long-instruction-word architecture** Complex instruction set computer (CISC) **Reduced instruction set (RISC-like)** **Organizational structure** Traditionaldivision-oriented structure **hybrid, more flexible structure** Structure oriented around basic HPC tasks, subsystems, supporting technologies **structures oriented towards new tasks**
Technological availability	**Beliefs (design principles)**
Examples and ideas from West: Iliffe, Burroughs, horizontal architectures, etc. **Expanded opportunities for contact with West and use of Western products** Heavy reliance on advanced, newly developed components, subsystems, tools. Often un-reliable Extensive experience in computer development	Design fastest machines possible Have high average performance Develop computers for real users Place as much control as possible in hardware **Place much control in software** Develop integrated hardware/software system Develop all software in high-level language
Organizational slack	**Strategy**
High levels of funding **Inadequate funding** Integrated funding stream oriented towards large-scale projects **Greater reliance on small-scale contract/project work**	Serve as driver for supporting industries Develop close relationships with factories Cultivate in-house capabilities in variety of supporting technologies Use modular architecture, hardware support for high-level language, general-purpose architec-ture, etc.

Table 4-3 Factors Influencing El'brus Evolution

gies, and technology, and in turn has shaped the R&D strategies employed and, ulti-mately, the technologies themselves.

A foundational principle at ITMVT since its inception has been that this institute would work to design and develop the fastest computers possible. Since high-performance computers (generally categorized by their performance rates) have often been viewed as a symbol of a country's technical achievement, the target speed for a new computer has always been made with an eye towards what has been achieved elsewhere in the world. A companion principle for the El'brus ystems is that they deliver high average performance, not just theoretical peak performance.This mission has always been tempered by a second foundational principle, that ITMVT would be in the business of developing real computers for real users, not systems whose primary purpose is to demonstrate an interesting idea or feature. While the first principle has caused ITMVT to set its sights high, the second has forced developers to take a realistic view of the capabilities of the supporting industries, both at the time of design and in the near future. It has also compelled ITMVT engineers to consider all the parts of industrial computer systems. They cannot concentrate exclusively on developing a fast processor; they must consider each subsystem in the context of a complete configuration consisting of processors, memory, I/O capabilities, peripheral devices, interconnect systems, and software.

A key development strategy arises at the junction of these two principles. ITMVT is and should be, it is felt, a driver of the Soviet computing industry. The desire to build the fastest machine possible has caused ITMVT to make efforts to raise the capability of the supporting industries by being a demanding customer for advanced technology, actively participating in R&D in many fields, including microelectronics and printed-circuit boards, and playing an active role as spokesman and lobbyist for the entire high-performance computing sector. Moreover, ITMVT's leadership has felt–correctly, in our view–that if it had not play this leading role, the upstream industries would have advanced much more slowly than they did.

Taken together with the realities of developing advanced technology in the Soviet Union, the above create a Catch-22 for ITMVT. We discussed the impact that being an industry driver had on El'brus development times in section 4.4.2. Factors in ITMVT's environment, such as the nature of industrial structure and management and the corresponding disinclination of factories to assimilate new technology made the acquisition of the necessary inputs an extremely arduous task. The use of a wide array of newly-developed, unproven technologies made development difficult and time consuming. If ITMVT were not an industry driver, the supporting industries would have advanced more slowly. Because ITMVT was a driver for so many industries, however, development cycles were extremely long, by world standards. In order to remain a world-class player, ITMVT had to increase the performance of a new generation system not by 2-4 times over its predecessor like Cray, but by one or two orders of magnitude. This non-incremental approach, in turn, placed additional stress on the supporting industries, helping to ensure that development times would remain long and difficult.

Strategic decisions were made to address this issue. The first, dating back to Lebedev, was to cultivate close ties with the factories and other upstream organizations. While development cycles remained long, they would have been even longer had this strategy not been pursued. A second decision was to build in-house capability in a variety of supporting technologies.

At a more system specific level, the guiding design principles and strategies were shaped by such environmental factors as the principal users' requirements and the nature of the available technology, machine architecture ideas from the West, and the design philosophies of the engineers.

The requirement to provide high performance on a wide variety of applications reinforced the existing ITMVT mission. The reliability requirement forced engineers to con-

sider specific design features to compensate for low-quality inputs. The ease-of-programming requirement forced a departure from the traditional ITMVT approach under S. A. Lebedev.

The strategy employed to meet these requirements included the following elements. First, the machine would be modular in nature, providing redundancy in CPUs, memory, I/O processors, and data communications processors. This would facilitate maintenance, improve performance, and allow a variety of configuration sizes to suit different users. Second, high-level language and systems software constructs would be supported in hardware. Third, the system would have a general-purpose architecture using coarse-grain parallelism and shared memory. This provided the least problematic and most certain route to achieving high performance on a variety of applications.

Supporting these strategies were two design philosophies strongly held by Babayan, Burtsev and others. First, as much control as possible–over instruction scheduling, configuration control, error detection and management–should be placed in hardware. The machine should be designed to ease not only applications development, but also systems software such as operating systems, compilers, and utilities. Second, the machine should be an integrated hardware/software system.

Many of these philosophies and strategies had been forming at ITMVT during the 1960s, but the ideas put forth by Iliffe and their concrete implementation in the Burroughs machines served as a catalyst, inspiring and refining the thinking of the El’brus engineers. Similarly, the El’brus-3 evolution was shaped significantly by information about Western developments in horizontal architectures.

As we have seen, many of the requirements, strategies, and design philosophies remained constant between El’brus generations. The El’brus-2 and El’brus-3 have much in common in the basic requirements, the development environment, and the architecture on

the systems level. In the underlying CPU architectures, however, they are very different. These two machines and the El'brus microprocessors provide an excellent example of the impact of one generation of technology on the next, and the impact of shifts in certain basic design philosophies.

The El'brus-3 was developed within the context of the same basic system requirements as the El'brus-2. The latter had been a design goal of the early El'brus machines [Burt75], but was a strong requirement for the El'brus-3, since a considerable volume of systems and applications had been built. The design team of the El'brus-3 was to a large extent the same team that designed the El'brus-2; their experience and and many of their design philosophies were applied directly to the new machine. The modular system structure and coarse-grain parallelism on processors with shared memory remained in the El'brus-3, as did multiple functional units, hardware tags and the corresponding support for high-level language constructs.

The existing software, requiring compatibility at the level of the El'-76 programming language, provided both opportunities for change and constraints. The designers had great freedom to alter the underlying processor design because El'-76 did not directly reflect the low-level structure of the El'brus-2 processors. The compiler provided the transformation from a rather abstract (high-level) program representation to the executable representation which ran on the hardware. Changing the hardware required changing the compiler, not the program being compiled. In contrast, a computer's assembly language generally reflects the structure of the underlying hardware explicitly and is very sensitive to changes in it. When systems software is written (at least in part) in assembly language, the need for compatibility becomes a constraint which forces a great deal of continuity of the hardware design from one generation to the next.

Given this freedom, El'brus-3 designers could implement a VLIW architecture which reflected not only a change in instruction set, number of functional units or registers, etc., but a fundamental change in one of the design principles. Control over instruction scheduling–handled dynamically by hardware in the El'brus-2–is handled statically by software in the El'brus-3. This change in philosophy was necessitated by the requirements for performance and the perceived inadequacies of the El'brus-2, and encouraged by the experience of Western horizontal architecture efforts known El'brus-3 designers.

While they did not greatly constrain the design of hardware, the operational characteristics of El'-76 did impact some of the mechanisms which were migrated into software in the El'brus-3. The primary example is the nature of the scheduling mechanism. El'-76 is a procedure-oriented language, and the need to support re-enterable, independently compilable procedures in an El'brus configuration forced designers to develop the procedure-static scheduling model described in section 4.5.5.1.

The level of financing, or organizational slack, available for the El'brus projects also had a significant influence over the machines' design. For much of the project's history, relatively high levels of funding were provided for systems development. Resources were not unlimited, of course, but the El'brus systems were among the highest priority computing projects. Resources were available to build expensive components and subsystems, or provide additional systems resources. For examples, designers of the El'brus-3 incorporated duplicate sets of functional units, a full crossbar switch, and multiported registers to improve performance, even though they were expensive and increased the volume of hardware. The directors of ITMVT were also able to acquire additional funds to cover cost over-runs at the development factory.

The changes in ITMVT's environment since 1985 have had little impact on the design, architecture, or construction of the El'brus computers. As discussed in section 4.8.2,

basic requirements and guiding principles remained quite constant, the cost of significantly changing a project became higher as the project advanced, and the principal customers continued to fund development. The main impact of reform-related changes was to the infrastructure needed to support such a project. The relationship between ITMVT, the factories, and the supporting industries became less administratively-oriented and more based on economics as decision-making and financial authority grew more decentralized. ITMVT made a great effort to sustain the funding and preserve the industry infrastructure and capability necessary to carry the El'brus through to completion. This task grew more difficult through the 1990s and the El'brus project suffered delays as a result, but the core infrastructure and the basic goals and strategies for the technology remained intact through 1992. The prospects for the technology are discussed at greater length below.

The El'brus-1, -2, and -3 lie along a ''technological trajectory'' which has been quite consistent for over 20 years. In each generation, designers sought to increase performance through some combination of faster and improved components, reduced clock periods, greater volumes of primary and secondary storage, greater numbers of processors and functional units within processors, and improved processor architecture. Improved performance did not come at the expense of loss of generality however; the El'brus machines performed well on a wide spectrum of problems. Basic systems characteristics–coarse-grained parallelism through a moderate number of powerful processors with shared main memory, modularity, multiprocessing, independent I/O and data transmission processors, hardware support for high-level language constructs, software compatibility with previous generations–remained very similar throughout the generations. There are many points of consistency in the lower-level design: the use of a segmented cache, hardware tags, etc.

One of the few points of sharp discontinuity in the technological trajectory was the design of the individual processors, as the stack-based architecture of the El'brus-1, -2 processors was replaced by a VLIW approach in the El'brus-3. Does this reflect a shift in the technological paradigm of the El'brus? We will examine this question in greater detail in chapter 8.

4.9.2 The Organization

We described the structure of ITMVT and, in particular, the El'brus division in sections 4.4.6 and 4.8.1. Table 4-4 outlines some key influences on organizational structure. The basic structure, a traditional partitioning into division, subdivisions, and laboratories, was defined largely by the regulatory environment surrounding ITMVT and other applied research institutes throughout the USSR.

The specific structure, the number of structural units at each level and the specific tasks were determined primarily by the nature of the tasks they were to address and, occasionally, by political considerations. Because the research focus of ITMVT remained quite constant for decades, its structure remained reasonably stable. Structural changes reflected the changing nature of the research, or the desire to expand research in a particular direction. The creation of divisions to develop components, PCBs, and memory systems reflect a desire to try to compensate for weaknesses in the electronics ministry, for example.

The changes to ITMVT structure over the last decade reflect the interaction between the pressures and opportunities brought by reform processes, and the need for certain types of structures to support the institute's mission. The most significant factor creating a need for structural change has been the worsening economic climate. As financial conditions worsened, changes had to be made which would either increase the amount of organizational slack and/or improve productivity to make better use of that which was

Environment Laws establishing norms for job titles, wage levels Involvement of high-level officials in approving changes to organizational structures **Legislation allowing alternative organizational forms** **Legislation giving individual institutes the authority to determine their own structure** **Legislation implementing *khozraschet* principles at institute and sub-institute levels** Strong market for ITMVT products **Declining market for ITMVT computers**	Technology Functional division of tasks needed for El'brus development Organizational structure Traditional division, laboratory structure **More flexible, autonomous organizational forms: cooperatives, VNTK, small enterprises**
Technological availability **Examples of alternative organizational structures at other institutes**	Beliefs **Key to survival is maintenance of integrated structures** **Key to survival is retaining core development personnel**
Organizational slack Stable funding for HPC development **Decreasing government funding** **Weak, but existent funding for core R&D teams** **Payment of higher wages through non--traditional organizational units** **Foreign investment (Sun Microsystems)**	Strategy **Maintain integration of core capability** **Creat flexible autonomous organizational structures** **Seek foreign partners**

Table 4-4 Factors Influencing Organizational Structure within ITMVT

available. ''Organizational slack'' was increased in two ways, by efforts to generate increased revenues, or to convert existing resources into a more flexible form, i.e. the conversion of accounting rubles *(beznalichnyye)* into cash *(nalichnyye).* A basic goal was finding a way to pay workers wages that would sustain them and keep them at ITMVT.

The creation of cooperatives, small enterprises, and temporary collectives served each of these purposes. They provided a way to skirt legislative restrictions on wages, to enter into contracts with negotiated (i.e. higher) prices, to convert accounting rubles into cash to pay the workers, and bring together individuals best suited to carrying out a particular task in an efficient, timely manner.

Such organizational structures would not have been possible without changes to the laws regarding institutes and enterprises which made such structures possible and gave institute leaders the authority to create them. Although not a highly significant factor, ITMVT benefited from the experience of other organizations which had pioneered organizational forms soon after the laws were changed. In other words, the ‘‘technological opportunity’’ or know-how available in society helped encourage ITMVT leaders to make these changes.

Whole-scale conversion to small organizational units would have had a highly detrimental effect on the operations of the institute as a whole. ITMVT’s divisions had been designed to complement each other, and to work towards common goals. The high-performance computing divisions could not have been successful under the Soviet conditions without the contribution of the electronics and CAD divisions, for example. The various systems built at ITMVT had much overlap in their component base and construction. A non-trivial measure of centralized management and coordination was needed to integrate the efforts of the divisions.

The cooperatives and small enterprises offered new ways of increasing the wages of workers and improving certain types of work, but also decentralized the management of the institute as a whole. The small enterprises in particular were given considerable responsibility for finding their own contracts.

To maintain the institute’s overall capability and counteract the tendency towards decentralization and smaller-scale tasks, ITMVT has retained core capabilities in the traditional structures under the coordinating management of the directorate. Maintaining this structure requires financial and material support for the projects, however. To date, the government and principal customers for the large machines have continued funding at

levels which have kept the teams largely intact, but which are inadequate for comprehensive projects.

4.9.3 Prospects

The reform process has brought a number of changes with positive implications for innovation at ITMVT, but many negative consequences that threaten the institute's ability to carry out the R&D of large-scale systems that has been its primary activity for decades.

Idea generation is directly related to amount of ''cross-fertilization'' of ideas which comes about through the intensive interaction of developers with other individuals or organizations who are working in areas related to HPC. In recent years, the some necessary preconditions for fruitful cross-fertilization have been established, although the short-term effect on the development of large-scale systems will not necessarily be great. We have seen through the activities of Ryabov and the Supercomputer Association that as the centralized support and coordinating mechanisms for HPC development have grown less certain, the developers, manufacturers, and users have become more genuinely interested in working together to save the industry. Developers at ITMVT must compete for customer finances and the goal of developing a product to suit a customer's needs and requirements has become more pressing. This is particularly evident in the activities of the small enterprises which must secure contracts for themselves to survive. Close interaction with a demanding customer base is crucial for generating ideas for future development. However, if the contracts are not for leading-edge systems which demand the generation and refinement of new ideas, work will stagnate.

As restrictions for interaction with Western organizations have been relaxed, cross-fertilization can potentially come about through a greater flow of ideas to and from the West. The SPARC Center is likely to have a high-level of idea generation, although not as high as it could be. Working under contract with Sun Microsystems, the center has

close interaction with a very demanding customer who is pushing the Center to generation, refine, and implement state-of-the-art ideas at a greater rate than would have been the case otherwise. Sun is supporting the work financially and materially, but beyond the basic information the SPARC Center needs to fulfill its contract, the flow of knowledge is largely uni-directional. Because of export control restrictions, Sun is forbidden to share with SPARC Center workers its cutting-edge ideas, even when the Center's work is on a par with Sun's.

''Coalition-building''–the securing of support from high-level authorities and the infrastructure needed to supply financing, materials, tools, and know-how–has become much more problematic for ITMVT as a whole, but easier, in some respects, for Babayan's division. In a nutshell, maintenance of the extensive infrastructure necessary to support large-scale, cutting edge systems development now depends on the good-will participation of the hundreds of upstream factories and institutes. The centralized administrative coordination no longer serves this purpose.

The infrastructure now depends more on horizontal, economic links than vertical, administrative ones. While a positive development in the long term, the short term effects are disheartening. The two levers now available to keep the infrastructure together are economic self-interest and persuasion. Given the worsening state of the economy, the cutback on government funding, and the corresponding low demand for high-performance systems, neither level is operating very effectively. Complicating matters is the lack of redundancy in ITMVT's existing infrastructure. The failure of a small number of factories to deliver certain parts can easily cause the delay or failure of a project.

A fundamental problem faced by ITMVT is that it currently has few alternatives to seeking direct funding of R&D from outside sponsors, either industrial, foreign, or government. ITMVT is not administratively integrated with any organization which sells a

product which can generate in-house revenue for R&D. It does not receive an portion of a factory's profit from the sale of ITMVT high-performance systems, personal computers, watches or anything else. The small enterprises help generate small-scale revenue, but not enough to support large-scale projects. Without the possibility of directly generating revenue from its work, it will be very difficult for ITMVT to find investors other than those who are paying directly for research results. In the current economic climate, ITMVT is likely to be funded more for many, small-scale projects than for a few large-scale projects; there are now no indications that ITMVT will receive support for the development of large-scale successors to the El'brus-3 and MKP. Small-scale funding, although sometimes designed to keep core teams intact, will have a tendency to reduce the level of integration of ITMVT as a whole, making it more difficult to re-establish large projects in the future.

Babayan and his team at the SPARC Center are in a more favorable position. First, their work is being sponsored at a level adequate to the task. Coalition building for Babayan in this work is easier because he is no longer relying on a vast network of Soviet organizations. The Center's ''product'' is software, and a microprocessor design. Implementation and manufacturing will done in the West by Sun, a company with all the facilities necessary to bring the SPARC Center's results to market. The SPARC Center will also generate revenue through participation in the distribution channel for Sun products in the former Soviet Union. Thanks to the prestige of being hired by Sun Microsystems, Babayan's team should be in a good position to find additional support from the West, should this be necessary. If the relationship with Sun were to sour badly, however, this might not be the case. While the SPARC Center is certainly maintaining capability in machine architecture and software, the work is being done for Sun. The links between this

team and others at ITMVT which used to support the El'brus division's work are likely to grow weaker over time.

ITMVT's ability to construct prototype of large-scale systems has also been compromised by several factors. The issues of ''technological availability''–the availability of the tools, components, and manufacturing technologies–remains particularly acute for the ITMVT machines, since they have always been very dependent on advanced, customized technology. Unlike other projects which used existing, proven technology, the El'brus and MKP systems demanded the development of new technologies. We have discussed the difficulties ITMVT has had acquire the necessary inputs.

Are the prospects any better today? Could Western technology substitute for Soviet technology in its machines? The possibilities are greater now, but the success of such an approach is questionable. In principle, ITMVT can now use non-Soviet parts in its computers. National policies in the past have stated that ITMVT computer had to use completely indigenous parts. This restriction has now been lifted. If it had the necessary financial resources (a non-trivial problem), ITMVT could acquire Western integrated circuits, CAD systems and workstations, disk drives, and many other pieces of commodity, or near commodity, technology. The borders between East and West have now become quite permeable to this kind of trade.

However, systems of the scale and complexity of the high-end ITMVT machines cannot be built solely out of commodity parts. They use customized chips, printed-circuit boards, cooling systems, cabling, etc. The development of the architecture is intimately linked with the development of these elements. ITMVT could reduce its dependence on some of the upstream Soviet industries, but would not be able to build machines solely out of Western parts. First, although CoCom restrictions have eased in recent years, they are still very much in force for advanced computing and the technologies–particularly

manufacturing technologies–which are necessary for their development. Second, the customized or special-purpose components and subsystems are very expensive in the West and could easily cost more than ITMVT or any customer would be willing to pay at present. Third, the logistical problems of trying to deal with vendors of specialized technology in the West would at best delay development and increase costs considerably.

Although the creation of flexible teams to address specific, temporary tasks may increase the institute's productivity in specific areas, the problem of prototype development is completely overshadowed by the difficulty of acquiring or developing the necessary inputs.

The path for ''innovation diffusion'' of ITMVT's machines has been well defined since the early 1960s. The relationships with the factories and associated special-design bureaus have been quite stable. ITMVT designed its machines for industrial use. Since the late 1980s, however, ITMVT's ability to get machines into series production has deteriorated as the market for the large systems collapsed and the relationships with the factories moved towards a greater economic basis. On the other hand, ITMVT is paying closer attention to what customers are willing to purchase may in the future develop systems for which a greater market exists.

What are the possible scenarios for ITMVT in the future? Will it be able to continue to develop large supercomputers? In the short term, the answer is most likely, no. The projects nearing completion–the El'brus-3 and MKP–will probably be completed. This task is difficult, but not impossible given that the necessary inputs have already been developed. The prospects for funding for a new large-scale project are currently very slim, and even with funding, the challenge of getting the cooperation of the upstream industries is very formidable. Without a steady stream of large-scale funding and support, there will not be a new generation of high-end systems.

For the intermediate term, ITMVT leaders are concentrating on keeping the core teams intact in anticipation of improved circumstances in the future. So far, they have succeeded. In the absence of a single, over-arching project to keep them integrated and pursuing a common goal, the work of teams is likely to drift apart over time. Even if each team works on state-of-the-art projects (such as at the SPARC Center), incompatibilities in technology, orientation, or culture will gradually arise between them. The more time passes, the more difficult it will be to reunite the pockets of capability into a single large-scale project.

The most likely scenario in our opinion is that ITMVT will continue to be transformed from a single, focused institute to a collection of smaller organizations pursuing their own projects. ITMVT will continue to lose its ability to organize and support the upstream industries and will have to become more of technology follower than driver. In other words, the upstream industries (some domestic, some foreign) will evolve in their own fashion, and ITMVT will increasing have to accept the technology available and build what it can using it. Although the ability to develop large-scale systems will atrophy, its ability to build quality systems on a smaller scale may very well be enhanced.

CHAPTER 5. SCIENTIFIC RESEARCH INSTITUTE OF CONTROL COMPUTERS

5.1 Introduction

In this chapter we examine high performance computing developments within the Problem-oriented Computing Systems Division of the Scientfic-Research Institute of Control Computers (NIIUVM) of the Impul's Scientific Production Association in Severodonetsk, Ukraine. This division was created to develop high-performance systems oriented towards problems that are computationally intensive and have a high degree of data parallelism. The reconfigurable structure (PS-) parallel processors are among the most successful in Soviet high performance computing. While not the most powerful machines, the PS-2x00 claimed more installations than any other Soviet high-performance machines and have had a signficiant impact in a host of applications domains from geophysics to atomic energy to space research.

These machines were developed in an industrial setting and historical tradition which strongly influenced their design, construction, and use. After examining some of the history of computer development at NIIUVM, we look in detail at the PS-2x00 computers, highlighting their technical and performance characteristics. With the third generation of the PS-2x00 machines currently under development, we have the opportunity to trace their evolution, identifying the environmental, technological, and organizational factors and design principles and strategies which shaped the machines' development. To a lesser degree, we examine their contribution to the Soviet economy and to developments in computing as a whole.

The Soviet Union witnessed many changes between the start of *perestroika* and its dissolution in 1991 but not all of them had a significant impact on HPC research and development at NIIUVM. We examine the most significant changes in the environment of

the Problem-oriented Computer Division, including changes in legislation, the relationship with suppliers, factories, the ministry, and with the sponsors of the HPC work. We also track the changing demand (market) for these machines.

We look at the impact of change in two areas: the evolution of HPC technology and organizational change in the Problem-oriented Computing Systems division. We conclude by discussing the prospects for HPC at NIIUVM.

5.2 History of NIIUVM Research

During the early 1930s, the Soviet government founded the Lisichansk Chemical Combinat in a virtually uninhabited portion of the Donetsk region in Eastern Ukraine. The village that grew up around the Combinat was renamed from Liskhimstroy to Severodonetsk in 1950, and attained the status of city in 1958.

In 1956, the Lisichansk Subsidiary of the Institute of Automation (LFIA) was created in Severodonetsk to provide research and development facilities for the growing chemicals industry. The first director, V. Yu .Tolkachev, attracted many bright young scientists and engineers to Severodonetsk to work on the design and implementation of electronic equipment for industry [Raki91, 1-9]. Two of these, V. V. Rezanov and I. I. Itenberg, were to play critical roles in the development of both general-purpose control computers and the parallel processing computers discussed below. During the early 1960s, the institute was renamed the Scientific Research Institute of Control Computing Machines (NIIUVM).

Built at approximately the same time as the institute, the Severodonetsk Instrument-Building Factory (SPZ) in 1959 began series production of its first devices, temperature and pressure regulators for the chemical industry. Since then, it has focused primarily on electronic instrumentation for the chemical and other industries, and only secondarily on

computers [Pert86, 22]. The first series production of digital computers here began during the mid-1960s

In 1971, the Impul's Scientific-Production Association (NPO Impul's) was formed on the basis of a number of Severodonetsk organizations as part of a national reform effort to improve the linkages between science and industry through the creation of the scientific production associations [Pert86, 21]. Impul's consisted of three structural units and three independent enterprises. The head organization, NIIUVM, was responsible for all Impul's research and development activities in the area of computing systems for data processing and process control systems. The Experimental Factory of Computer Technology was closely associated with NIIUVM and built prototype models of computers designed at NIIUVM. The third structural unit was the Impul's Foreign Trade Firm. The three independent Impul's enterprises were the Severodonetsk Instrument-building Plant (SPZ) which was responsible for series production of computers developed at NIIUVM, the Severodonetsk Training-Computing Center which trained thousands of users of Impul's computers through regular seminars, and the Volgograd Design-Engineering Bureau of Control Computing Systems which developed computer technology for building control computer systems [Impu89c]. By the end of the 1980s, approximately 12-15,000 people worked at NPO Impul's.

NIIUVM became a pioneer in Soviet computing. Numerous innovations in the areas of control systems and data processing appeared for the first time in the Soviet Union in NIIUVM machines. As described below, these include the first commercial multi-machine and single-instruction multiple data (SIMD) parallel processors. Thousands of NIIUVM control systems were installed throughout Soviet industry, in atomic and other power stations, chemical plants, metallurgy enterprises, engine manufacturing plants, seismic data processing centers, hospitals, etc. NIIUVM computers were used in the first

Soviet airline reservation system SIRENA [Zhoz85; Pert86; Raki91] and in an information system supporting the 1980 Summer Olympics [Vdov81; Pert86; Impu91, 5; Raki91, 15]. NIIUVM machines reportedly have provided climate control in Lenin's Mausoleum [Impu91, 5].

Initially, NIIUVM research focused on developing small-scale control systems called ''auto-dispatchers,'' punched-card readers, and other small process control devices, but during the 1960s researchers developed a number of indigenous computers. Among their accomplishments were the MPPI-1 (1963) for gathering information in technical processes, the UM-1 (1965) for real-time control, and the KVM-1 (1965) coordinating computer to manage installations consisting of a number of subordinate control computers. These systems together formed a hierarchical System of Operational Control of Production (SOU) which was one of the earliest Soviet attempts to create an upwardly compatible family of computers [Rudi70, 29; Apok74, 275; Raki91, 7]. Also developed at NIIUVM were the Severodonetsk, the first stack-based architecture with zero addressing built in the USSR and the M-2000 and M-3000, the first Soviet systems based on the architecture and instruction set of the IBM System/360 computers, even though they were built using transistors instead of integrated circuits [Prsu70; Yers80b, 10; Raki91, 6].

Most of the computers developed at NIIUVM from the mid-1960s on fell into the ASVT (Aggregate System of Computer Technology) classification. ASVT computers fall into four primary categories: ASVT-D, ASVT-M, ASVT-SM, and ASVT-PS. The principal machines are shown in Table 5-1. Development of ASVT machines, particularly the ASVT-M, was spread out over a number of institutes, but the machines reflected a consistent underlying design philosophy: the use of modular designs, standardized internal and external interfaces, and upward compatibility of software would support the creation

	ASVT-D		ASVT-M		
	M-2000	**M-3000**	**M-6000**	**M-7000**	
Word length (bits)	32	32	16	16	
Peak Performance (add) (KOPS)	40	100	200	400	
Main memory, maximum (bytes)	48K	96K	32/64K	128/256K	
Start of series production	-	1971	1972	1975	
Factory	-	SPZ	VUM/SPZ/ ELVA	SPZ	
	ASVT-SM				**ASVT-PS**
	SM-1	**SM-1M**	**SM-2**	**SM-2M**	**PS-1001**
Word length (bits)	16	16	16	16	16
Peak Performance (add) (KOPS)	200	200	400	450	1000
Main memory, maximum (bytes)	32/64K	128K	128/256K	128/256K	512K-4M
Start of series production	1977	1982	1978	1983	1989
Factory	SPZ	OZVM/ SPZ	SPZ	OZVM/ SPZ	SPZ

Key: VUM - Computer and Electronic Control Machines Plant (Kiev)
SPZ - Severodonetsk Instrument-building Plant (Severodonetsk)
OZVM - Orel' Computing Machine Plant (Orel')
ELVA - Elva Scientific-Production Association (Tblisi)

Table 5-1 NIIUVM Control Computers
Sources: [Rudi70; Shel73; Apok74; Kovt75; Naza75; Pevn76; Aman77,405; Eg78; Uprs77; Grub77; Polu78; Naum79; Zhim79; Zamo85; Grub89; Impu89b,5]

of a wide array of problem-oriented, upgradable configurations with clear migration paths for users without a corresponding proliferation of hardware and software nomenclature.

The M-1000, M-1010, M-2000, and M-3000 were called ASVT-D computers since they were built using discrete components. These machines were targeted chiefly towards data processing applications. The M-3000 entered series production around 1971

[Prsu70; Reza71; Aman77, 405]. It had a reported average performance of 50 KOPS [Rudi70, 31; Doly82, 13-16].

The ASVT-M class consisted of the M-40, M-400, M-4000, M-4030, M-5000, M-5010, M-5100, M-6000, and M-7000 models, and all were manufactured using integrated circuits [Shel73; Grub89, 89]. The development of these machines was divided between NIIUVM and other institutes, including the Institute of Electronic Control Computers (INEUM) in Moscow which was chosen to be the lead institute in a large-scale CMEA program initiated in 1974 to develop the SM- series of ‘‘small systems’’ (*sistema malykh*) minicomputers. NIIUVM was responsible for the M-6000 and M-7000. Together, the ASVT machines represent a considerable range of underlying architectures. While the ASVT-D systems, as well as the M-4000 and M-4030, were based on IBM’s mainframe architectures, the M-6000 and M-7000 were based on the architecture of Hewlett-Packard’s HP-2116. The M-400, on the other hand, designed at INEUM, was based on the Digital Equipment Corporations PDP family of minicomputers [Boya77, 7; Grub80, 19-20; Hamm84; Sini87]. They were not copies of their Western counterparts, but incorporated modifications which facilitated interoperability in keeping with the ASVT philosophy. Thousands of M-6000 and M-7000 were manufactured and employed in process control applications in all branches of industry.

The SM-1 and SM-2 both were upward compatibile with the M-6000 and M-7000. The SM-1 was designed for use in simple, single-level control systems as a control unit for instrumentation and a real-time processor for data generated by these instruments. The SM-2 was designed for complex, multi-level, multi-machine process control systems with stringent performance and reliability requirements. It could be configured as a dual-processor systems with shared memory to enhance reliability.

The SM-1M and SM-2M differed from the SM-1 and SM-2 chiefly in that they were manufactured using smaller boards with an improved component base consisting of LSI chips and semiconductor memory, making it possible to reduce their size significantly, decrease the number of internal connections between the processor and memory, and improve reliability overall [Grub89, 126-127].

The SM-2M reflected a number of further developments in NIIUVM systems design. First, the SM-2M was designed to have much higher levels of reliability than its predecessors. In a dual-processor configuration, the SM-2M had full redundancy such that the failure of any one component would not cause the failure of the system as a whole [Bara82; Grub89, 127]. Second, the systems software reflected the application in systems software of the modular principles used in the hardware of earlier machines. In machines like the M-6000 the systems software was quite inflexible and closed; there was little consistency in the interfaces of the composite modules and no easy way of adding modules to existing software. The essence of the Aggregate System of Software (ASPO) was a uniform, or compatible set of interfaces between systems modules, a macro-language for describing how modules could be combined, and a program generator which would translate the high-level configuration description into a full program which could then be compiled in the conventional manner. Using libraries of macro specifications and relocatable code, even the operating system could be tailored to the requirements of a specific installation [Klio89, 127-129].

The PS-1001, the primary system in the ASVT-PS category, was a considerable advancement in all respects over the SM-2M. Introduced into series production at least eight years after the SM-2M, the PS-1001 consists of much greater amounts of memory, higher processing rates, and additional functionality for the most demanding process control applications. These included networking capabilities, and the ability to configure the

machine into a system with triple-redundancy and backups of I/O and communications channels. It also can be equipped with external solid-state memory [Prsu87, 50]. This system was developed primarily for use in nuclear power plants [Astr88].

Each of the machines of the ASVT series developed at NIIUVM was developed for industry. Close relationships with users and real-world requirements forced developers to design entire systems–not just isolated computational engines–which could be manufactured in a reasonable manner and at a reasonable cost and be of high utility to users. The industrial nature of the R&D environment served as a filter for design ideas. Practicality was favored over theory. Building machines which could be used was more important than demonstrating innovative architectural ideas. This tradition had a profound effect on the development of a new line of high-performance data processing machines initiated in the mid-1970s.

5.3 The PS-2x00 Parallel Processors

Two early Western high-performance computer prototypes, the ILLIAC IV and the STARAN, inspired Soviet designers to work on a related architecture. Although the Soviet work cannot be considered an effort to duplicate the Western machines, many design features of the latter were closely studied and, to varying degrees, adopted by the Soviets. In some cases, the Soviet work was an improvement over Western approaches. We highlight the most significant similarities and differences in this section.

5.3.1 Western Antecedents

The ILLIAC IV project was initiated in 1966 when the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense awarded the University of Illinois a contract to build a computer based on a design called the SOLOMON (Simultaneous Operation Linked Ordinal MOdular Network) proposed in 1962 by D. L. Slotnick and oth-

ers. The 1962 design described a two-dimensional array of 1024 processing elements performing bit-serial arithmetic on a single instruction stream originating from a single control unit. The ILLIAC IV was not an exact implementation of the SOLOMON but was a pioneering effort into the realm of highly parallel systems.

Designed for solving partial-differential equations, the original ILLIAC IV plans called for a system consisting of four quadrants of 64 floating-point processing elements. Each quadrant was to have a single control unit interpreting a single stream of instructions to be executed simultaneously by all the processing elements in the quadrant. The quadrants were to be connected with one another by a high-speed parallel I/O bus. Each processing element was to have a local store of 2K 64-bit words, and within each quadrant the processing elements were to be arranged in an 8x8 array using a NEWS (North East West South) nearest-neighbor interconnect topology, connecting each processing element with four others. The main memory of the system was implemented in disk with a capacity of 16 megawords and a transfer rate of 500 Megabits per second. The individual processing elements operated in two modes, enabled or disabled. The mode could be set independently under program control for each of the 64 processing elements. The ILLIAC IV was attached to a conventional computer, the PDP-10, which managed I/O, the memory subsystem, and interaction with users [Hord90, 21].

The Burroughs Corporation was selected as the systems contractor and the first unit, consisting of only one quadrant, was delivered to the NASA Ames Research Center in 1972, although usable service was not offered until 1975 [Hock88, 24-26; Hord90, 28]. Although its development path was long and arduous, the ILLIAC IV had a profound impact on computer development. An internal memo at the Institute for Advanced Computation by G. Feierbach and D. Stevenson in August 1976 listed sixteen distinct advances attributed to the ILLIAC IV. These included the first large-scale use of ECL integrated

circuits which served as a major impetus to the development of this technology, the first significant use of semiconductor memory, the first successful implementation of large multilayer laminated boards, a definitive demonstration of the array approach to computation, the development of new algorithms, and others [Hord90, 79-84]. The ILLIAC IV was to be a strong source of inspiration for Soviet computer developers as well.

The STARAN associative processor, built by Goodyear Aerospace, was conceived in 1962 and completed in 1972. It incorporated some associative, or content-addressable ideas proposed in 1960 by W. Shooman. The STARAN consisted of four array modules, each incorporating 256 single-bit processing elements. In contrast to the SOLOMON in which each processing element had a local memory store, the STARAN processing elements shared a single store of between 64K and 64M bits through a flexible 'FLIP' network [Hock88, 29-30]. Memory was accessed in terms of 256-bit ''slices'' according to a 256-bit code, or pattern. Data elements were passed between processing elements through the FLIP network which provided a highly flexible interconnect system. The STARAN was particularly suitable for image processing applications.

5.3.2 The PS-2000

5.3.2.1 History

In the years following the 1972 introduction of the ILLIAC IV and STARAN, literature about these machines was widely circulated in the Soviet Union. A group of computer scientists at the Institute of Control Problems (IPU) in Moscow began considering how they might make use of some of the ideas presented in these two machines to achieve high performance computing capability in the Soviet Union. The two Western machines were attractive in large part because they presented the possibility of achieving relatively high computation rates using a rather slow component base, albeit in a rather

narrow application domain. Under the direction of Izrail Medvedev, this group of enthusiasts developed a proposal for a machine consisting of a single sequential, scalar processor controlling a field of processors executing identical instructions simultaneously [Mnw84; Smyk85].

The design grew out of an analysis of Western developments as well as an examination of the computational models for a set of core algorithms found in many HPC applications. These included regular algorithms like FFT, recursive sum, sorting, difference equations, multi-dimensional array transformation, systolic algorithms, and others [Medv92]. Medvedev's team developed a methodology for transforming these computational models to computer architectures which could execute them. The PS-2000 was designed using this methodology to identify an architecture which would best support the computational models (or parallel-sequential transformations of them). The identified three categories of models. The first consists of identical parallel processing streams with no program branches. The second have multiple segments of type-1 processes with intersecting sections which are of the same type. Problems like matrix algebra, FFT, and others fall into this category. The third are arbitrarily structured sets of arbitrary operators. At any given time there are groups of similar operators to be executed. Their readiness to execute depends on the availability of input data and the truth of an associated predicate function. Data searching and sorting, and a number of iterative procedures have this quality. Medvedev determined that the third type of problem can be run of SIMD computers if each processing element is provided with a) an address processor for independent memory access and activation, and b) an activation processor for predicate computation. The PS-2000 is a machine designed for type-3 problems [Medv92, 50-52; Medv92b, 19-21].

In 1976 the group from IPU met with representatives from NIIUVM who had worked on the M-6000 and M-7000 under I. I. Itenberg. NIIUVM was a logical partner in this effort. IPU had dual subordination to the USSR Academy of Sciences and the Ministry of Instrument-Building, Means of Automation, and Control Systems (Minpribor), for which NIIUVM was a leading computer R&D facility. NIIUVM had extensive experience in industrial computer development and was closely associated with prototype development facilities. Furthermore, IPU and NIIUVM had already worked together on a number of projects. From 1976-1978 groups from the two institutes worked closely together to develop a design which could realistically be implemented. They were able to put together a rough design which called for 64 processing elements housed in five standard computer racks. Although the earliest design proposals called for an associative processing field inspired by the STARAN architecture, this idea was rejected in the draft design completed in 1978 because the components necessary to build it at that time were not available from the Ministry of the Electronics Industry (Minelektronprom), and NIIUVM did not have sufficient capability to design and manufacture the chips itself.

To build the machine, dubbed the PS-2000, financial and material backing was necessary. For several years the designers had been seeking to find potential customers who would be willing to support the project. The design, laying out in rough terms how one might go about developing a machine with a performance rate of 200 million simple addition operations per second was greeted with considerable skepticism, both at NIIUVM and elsewhere. Until that point, the most powerful machines developed at NIIUVM were only capable of 400 thousand operations per second. According to Aleksandr Nabatov, one of the principals from NIIUVM on the project, for the first few years they ‘‘had to work under conditions in which everyone doubted in what we were doing.’’ Thanks to the strong support and involvement of V. V. Rezanov and I. I. Itenberg who had gained

considerable authority through earlier, successful, NIIUVM projects, work on designing the parallel processor continued. Foreign opinion of the machine's design also played an important role. Around 1974, Control Data Corporation had examined premiliary designs at IPU and expressed an interested in some joint work on such a project. The U.S. government vetoed any collaborative work in this area, but CDC's enthusiasm was a big help in convincing Soviet authorities that the project should be supported [Smyk85; Gure85]. Minpribor agreed to fund the project, with the Ministry of Geology and the Ministry of the Oil and Gas Industry serving as principal target users.

While work on the draft design of the PS-2000 was in progress, the Soviet leadership initiated a massive campaign to develop the nation's oil and gas resources [Gust83]. During the 1960s the Soviet Union had enjoyed a steady supply of cheap hydrocarbons, but from the mid-1970s, the Soviet leadership grew increasingly concerned about the state of energy production. The 10th five-year plan, trumpeted at the 25th Party Congress in February 1976, placed great emphasis on coal reserves which were greater than those of oil or gas. When coal production during 1976 and 1977 fell far short of the annual targets, the Soviet leadership perceived a looming energy crisis.

In late 1977, general secretary Leonid Brezhnev announced a sharp change in policy and initiated a crash program to speed up the development of West Siberian oil [Gust83, 29-31]. Beginning in 1978 oil investment was increased sharply, growing by roughly 100% in absolute terms by 1980 [Gust83, 31]. In 1980, the Soviet leadership once again shifted the direction of their energy policy, launching a campaign to increase the output of natural gas by over 50% in five years [Gust83, 1]. In his address to the 26th Party Congress in February, 1981, Brezhnev stated: ‘‘I consider it necessary to single out the rapid development of Siberian gas output as a task of first-class economic and political importance’’ [Gust83, 34].

The NIIUVM/IPU efforts profited from the intense, high-level attention being focused on hydrocarbon exploitation. The oil, gas, and geological ministries had a desperate need for computing power to process seismic data. The SIMD architecture is very suitable for a large number of seismogram processing functions including filtering, tracing, amplitude control, static correction, seismic formation analysis, etc. [Akhm82; Neft91]. The PS-2000 offered a peak performance rate of 200 MIPS, far beyond the capabilities of any indigenous computer then available. The project quickly gained considerable high-level support and was included in the plans realizing the energy campaigns [Akhm82].

While the crisis atmosphere of the oil and gas campaigns generated material and political support for the new parallel computers, it provided a set of constraints which shaped the development of the system as well. In 1978 the PS-2000 was only at the draft design stage. Powerful computers were needed quickly so the PS-2000 had to be built in as short a time as possible. In particular, with the 26th Party Congress looming ahead in February, 1981, it was very desirable that working models be in existence by that time [Trap81, 31]. Furthermore, the systems had to be usable. There was little freedom to experiment with radical approaches or to test new theories.

The pressure to get machines into production in under three years forced a number of design decisions. First, components already in series production had to be used. From the start, PS-2000 designers had sought to create a machine using existing components, but the present time pressures confirmed this approach; there simply was no time to develop new, more suitable chips. The available component base determined the physical size and performance of subsystems and therefore constrained the functionality that could be built into the system. Second, since there was limited time in which to develop hardware, peripheral systems, and a full complement of systems software, the system would have to

be built to make the greatest use possible of existing hardware and software systems. These decisions, discussed at greater length below, included the use of existing, general-purpose host computers, a simplified, linear interconnect system, and a relatively short word-length.

From the outset, IPU and NIIUVM engineers began working closely with the Ministry of Geology, in particular with scientists under Vladimir Kreysberg in the Department of Automated Computer Systems in the All-Union Scientific Research Institute of Geophysics (NIIGeofizika) in Moscow. The geophysicists provided valuable input into the nature of seismic applications, played a significant role in evaluating design decisions and defining the software library, and were responsible for much of the applications software development for the system [Trap79; Impu91, 6]. The first applied program package developed for the PS-2000 was a seismic data processing system which allowed one to obtain time profiles of geological structures. This package was also developed for use in the acceptance testing of the state commission in 1980. During these early years a geophysical job description language was developed to enable geophysicists to interact with the system using geophysics terminology [Trap81, 29-31].

Several PS-2000 prototypes were built between 1978 and 1980 when the machine passed state testing. One year later it entered series production at the Severodonetsk Instrument-building Plant (SPZ). Completed in time for the 26th Party Congress, the machine was touted by the Minister of Minpribor, M. S. Shkabardnya [Shka81].

5.3.2.2 Architecture and Construction

The PS-2000 is a SIMD machine which consists of a four basic parts: a monitor system, a parallel processor (called the PPS-2000) with between 8 and 64 processing elements (PE) (in 8 element blocks), a control unit for the processing elements, and an external memory system. These are shown in Figure 5-1. The PPS-2000 parallel processor can

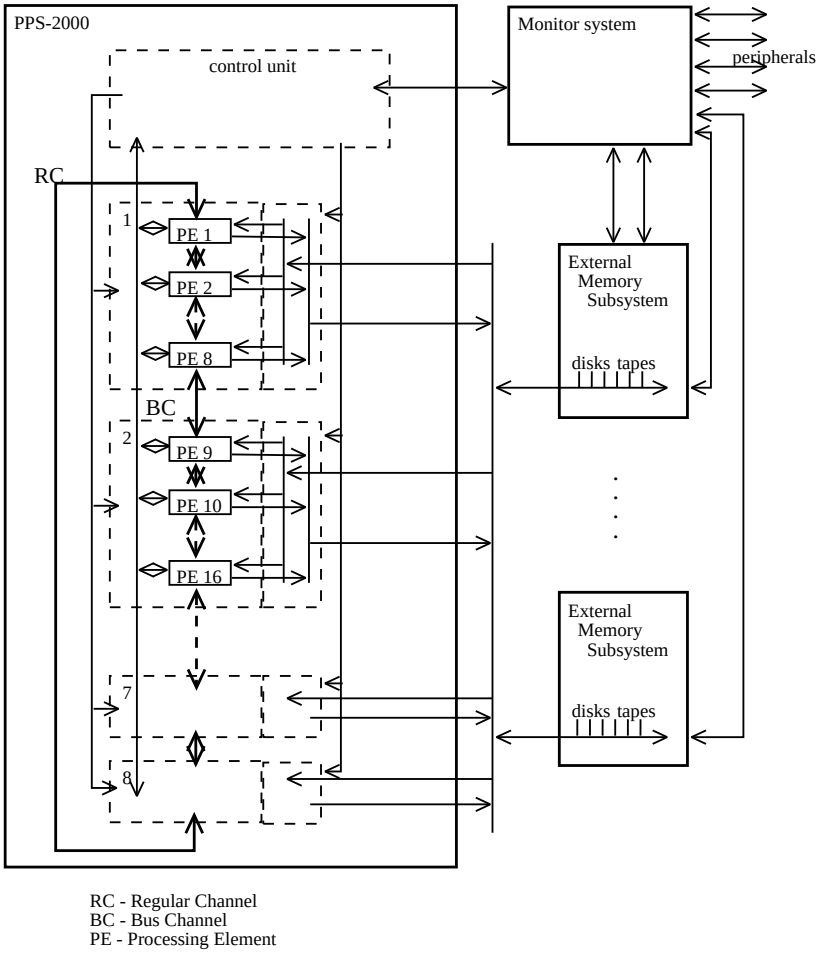


Figure 5-1 PS-2000 Multiprocessor
Source:[Pran83].

be considered a special processor which is attached to the monitor system, a standard control computer. A program runs on the monitor system, but uses the parallel processor to execute special functions such as fast Fourier transforms (FFT), linear algebra operations, etc. The PS-2000 was issued in six standard configurations. Three configurations had 8, 16, and 32 processing elements respectively. Two configurations had 64 processing elements, differing primarily in the amount of secondary storage provided. A dual configuration, containing 2x64 processing elements but only one monitor system was

also installed in a number of locations [Grub89, 183]. Even larger configurations with up to 6x64 processing elements were reportedly installed [Medv92b, 30].

Monitor system. The PS-2000 was designed as a high-performance computing attachment to a standard-manufacture host, the SM-2, SM-2M, or SM-1210. The monitor system provides the primary user interface to the PS-2000. A standard production computer, the monitor system runs the customary operating system (with a few enhancements) and the systems utilities familiar to SM- users. In the PS-2000, the monitor system [Pran83, 114]:

- loads the control unit of the PPS-2000 with microprograms, programs, and constants;
- initiates and terminates execution on the PPS-2000;
- monitors execution on the PPS-2000 and performs diagnostics functions for the entire system;
- executes functions from the engineering console;
- supports the writing and compiling of PPS-2000 programs;
- manages time-sharing in a multiprocessing mode; and
- runs user programs which include routines to be run in the PPS-2000.

The user's application runs primarily on the host computer, the monitor system. Selected routines run on the PPS-2000. The monitor system loads the microprogram routines, initiates them at the appropriate time in the run of the application program, and waits for their completion. Once loaded into the PPS-2000, a routine –which would typically perform an operation like matrix generation, FFT, transformation of a tri-diagonal matrix, etc.–remains resident and is subsequently invoked simply by sending the PPS-2000 the correct parameters to initiate it. If there is insufficient memory in the control

unit to accommodate a routine, systems routines are invoked to swap out one or more of the currently resident routines.

Rather than create specialized hardware and systems software for a host, the PS-2000 designers took as their base the SM-2 and SM-2M with the associated systems software, and merely had to modify these to support interaction with the parallel processor. The ‘‘Aggregate System of Software’’ (ASPO) which included the operating systems and a variety of systems utilities on the ASVT models was modified by including means of loading and initiating the execution of PS-2000 parallel code, a protocol for information and data exchange between the host and the parallel unit, a set of software development tools to facilitate the writing and debugging of parallel code, and a library of widely-used functions designed for the parallel unit [Pran83, 136]. In this manner, the user could operate in an environment with which s/he was already familiar, and the developers did not have to spend a large amount of time re-creating systems software which had been developed earlier. The PS-2000 operating system is a variant of the ASPO disk operating system of the SM-2x machines and could thus operate in real-time multitasking, timesharing, and batch modes [Pran83, 136].

Using a standard host also meant that existing applications written for the SM-2, -2M, and -1210 could be adapted to run on the PS-2000 by modifying only the portions of the application to be executed on the parallel processor.

A pragmatic solution, the idea of attaching the parallel unit to a conventional host computer was by no means a new idea. The ILLIAC IV, the STARAN and other early parallel machines such as the Parallel Element Processing Ensemble (PEPE) used conventional hosts to handle many systems functions.

The SM-2M and SM-1210 could be installed in dual configurations in which two machines shared a common memory space. Two monitor systems could therefore be incor-

porated into a PS-2000 system to improve reliability through redundancy; only one system could function as a monitor at a time, however [Pran83, 135]. The mean-time-to-failure rate for the PS-2000 could reportedly reach 3000 hours.

Parallel processor. A processing element contains:

- a 24-bit arithmetic-logic unit;
- a 24-bit main memory unit of between 4K and 16K words, depending on the capacity of the memory chips;
- a 14-bit index processor for indexed access to memory, incorporating ten index registers;
- an element activation processor for evaluating functions determining whether the processing element is to be activated or not;
- input and output registers which serve as buffers between main memory and the I/O channels.

The 24-bit word-length was chosen chiefly because of space considerations. Implementing a larger word-length would have required considerably more hardware per processing element, more cabinets, greater cost, etc. While technically feasible, larger word lengths were, at the time the PS-2000 was designed and built, not strictly necessary. For the target geophysics applications, 24-bit fixed- and floating-point formats were sufficient. In this respect the PS-2000 differs from the ILLIAC-IV which used 64-bit words to obtain the precision necessary for its intended use, solving partial differential equations [Hock88, 25]. Furthermore, the ILLIAC IV was intended as a computational vehicle for users with a variety of computationally intensive tasks such as linear programming, hydrodynamic simulations, ballistic missile defense analyses, and many more.

The PPS-2000 ALU includes an extension bit which was used to indicate double-precision, 48-bit values. This feature was not incorporated into the earliest units, appar-

ently, for [Trap81, 20] claims that a 48-bit fixed-point processing format ‘‘is envisioned.’’ Being an add-on to the original design, operations on such values run very slowly, relative to the 24-bit operations. While each processing element executes one 24-bit register-register add per 320 nsec clock cycle and one 24-bit floating-point operation in three cycles, double-precision floating-point additions requires approximately 143 cycles [Impu89, 7].

The index processor of the memory module includes an ALU, a result register, a memory address register, and ten index registers which can be loaded in parallel. The use of multiple index registers gives the processing element a flexible, multi-level memory addressing scheme which allows many independent data exchange processes to occur simultaneously in different areas of memory without the need to reload index registers or use multiple operations [Pran83, 119].

The ILLIAC IV processing elements incorporated an enable/disable mode which could be set directly by the control unit or on the basis of a comparison of the contents of some registers, but the PS-2000 activation processor greatly expanded on this capability. The activation processor can enable/disable a processing element on the basis of logic operations performed on five internal registers or the results register of the ALU. It can also modify the links between adjacent registers, control data reception and transmission sequences on the bus channel, control the modification of the address data in the index-math processors, carry out associative attribute-based data search operations, and more [Pran83, 119-120]. The idea of building a parallel processor with associative memory as in the STARAN had been abandoned because of the lack of adequate components with which to build it. A remnant of this idea survived, however, in the form of associative capabilities in the activation processors.

The interconnection scheme for the processing elements is a novel features of the PS-2000 and it differs considerably from that of the ILLIAC IV. The ILLIAC IV used a nearest-neighbor interconnection scheme in which each processing element was directly connected to four others. Routing between processing elements took place via a ROUTE instruction which used a circular shift mechanism. This mechanism can be thought of as a circle with 64 slots in it. Each processing element can (optionally) place a data element in a slot. Then the entire circle is rotated some number of slots, and the data unloaded at the destination processing element. While this scheme can transfer up to 64 words at a time, it is restrictive in that all data items are moved the same fixed distance [Hord90, 31].

The PS-2000 uses two different interconnect channels to link processing elements, and a third, a data channel, to link the processing elements directly to external storage. A so-called ‘‘regular channel’’ provides a parallel link between each processing element and its two neighbors, forming a circular chain of processing elements. During *n* cycles, it can shift values which are located in identical registers, or in identical locations in memory in different processing elements, *n* processing elements to the right or left [Trap81, 29-31]. This mechanism is similar in concept to that of the ILLIAC IV, and suffers from the same restrictions. A so-called ‘‘bus channel’’ provides a serial link between each processing element and the control unit. The bus channel has a broadcast nature in which only one processing element (or the control unit) can send a data word at a time, but any number can receive it.

The PS-2000 approach differs principally from that of the ILLIAC IV in the reconfigurable nature of the links. The PS-2000 gets its ‘‘reconfigurable structure’’ (*per-estraivayemaya struktura*) designation from the fact that these channels can be segmented and software reconfigured into 8, 4, or 2 segments consisting of 8, 16, or 32 processing elements, respectively [Berk82; Pran83, 119]. This serves two primary purposes. First, it

becomes easier and faster to transfer data which is localized to small(er) sets of processing elements. Second, it makes it possible to broadcast different data elements to different processors over the bus channel.

When designers discussed how the processing elements should be interconnected, they naturally considered the four-way scheme used in the ILLIAC IV. Such a scheme was not feasible given the manufacturing technology available to them, however, since the number of interconnections was large, and would have required too much hardware to implement. They therefore considered linear interconnect structures and examined a scheme in which all 64 processing elements were linked in a single chain. They soon realized that considerably greater flexibility and performance could be achieved if instead of using one single chain they used a set of linear rings which could be combined to form larger rings.

Such an arrangement would have been worthless if it did not adequately support the information flows in the parallel algorithms for which the PS-2000 was designed. Designers asked the algorithm specialists to analyze the proposed configuration and determine to what degree their linear approach was worse than a four-way matrix interconnect. This interconnect did prove to be somewhat worse than the matrix scheme to no one's surprise, but it did prove acceptable, especially considering the hardware constraints they were facing. Certain algorithms such as the FFT in fact map well to sets of processing elements with a ring interconnect.

Control unit. The control unit is linked through a serial channel to the monitor system. Its primary function is to control the execution of processes on the parallel processor. In addition, it receives commands from the monitor system, executes them, and manages the exchange of data between the PE memories and monitor system or external memory.

The control unit processes two types of instructions, each of which can be downloaded from the monitor system. One kind, 64-bit microinstructions, provides the control information that each PE needs to perform some action. The microinstructions, in some respects similar to horizontal architectures, simultaneously contain operators for the processing element ALU, the activation processors, the regular and bus channels, the memory unit, the index-math processor, and all microinstruction processor devices in the control unit. Thus the microinstructions can contain information about how data is to be moved among registers and memory within each PE, the arithmetic-logic operations to be performed, the activation status of a PE, transmission on the various channels, etc. The microinstructions allow complex operations to be represented in a single expression. For example, one such microinstruction can mean: ‘‘in device S add the contents of the R3 and B registers in each processing element, store the results in the C and M registers of the processing elements for which (T1 or T3 or T4) is true and jump to label M2 if the contents of the U register are less than or equal to the contents of register I2.’’ This control information is transmitted to each PE via a control signal bus. There are various formats of microinstructions which execute in 1-, 2-, or 3-cycles. Most microinstructions execute in one 320 nsec cycle, however [Pran83, 30-31].

The control unit also contains a data channel interface through which passes all data going to processing elements from the monitor system, or vice-versa. This interface makes the conversion between the 24-bit data used by the parallel processor, and the 16-bit used by the monitor system and external memory. This interface contains an input channel which has a throughput of 1.8 Mbytes/s and an output channel with a throughput of 1.4 Mbytes/s.

External memory. The ILLIAC IV used 16 Mbytes of disk memory, directly accessible from each processing element, as main memory. Access to additional external stor-

age took place through the minicomputer host. The PS-2000 provides processing elements with direct access to external memory subsystems through an I/O bus. The notion of ‘‘main memory on disk’’ is not present. Each set of 8 processing elements has an input channel and an output channel onto an I/O bus which leads to one or more external memory subsystems. Input and output channels can be linked together to form sections of 16, 32, or 64 processing elements. Each external memory subsystem contains a maximum of two disk-drive controllers and four tape unit controllers for a total of four disks and eight tapes. Each complex can execute two transfers (one input and one output) simultaneously. PS-2000 configurations with 64 and 2x64 processing elements were equipped with 16 ES-5061 (29 Mbyte) removable-disk drives and 16 ES-5012 tapes [Grub89, 183].

Alternately, data exchange with peripheral storage can take place via the monitor system’s memory. The monitor executes I/O instructions storing data in main memory; this data is passed to or received from the parallel processor through standard channels. The monitor system can in this case be used as a sort of filter to selectively send data to the processing elements [Trap81].

Data exchange with external memory has proved to be a serious barrier to high performance. In 1983 the system used 29 Mbyte disks with data transmission rates of 312 KBytes per second. To achieve the highest I/O rate, the I/O channels should be segmented so that each group of eight or sixteen processing elements has independent input and output channels. In a maximum configuration up to four disks can feed data to the processing elements at an aggregate rate of 1.2 Mbytes/sec [Pran83, 134-135]. The input and output channels from each set of eight PEs have a combined throughput of 3.2 Mbytes/s (1.8 Mbytes/sec input, 1.4 Mbytes/sec output), or 25.6 Mbytes/s total for eight sets of eight processors. If the aggregate input transmission rate is 1.8x8 = 14.4 Mbytes/sec, then the bottleneck is clearly the data transmission rates of the magnetic

disks. Later installations such as that at the Tsentrprogrammsistem software distribution center incorporated ES-5067 (100 Mbyte) disks with transfer rates of 806 KBytes/sec. Even these would be able to feed the processing elements at a rate of only 3.1 Mbytes/sec, a fraction of what the I/O busses can accommodate.

5.3.2.3 PS-2000 Success

The PS-2000 proved to be a highly successful computer, by Soviet standards. Only four years passed from the start of prototype development in 1976 to state testing. Series production was achieved only a year later. Between 1981 and 1989, approximately 200 units were manufactured at SPZ and used not only for geophysical applications, but also weapons design, mission control for the space program, metrology, medical diagnosis, atomic energy plant operation, etc. [Pipk83]. In 1988 the machine was nominated for (but did not receive) a State Prize in Science and Technology for its contributions [Izv880329]. The PS-2000 was a popular machine. It filled a high-performance computing niche which desperately needed computing resources. Other Soviet computers in series production during the early 1980s, mainframes, offered general-purpose processing capability, but had a peak-performance of less than 10 MIPS. It reportedly offered 5-10 times better performance on many applications than systems with attached array processors (see section 7.12) [Medv92b, 21]. The El'brus-1 was, for all but a small number of users, non-existent, and the El'brus-2 had not yet gone into series production. The PS-2000 was available, usable, reasonably reliable and could be purchased for several hundred thousands of rubles, a modest sum compared with high-end mainframes and the nascent El'brus computers which cost several millions of rubles. [Trap79; Pran83, 455; Mnw84; Smyk85].

The PS-2000 was successful for a number of reasons. First, the developers were experienced, capable computer engineers who had a good understanding of computer use in

industry. Having designed computers since the late 1950s, NIIUVM researchers had much experience in developing computers geared towards use in industrial applications. They worked together with the IPU computer scientists to translate the ideas of the IPU ‘‘ideologists’’ into a viable design. The project was very much a joint effort. Every couple of months a team from Severodonetsk would travel to Moscow or vice versa to review alternatives, devise compromise solutions, make design decisions.

Since NIIUVM was part of the same association as the factory, SPZ, it had ready access to the same tools, materials, and components available to the factory. The prototype could therefore be built with an orientation towards production, shortening the time needed to put the prototype into series production.

Second, by using components already in existence, the length and complexity of the development cycle could be reduced considerably. Projects which push many technological boundaries simultaneously consume greater developmental resources, contend with greater uncertainties in supplies, are difficult to construct and debug because of the levels of new technology, and are very likely to experience significant delays.

Third, for its performance, the PS-2000 was not a terribly complicated machine to produce. The SIMD architecture required the manufacture of up to 64 identical processors per unit. With this degree of replication the cost per processor of setting up production was reduced. Since production quality for a given manufacturing line generally improves over time as the problems are ironed out, the high degree of replication of boards enabled stable quality levels to be achieved more quickly than would have been the case for a machine with little duplication of boards [Akhm82].

Fourth, the project had very focused goals. Developers had to create a machine which could be produced and used, and had to do so in a short period of time. These constraints

served to focus the development effort and further keep the design within the bounds of what was feasible.

Fifth, the project enjoyed considerable high-level support. This ensured that the necessary links with Minelektronprom would be established expeditiously, that requisitions for development equipment and other necessary materials would be provided, and that the necessary monetary funds were allocated. The strong demand for the machines and political pressure also served to ensure that once prototyped the machine would be manufactured.

A sixth factor, difficult to quantify but shaping both Soviet policy towards computing and the market for the PS-2000, was the effect of Western export control policies. During the mid-1970s Control Data Corporation was selling a number of Cyber 17x computers to the Soviet oil and gas industry. During the last year of the Ford Administration, CoCom relaxed some restrictions on the export of computers to the Soviet Union and the U.S. administration approved the export of computers for use in Soviet air traffic control systems and production planning in the Kama River Truck Plant (KamAZ). Amendments passed in 1977 to the Export Administration Act of 1969 served to lessen export control restrictions by, among other things, calling for more expeditious handling of export applications, shifting emphasis away from shipments to ‘‘Communist’’ countries to emphasis on the commodity to be exported, and limiting the grounds under which the Secretary of Defense could recommend against export for national security reasons [Rich80, 165-167].

When Jimmy Carter became president, relations between the superpowers chilled considerably. Growing Soviet activism in developing countries and a military buildup which had given the USSR rough military parity with the U.S. had tarnished relations between the two countries. With his emphasis on human rights, and faced with the need to

counter Soviet initiatives throughout the world, President Carter adopted a more confrontational approach.

Beginning in 1977, a series of executive actions and policy statements by both the Carter and Reagan administrations made it more difficult for the Soviets to acquire high-technology. In 1977 the Carter administration canceled the sale of a Cyber 76 to the Soviet Union's weather research center on grounds that it was ''a scientifically oriented computer that has wide uses in the United States for military research, development and support'' [Nyt770624; Rich80]. In July, 1978, the Administration canceled the sale of a Sperry Univac computer to the Tass press agency for use during the 1980 Olympics. The reasons behind the cancellation were not only a claim that the Soviets might use the system for military purposes, but also a protest of the trials of two dissidents, Anatoly Shcharanskiy and Aleksandr Ginzburg, in Moscow [Burt78b; Wsj790328]. The export of a Cyber 173 for seismic research was turned down when the Department of Defense withdrew its support in 1979 [Rich80, 179]. When the Soviet Union invaded Afghanistan, U.S. export control policy shifted significantly. While the earlier measures were rulings on specific export licenses for specific machines, Carter's response to the Soviet invasion, announced on January 4, 1980, included a cut-off of sales of high technology such as advanced computers and oil-drilling equipment until further notice [Cart80]. The tightened export controls on computing remained in place when Reagan assumed the presidency and were further strengthened after the military crackdown in Poland, for which Reagan held the Soviet Union responsible. ''The issuance or renewal of licenses for the export to the USSR of electronic equipment, computers, and other high-technology materials is being suspended'' [Reag81].

The PS-2000 was a source of great pride in the Soviet Union for it provided the Soviets with a system which could be used as an example of technological achievement at a

time when the Soviet government felt considerable pressure to demonstrate its ability to develop all the technology it needed for its own uses. Numerous reports in the domestic and foreign press proclaimed the Soviet Union's ability to proceed with world-class technological developments in spite of a moratorium on technology transfer from the West. R. Akhmetov's comments in *Sotsialisticheskaya Industriya* about the use of the PS-2000 in the oil and gas campaign were typical of the tone which characterized these reports: ''Incidentally, the Reagan administration, which has blocked the exchanges with the USSR of any information on computer technology, has done this to the detriment of science in its own country. More and more American scientists and specialists justly believe that USSR achievements...cannot continue to be ignored'' [Akhm82]. The Moscow World Service in 1984 claimed that the PS-2000 was ''quickly developed'' after the American administration imposed trade restrictions on the sale of computers to the Soviet Union [Mosc84]. The PS-2000 project began before Carter's more comprehensive restrictions in 1980, but a recognition of the writing on the wall following the Cyber 76 cancellation certainly played a role in advancing the project [Suga77].

The U.S. export controls also served to bolster demand for the PS-2000. When sales of computers and spare parts by Control Data Corporation–many to the Ministry of Geology–''dried up'' following the tightening of export control in 1980, the PS- computers lost a major competitor [Bosg811230]. ''If the flow of these [CDC] machines had been without restrictions, then perhaps we would not have had any customers. To some degree [the export controls] helped us'' stated A. S. Nabatov.

5.3.3 PS-2100

5.3.3.1 History

During 1981-1982, NIIUVM engineers worked primarily on installing and popularizing the PS-2000. During the first year of production the production process was still being perfected and customers needed considerable hand-holding to bring up their systems, keep them working, and learn to program and use them effectively.

In 1982, however, the NIIUVM engineers began to consider building a PS-2000 successor called the PS-2100. There were a considerable number of orders for the PS-2000 and the initial field results confirmed the system's usefulness. There was a demand, and there was reason to believe that there would be a demand for a successor as well. This work was carried out solely at NIIUVM without the participation of IPU researchers.

The early phases of development involved a careful assessment of both the requirements for a new machine and the current and future capabilities of NIIUVM, the production factory, and the supporting industries. The principal sponsor of the PS-2100 remained the Ministry of Geology, through NIIGeofizika, and its requirements continued to dominate the formulation of the technical statement of work. But during the design phase NIIUVM engineers made a concerted effort to broaden the market through an analysis of the nature of the computational problems faced not only by the geophysicists, but also by many of the other Impul's customers in the atomic energy, chemical, and other industries. For some months representatives from NIIUVM traveled around the country and held seminars in Severodonetsk to meet with actual and potential users of the PS-2x00 machines to analyze their applications and the associated hardware and software requirements. These requirements were packaged into the ‘‘*tekhnicheskoye zadaniye,*’’ or technical statement of work, which had to be approved by the primary sponsors, the geophysicists.

At the same time NIIUVM engineers made a careful evaluation of what technology–components, cabling, production tools, design tools, power supplies, etc.–was available. They determined which components and subsystems already in production could be used and which would have to be developed. In the latter case, they determined which of the missing components and subsystems could be developed in-house, and which would have to be developed by others.

In reviewing their work on the PS-2000 they concluded that the PS-2000 architecture used components effectively enough that a new machine built using the same MSI component base would yield only marginally better performance. The new machine should be built with a new generation of components.

In 1982, however, large-scale integrated circuits were not yet available. The component base was manufactured in another ministry, Minelektronprom, which, according to developers, was overloaded with orders from all quarters. Although the centralized planning mechanisms could specify that new types of production were to be initiated, if a factory had more total orders than it could fulfill, the director had considerable *de facto* authority to decide which part of the plan was to be met and which was to be disrupted. It was more in factories' interests to manufacture chips with small- and medium-scale integration. Production for these was well established and they were being ordered in sufficient quantities to keep factories producing near capacity. By minimizing disruption in a factory's production, higher volumes of production could be sustained, and the political fallout for factory directors for missed production targets could be minimized. To be sure, Minelektronprom was manufacturing components with large-scale integration for powerful military, or military-related customers. Insufficient numbers of such components were available for Minpribor's non-military projects, however.

During 1983-1984, Itenberg's team searched for an appropriate component base for the PS-2100. Finding nothing suitable from Minelektronprom, they formed a branch at NIIUVM to work on the development of integrated circuits. Lacking the technology to manufacture chips themselves, they experimented with hybrid technology, since this would allow them to reduce the amount of space occupied by components without having to use higher levels of integration in the circuits themselves. Hybrid technologies were strongly emphasized throughout Minpribor during the 11th Five Year Plan (1981-1985) [Gavr86, 42]. A hybrid integrated circuit is similar in concept to the printed circuit board and consists of multiple components interconnected on a single ceramic substrate. Unlike the components used on a printed circuit board which are each encapsulated in rather large packaging (relative to the size of the circuit itself), components in a hybrid circuit are devoid of packaging and can therefore be combined in a smaller physical space. Hybrid technology, now often refered to as multi-chip modules, has been used for many forms of analog circuitry, but until relatively recently has not been very popular for digital circuitry [Bras83, 773; Muku93]. One package developed at NIIUVM could contain up to 10 MSI chips. The engineers at NIIUVM developed in particular some mock-ups of hybrid integrated circuits oriented towards floating-point operations, because they had no such hardware at all at that time or for many years following.

In 1984 NIIUVM researchers learned that gate arrays were to become available from Minelektronprom in the near future. Gate arrays are a regular configuration of logic gates whose interconnects can be relatively easily customized to produce special-purpose chips. Their advantage is that without changing the underlying manufacturing technology, a wide array of chips can be manufactured simply by using a lithographic mask unique to each different kind of chip. Although their levels of integration are not necessarily as high as custom-made chips, they provide a very cost-effective solution to system

development. Gate arrays have been popular as building blocks for many, if not most, Western computers.

In 1985 NIIUVM researchers began designing specialized integrated circuits and over a two year period developed seven types of gate array based chips which would be incorporated into the PS-2100. In the West at this time, chips based on gate arrays with on the order of 400 gates could be developed in a matter of months. There were a number of reasons why the NIIUVM efforts took more time. First, the technology was new to the engineers although processor design was not; there was a learning curve that needed to be climbed. Second, the base technology itself had not been perfected by Minelektronprom. At the time NIIUVM engineers started designing their chips, Minelektronprom's Elektronika Scientific-Production Association in Voronezh' was manufacturing experimental units. The electronics industry understood the technology, but had not yet mastered production. Third, development was slowed by a lack of sophisticated design tools. Such tools were developed both at NIIUVM and by a Lithuanian firm, but this took time as well.

The first prototype gate arrays for the PS-2100 were obtained at the end of 1987. They were correct, so at the beginning of 1988 they were manufactured in sufficient quantities to construct a complete base module. Several units were constructed during 1988, and the PS-2100 passed state testing in December, 1988 [Bere88]. By September, 1990, 15 PS-2100 base modules had been constructed and full series production began in late 1990 or early 1991.

5.3.3.2 Requirements

From their analysis of current and prospective applications, a number of architectural requirements emerged as dominant. Not surprisingly, higher performance was key. All the applications being considered, both the traditional seismic data processing as well as

real-time image processing and process control, etc. needed higher processing rates, more memory, improved I/O, faster communications channels, and so forth. Designers realized they needed to concentrate on particular bottlenecks in the PS-2000. These included the inter-processing element transmission time, the efficiency of the object code, the time required to configure the system for a particular problem, and the access time to secondary storage. Of these, the time required to transmit data between secondary storage and the processing elements was the most severe bottleneck.

A second dominant requirement was reliability. Reliability had always been a primary concern in Impul's computers since most of them had been oriented towards real-time processing control applications. Although the primary application of the PS-2000 was seismic data processing, it too had real-time capabilities and could be configured–such as in the dual configuration–to give it a mean-time-between-failure (MTBF) of reportedly 3000-5000 hours. For many customers whom NIIUVM was courting, reliability was even more important than high performance and MTBF rates approaching half a million hours became development goals. The emphasis on reliability was especially acute in the atomic energy industry [Impu91, 7]. According to Itenberg, the actual MTBF for the PS-2100 is several tens of thousands of hours.

The use of 32/64-bit data formats became a third major requirement. While 24-bit operands had been acceptable for geophysical applications during the late 1970s, they did not have the precision needed for modeling, solving systems of linear equations, etc. The shift to a 32/64-bit format was also attractive because these word-lengths had become standard in high-performance computing systems throughout the world.

These requirements demanded that changes be made to the system, but these changes were to be made based on the existing SIMD-oriented approaches. This was necessary for two primary reasons. First, a real demand for such machines existed, and would continue

to exist for the next generation machines. In particular, the geophysicists who sponsored development had found the basic architecture to be a sound one for their purposes. Second, the existing installed base of PS-2000s made it necessary to retain a large measure of compatibility between the the PS-2000 and PS-2100. Radical changes to the basic architecture were, early on, ruled out. The SIMD architecture had proven its usefulness in many applications; the fundamental nature of these applications had not changed, so a SIMD approach would continue to be useful. As A. S. Nabatov put it,

> ...in our collective we understood that we had to continue with the SIMD architecture as long as the market for it exists. There were lots of discussions and ideas about how to upgrade this architecture. The fight was not regarding the architecture, but how to implement [it] [Naba91, 5].

As long as there were no fundamental problems with the basic SIMD approach and a demand for machines of this sort existed, work in this direction could be justified. Reinforcing this perspective was the fact that lacking a compelling reason to change, engineers are likely not to spend a large amount of time and effort considering radically different alternatives. They are likely to continue introducing incremental modifications to familiar approaches. When asked what they would change if they had the liberty to build the system from scratch with no compatibility requirements, Nabatov and Itenberg answered that

> ...it's a very complex question. You understand, we have already become accustomed to this type of architecture. We think that it is rather promising for the future under our conditions, and in the West, too, a number of machines of this type of architecture–mostly experimental–are being developed. With regard to programmability, we would change part of the mechanism, but this is perhaps a question of small-scale changes in the architecture. Perhaps...I haven't thought about this...we would move in a very different direction, but still [stay] within the area of parallel architectures [Iten90b, 42].

The installed base created pressures for preserving compatibility between generations of systems. Considerable volumes of software had been written for the PS-2000 and, naturally, users strongly resisted any efforts to alter the PS-2000 structure in a way that would require that they rewrite their software. To accommodate the peripheral devices which were to be attached to it, the PS-2100 naturally had to accommodate established peripheral interfaces.

5.3.3.3 Architecture and Construction

The PS-2100 is compatible with the PS-2000, but only at the level of ASPS, not at the level of executable code [Iten90d, 12]. ASPS is a hybrid language, having low-level features typical of assembly languages such as register access and bit-wise operations and high-level constructs such as DO-loops, complex data types, IF-THEN-ELSE clauses. It also contains instructions for managing the flow of data between processing elements [Ryad88]. The chief sources of incompatibility were changes in some of the register structures and data formats of the new machine. The decision to maintain compatibility only at the level of ASPS reflects the tension between the need to improve the performance of the machine, expand its capabilities, and preserve the existing software base. It also reflects certain characteristics of the PS-2000 and its software development which made a lack of compatibility at the lowest levels more tolerable.

Moving software from the PS-2000 to the PS-2100 requires recompiling existing software; but only software written in the lowest-level language, MNEMOKOD, would have to be rewritten into MIKROKOD PS2100, the microprogramming language for the PS-2100. While some users did write their own routines in MNEMOKOD, the bulk of MNEMOKOD programming, an arduous task, was done at NIIUVM. Programmers here were responsible for the library of approximately 200 routines and systems software which execute on the PPS-2000. The applications programs were written chiefly by users,

but these typically were written at least at the ASPS level and relied heavily on the parallel routine library for code which actually ran on the PPS-2000. To be sure, some users did develop some code using microcode [Ivan85], but the number of programmers with sufficient knowledge of PS-2000 to write code at this level was probably not great. Finally, much of the systems software ran not on the PPS-2000, but on the host computer. In sum, the amount of code that needed to be rewritten was relatively limited and was chiefly the responsibility of NIIUVM programmers. The cost to applications developers and users, apart from recompilation, would be bearable. The PS-2100 engineers felt that the cost to their own programmers was tolerable, and worth the gains in performance that could be achieved.

Besides some low-level changes, the basic architecture of the PS-2000 remained intact. The PS-2100 architecture is best understood as a set of extensions to the PS-2000 architecture to improve performance and reliability. Table 5-2 compares the PS-2000 and the PS-2100.

The PS-2100 consists of the following basic parts [Impu89, 4-5]:

- 1–10 so-called ‘‘base modules’’ (BM), each containing 64 processing elements;
- switching system for intra-system data exchange;
- external semiconductor memory (SSM);
- external magnetic disk memory subsystem;
- external magnetic tape memory subsystem;
- monitor subsystem consisting of one or two PS-1001s;
- simplified configuration subsystem for dual systems.

The PS-2100 is shown in Figure 5-2.

	PS-2100	PS-2000
Data Formats (bits)		
fixed-point	16;32	12;16;24
floating-point	32;64	24;48
Number of base modules in configuration	1-10	1-2
Ability to join base modules in configuration into a single processor field.	yes	no
Number of processing elements in base module	64	8-64
Maximum main memory per processing element (bytes)	128/512K	48K
Theoretical peak performance of one processing element (base module):		
fixed-point addition (MIPS)	2.38 (150)	3.125 (200)
floating-point addition (single precision) (Mflops)	1.02 (65)	1.04 (67)
floating-point addition (double precision) (Mflops)	.12 (7.6)	.02 (1.39)
fixed-point multiply (single precision) (Mflops)	.40 (25)	.45 (29)
floating-point mult. (single precision) (Mflops)	.51 (33)	.35 (22)
floating-point mult. (double precision) (Mflops)	.07 (4.2)	.04 (2.3)
Throughput of one data channel (Mbytes/sec)	6	1
Total throughput of all data subchannels (Mbytes/sec)		
in one base module	48	25
in maximum configuration	480	50

Table 5-2 Comparison of the PS-2100 and PS-2000
Source: [Impu89]

Base modules. The architecture of each of the base modules is virtually the same as the PPS-2000 [Impu89, 6]. Each base module operates under SIMD control and contains the same types of reconfigurable channels interconnecting processing elements as the PPS-2000. During the early design phase of the PS-2100 a series of discussions were held about how these transmission channels might be changed, since during the development of the PS-2000 it was clear that the linear, reconfigurable rings were not optimal. As with

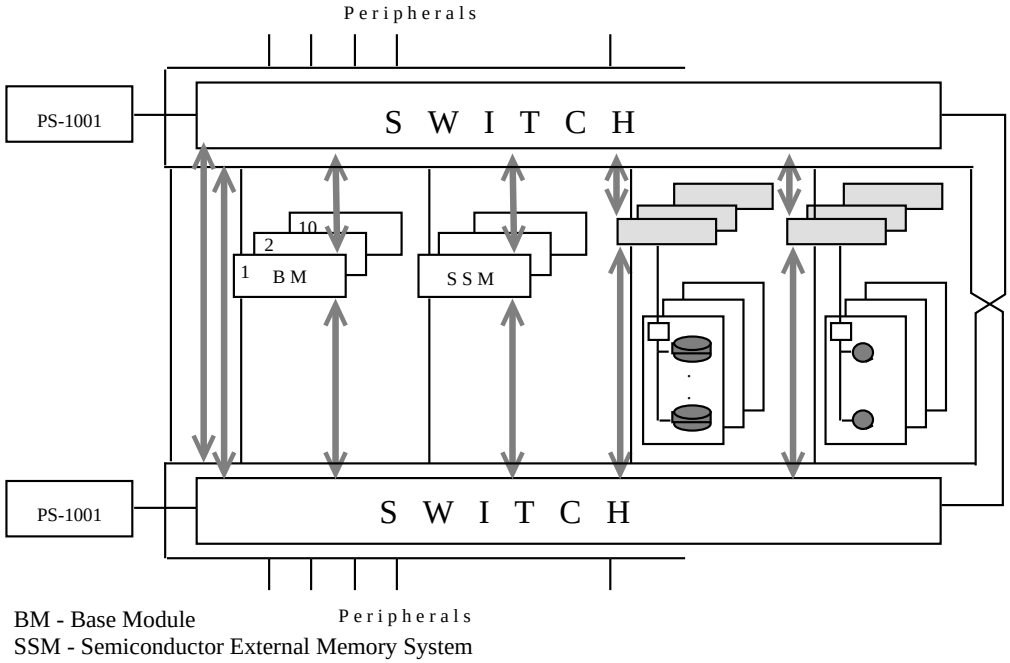


Figure 5-2 PS-2100 Multiprocessor
Source: [Impu89]

the PS-2000, the constraints of construction played a determining role. As Nabatov put it [Naba91, 6]:

> Each time we re-examined the possibilities [for changing the transmission channels] at the initial stages of each development....But the closer we got to the concrete implementation, the less willing we were to make a step forward in the area of lines....[I]t is a question of the construction and technology.

While improvements in technology made it possible to reduce the amount of hardware needed for a processing element, it also reduced the number of external connections to the processing element. The transmission channels have a fixed bus-bit width, so the fewer the external connections the more difficult it becomes to implement large numbers of transmission channels. In the PS-2000, eight boards were needed for one processing element. Together these offered on the order of 1200 external connections. The PS-2100 processing elements reside on one board, so the number of connections is reduced by a

factor of eight. For this reason, improvements in the component base have not made it easier to increase the number of transmission channels between processing elements, in spite of a desire, in principle, to do so. Similar constraints are encountered in the PS-2300 (discussed below) in which each processing element will reside in one chip with 132 (128 data) outputs. Such constraints limit the degrees of freedom designers have to develop creative interconnection schemes.

In contrast to the PS-2000 in which only one instruction is sent from the control unit to the processing elements, a setup mechanism is used in the PS-2100 base module to increase the rate at which instructions can be sent to each processing element. In addition, the base modules are equipped with enhanced features for real-time operation: an interrupt system and a real-time timer.

Up to ten base modules can be configured in a single system. Each base module can execute an independent instruction stream, making the PS-2100 as a whole a MIMD/SIMD architecture. The ten base modules can execute either separate programs, different parts of the same program, or the same part of a single program on different data sets.

The ability to configure 1–10 based modules together serves a number of purposes. First, it makes it possible to increase the theoretical peak performance of a system by a factor of ten. Each base module has a theoretical peak performance of 150 MIPS; a full configuration, 1.5 GIPS [Impu89, 7; Iten89, 3]. The base module approach makes it possible to tailor a configuration to the performance requirements of individual users, thus making it possible to serve a broader market. In this respect the PS-2100 continues the principle incorporated into other Impul's machines.

The base module approach also makes it possible to increase significantly the reliability of a configuration through redundancy. Individual subsystems, including the base

modules, can be switched off and the configuration automatically reconfigured via a sliding backup mechanism [Impu89, 15]. In addition, when a processing element fails within a base module, the half of the base module containing the failed processing element is switched off and the job completed on the remaining half or on another base module [Impu89, 15]. To reduce the amount of down-time, each PS-2100 base module comes equipped with extra boards of each type. The user does not therefore have to wait for a replacement board to arrive from Severodonetsk but can replace a defective board immediately.

Internal exchange switch. To provide the flexibility and high data throughput needed in configurations with multiple base modules, engineers developed a so-called ‘‘internal exchange switch.’’ This switch, organized in a matrix configuration and based on the K1509KP1 switch developed at the Scientific Research Institute of Multiprocessor Computing Systems in Taganrog [Iten89, 2], permits data exchange between arbitrary systems components which are attached to it. It can support the simultaneous transmission of up to 31 data streams. The internal exchange switch comes in two configurations. A single-section switch contains 11 ports; a dual-section switch, 31. A synchronous interface at each port permits full duplex transmission. At any given moment, devices at any two ports can exchange data at a maximum rate of six Mbytes/sec [Impu89, 8-10]. The switch is tuned via instructions issued by the monitor subsystem.

I/O and external storage. Relieving the severe I/O bottleneck required multiple solutions at different levels of the architecture. A fundamental constraint was the capabilities of the secondary storage available to PS-2100 manufacturers. The most powerful Eastern Bloc disk drives in series production were the Bulgarian ES-5063 317.5 Mbyte

drives. Produced since the mid-1980s, these drives have a data transmission rate of 1.25 Mbytes/sec [Impu89].[1]

Magnetic disk subsystems consisting of up to six ES-5063 disks attached to a single ES-5563 disk controller and each controller can be attached to one port of the internal exchange switch. This transfer rate per disk is approximately four times faster than that of the ES-5061 disks used in the PS-2000, although throughput from a disk subsystem is limited by the capabilities of the controller. Each controller can probably sustain read operations from not more than one or two disks simultaneously, putting an upper limit of 2.5 Mbytes/sec transfer rate per disk subsystem. The number of such subsystems is variable, theoretically limited only by the number of available ports in the internal exchange switch. In practice, the maximum configuration was equipped with four subsystems, each containing six disks for a total of 7.6 Gbytes [Iten89, 2-3].

The PS-2100 has fewer, but faster, I/O channels per set of 64 processing elements than the PS-2000. Each base module has four pairs of I/O channels, each accessible by 16 processing elements. Each channel has a throughput of six Mbytes/sec [Impu89, 6]. The aggregate throughput of the I/O channels for one base module is therefore 48 Mbytes/sec (relative to 25.6 Mbytes per second for a 64-processor PS-2000 configuration) so the gap between the capability of the disks and the I/O channels was reduced, but remained significant [Impu89, 8]. In contrast to the PS-2000 I/O channels, those in the PS-2100 have multiplexed subchannels, allowing greater flexibility in data exchange with external memory [Iten89, 1]. The throughput of the magnetic disks could not be increased, since larger capacity disks such as the 635 Mbyte ES-5065 were not being produced in large enough quantities to be accessible to Impul's [Vtp89, 19; Gors89, 4].

[1] Other sources such as [Dani84b; Dani86] put the transfer rate at 1.198 Mbytes/sec, comparable to the 317.5 Mbyte IBM-3350 disk drives which were introduced in 1976 [Rdd82, 5; Dpro82, 40].

To compensate for the slow disk storage, designers greatly increased the amount of semiconductor memory. Most significantly, they added a semiconductor external memory unit consisting of blocks of 16 Mbytes each. Over a dozen such blocks could be installed, if desired, but as a rule 1–6 were [Impu89, 12]. External solid state memory devices have been used widely in Western supercomputers to provide data transfer rates high enough to keep the central processing units loaded. The PS-2100 is the first Soviet high-performance computer to use such a device, however. The semiconductor memory has a peak transfer rate for read operations of five Mbytes/sec through each of four ports which interface to the internal exchange switch for an aggregate rate of 20 Mbytes/sec [Impu89, 12]. This nearly fully loads the four input channels internal to the base module.

A second means of relieving the I/O bottleneck involved increasing the amount of local storage in each processing element. While each PS-2000 processing element contained up to 48 KBytes of local memory, each PS-2100 processing element contains 128-512 KBytes, depending on whether 64 Kbit or 256 Kbit memory chips are used.

The ability to expand the PS-2000 architecture by multiplying subsystems was largely facilitated by improvements in construction which permitted greater functionality to be packaged in a smaller space. In the PS-2000, each processing element occupies eight boards. A 64-processing element configuration occupied five cabinets [Trap79; Impu89, 8]. In the PS-2100, 6-8 layer boards were used in place of the PS-2000 2-layer boards, and gate arrays with 400 gates per chip made it possible to place an entire processing element on a single board. As a result, an entire 64-processing element base unit could be housed within one cabinet. This in turn made it feasible to incorporate multiple base modules in a single configuration. The need to manage a growing number of possible data transmission paths between the base modules and the peripheral units necessitated the creation of a new form of intra-system data exchange, the internal exchange switch.

5.3.3.4 Performance

The gains in processing speed came primarily from these extensions to the architecture rather than from increased processing rates in each processing element. The size of each processor was reduced making it possible to reduce the clock time from 320 nsec to 140 nsec. The number of cycles to perform basic operations was increased, however, largely to accommodate the increased word-length. While a single-precision fixed-point addition in the PS-2000 required one 320 nsec cycle, a similar operation in the PS-2100 required three 140 nsec cycles. As a result, the aggregate peak performance of the PS-2100 base module was 150 MIPS (32-bit operands) vs. the 200 MIPS (24-bit operands) of the PS-2000 [Impu89, 7].

Using gate arrays, NIIUVM engineers were able to create their own floating-point unit. While in the PS-2000 floating-point operations were something of an afterthought, in the PS-2100 they were incorporated into the design from the beginning; a double-precision floating-point addition takes only 20 times longer than a fixed-point addition, rather than 143 times longer as in the PS-2000 [Impu89, 7].

Between September, 1990 and January, 1991, LINPACK, Livermore FORTRAN Kernels, and Los Alamos National Laboratory Vector Operations (VECOP) benchmarks were executed on a single PS-2100 base module with 64 processing elements. The effort was not as comprehensive or rigorous as comparable Western efforts and in some cases basic conditions of the benchmarks were unavoidably violated.[2] While the results cannot be considered conclusive and their generalizability is questionable[3], they do provide a

[2] For example, LINPACK policy is that the code be compiled and run without modification–even to the comments– except to incorporate a timing routine local to the system [Dong90]. However, by very nature of the PS-2100, additional routines to initiate and manage the execution of parallel routines on the parallel unit must be added to the source code.

[3] Jack Dongarra, the author of the LINPACK tests states, "Benchmarking, whether with the LINPACK Benchmark or some other program, must not be used indiscriminantly to judge the overall performance of a computer system" [Dong88, 14].

rough orientation of system performance on various types of computational algorithms. A complete account of these tests can be found in [Wolc91].

LINPACK. The LINPACK Benchmark is a collection of FORTRAN subroutines for solving certain systems of linear equations [Dong88]. The benchmark was originally developed by Jack J. Dongarra and others to give users of the LINPACK software package data with which to estimate execution times on their own machine. Since then it has become one of the most widely used (and abused) benchmarks. The basic LINPACK Benchmarks operate on a 100x100 matrix. This was originally felt to represent a ‘‘large enough’’ problem [Dong88]. In recent years the LINPACK has been modified to operate on a 1000x1000 matrix. Called ‘‘Toward Peak Performance (TPP),’’ these benchmarks present a problem which is very large, giving computers the opportunity to use the hardware as efficiently as possible by reducing the relative amount of time spent on overhead. LINPACK policy allows users to make changes to the benchmark code so that this test would represent a user’s ‘‘best effort.’’

Table 5-3 shows the results obtained on the PS-2100 for single and double precision LINPACK tests (unoptimized), and TPP, double precision (optimized):

Test	Performance (Mflops)
Single precision (32-bit), unoptimized	1.73
Single precision (32-bit), optimized	5.3
Double precision (64-bit), unoptimized	.568
TPP double precision (64-bit), optimized	1.65

Table 5-3 PS-2100 LINPACK Results

Table 5-4 gives a comparison of these results with those of Western computers.

Computer	LINPACK (Mflops)	TPP (Mflops)	Theoretical Peak Performance (Mflops)
Cray-1S	27	110	160
Cyber 205 (2 pipe)	17	113	200
IBM RS/6000-320H	12	37	50
PS-2100	.57 (unopt)	1.7	7.6
DEC VAX 6000/410	1.2	1.5	2.6
CDC 6600	.48		
VAX 11/780 FPA	.14		

Table 5-4 PS-2100, Western Machines LINPACK Performance
Source: [Dong92; Wolc91]

As will be discussed below, the relationship of the order of the problem to the number of processing elements plays an important role in determining overall performance. The LINPACK Benchmarks use problems of order 100, 300, 1000, i.e. not even multiples of 64. When the LINPACK algorithms were run on a 1024x1024 matrix (optimized), the performance was 10.877 Mflops.

Livermore FORTRAN Kernels. The Livermore FORTRAN Kernels (LFK) are a set of 24 small algorithms drawn from applications used at Lawrence Livermore National Laboratory. The degree to which the Kernels accurately characterize the LLNL workload has been a point of discussion [Lube88; Mcma86].

The LFK were executed on 32-bit data. Most published figures for Western machines report performance on 64-bit data. Table 5-5 compares PS-2100 performance with that of a Cray-1 running the CFT ’84 compiler. An analysis of PS-2100 performance on these algorithms is given in [Wolc91, 21-27]. Although there are some surprises, such as the low performance on Kernel 21 which involves a matrix multiply–something that should parallelize very nicely–, the variation in performance on the various kernels can be ex-

Kernel	PS-2100 (Mflops)	Cray-1 (Mflops)
1	2.315	100.0
2	0.081	41.7
3	0.643	33.3
4	0.066	24.3
5	0.113	7.7
6	3.638	7.0
7	4.787	120.0
8	10.500	55.4
9	6.671	68.0
10	2.364	36.0
11	0.211	2.9
12	0.829	25.0
13	0.778	4.0
14	0.103	5.6
15	1.305	
16	0.189	
17	0.033	
18	5.786	
19	0.163	
20	0.094	
21	0.040	
22	1.101	
23	0.281	
24	0.037	
Arithmetic mean	1.757	37.92
Harmonic mean		
Loops 1-14	0.263	11.1
Loops 1-24	0.148	
Maximum	10.5	120.0
Minimum	0.033	2.9
Ratio (Max/Min)	318.375	41.1

Table 5-5 PS-2100, Cray-1 Performance on Livermore FORTRAN Kernels
Source:[Worl84; Wolc91]

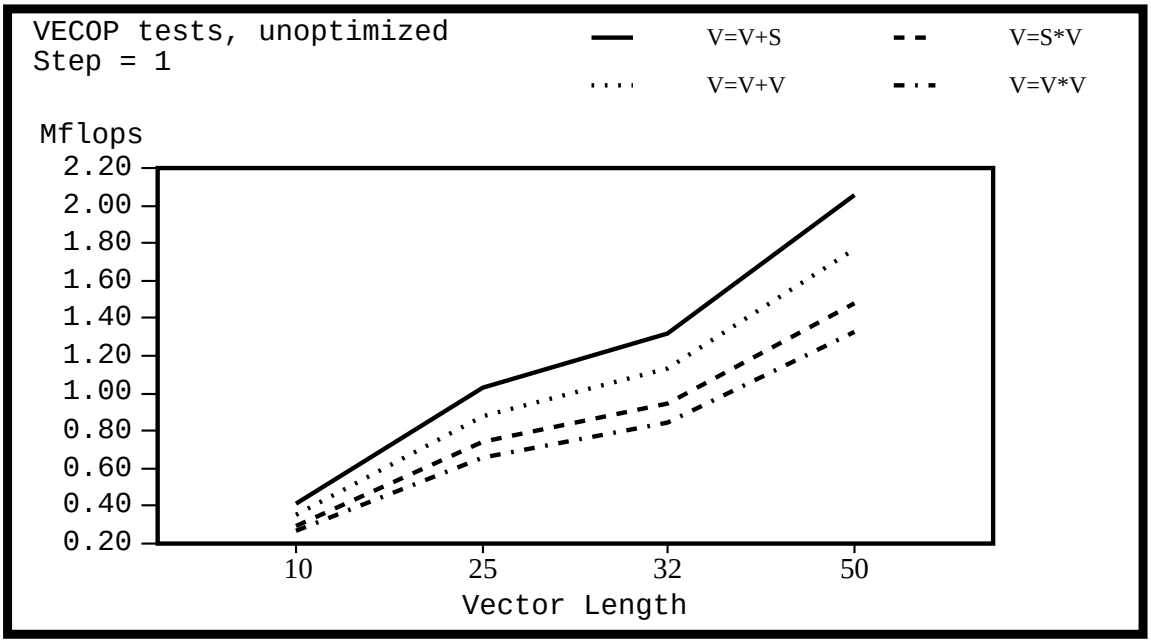
plained roughly by two basic factors: the degree to which iterations of the loop can be executed on multiple processors, and the relationship between the number of PS-2100 processors and the number of loop iterations in the kernel.

VECOP Tests. Researchers at Los Alamos National Laboratory developed a simple model to help understand the performance of supercomputers [Buch84]. The vector operation (VECOP) tests are a simple set of routines to measure the performance on 64-bit floating-point operations as a function of vector length.

The benchmark measures the performance of 1,000,000 executions of such simple vector operations as adding a scalar to a vector (V+S), adding two vectors (V+V), adding a scalar to the product of two vectors (V*V+S), etc. The results are dependent on both the vector length and the manner in which operands are stored in memory. In particular, the benchmarks consider three storage cases: operands and results are stored in contiguous memory locations, in non-contiguous locations but with constant stride, and in a random fashion (scatter/gather). Some VECOP results are shown in Figures 5-3 and 5-4. A more complete set of data can be found in [Wolc91]. Not surprisingly, the VECOP results rather clearly reflect the nature of the PS-2100. Some of the conclusions to be drawn from the tests are [Wolc91, 28-30]:

- Performance is very sensitive to the way data elements are stored in memory and the length of the vector. If the number of processors is a multiple of the stride (such as when stride = {2,4,8}, vector elements will be doubled up on some processing elements but not on others. When stride=2, as shown in figure 5-3, data are stored only at the even (or odd) processing elements. A maximum of 32 processing elements execute, leading to the observed decrease in performance, when the vector length is greater than 32. If the vector is short enough that all elements can be distributed to all active processing elements, then there is no degradation of performance.

- Performance is very sensitive to the relationship between the number of processing elements and the number of elements in the vector. In the graph in Fig-



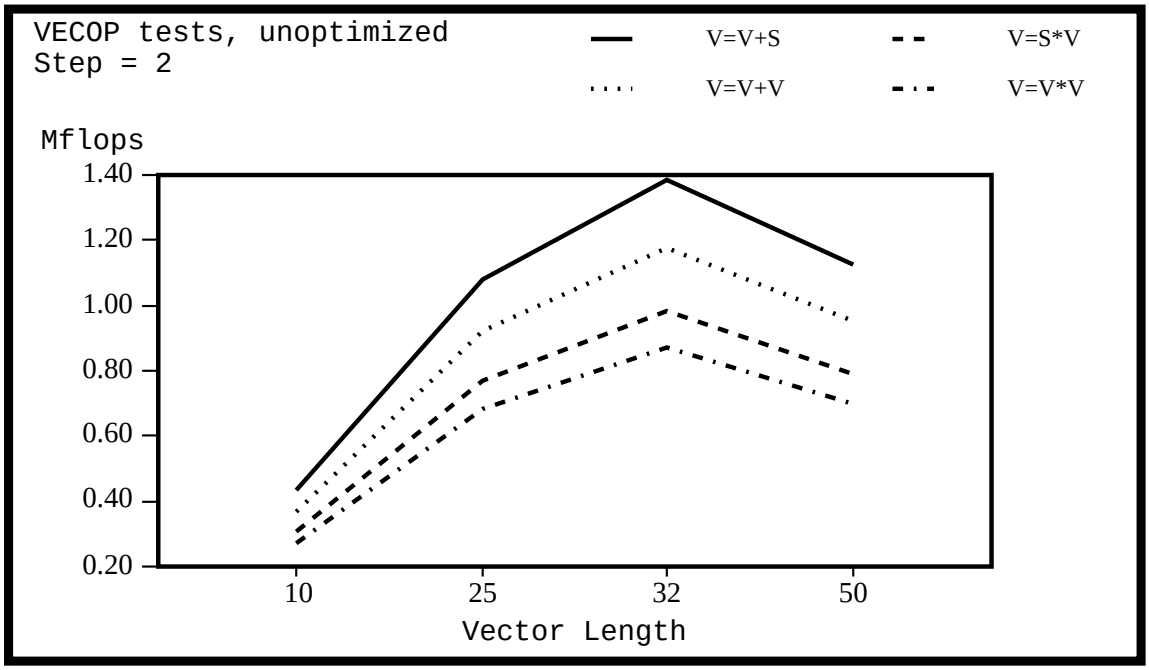


Figure 5-3 VECOP Results, Unoptimized, for Step = 1,2

ure 5-4, when the vector length <= 64, vector elements can be fully distributed among the processing elements. Increasing the vector length allows more

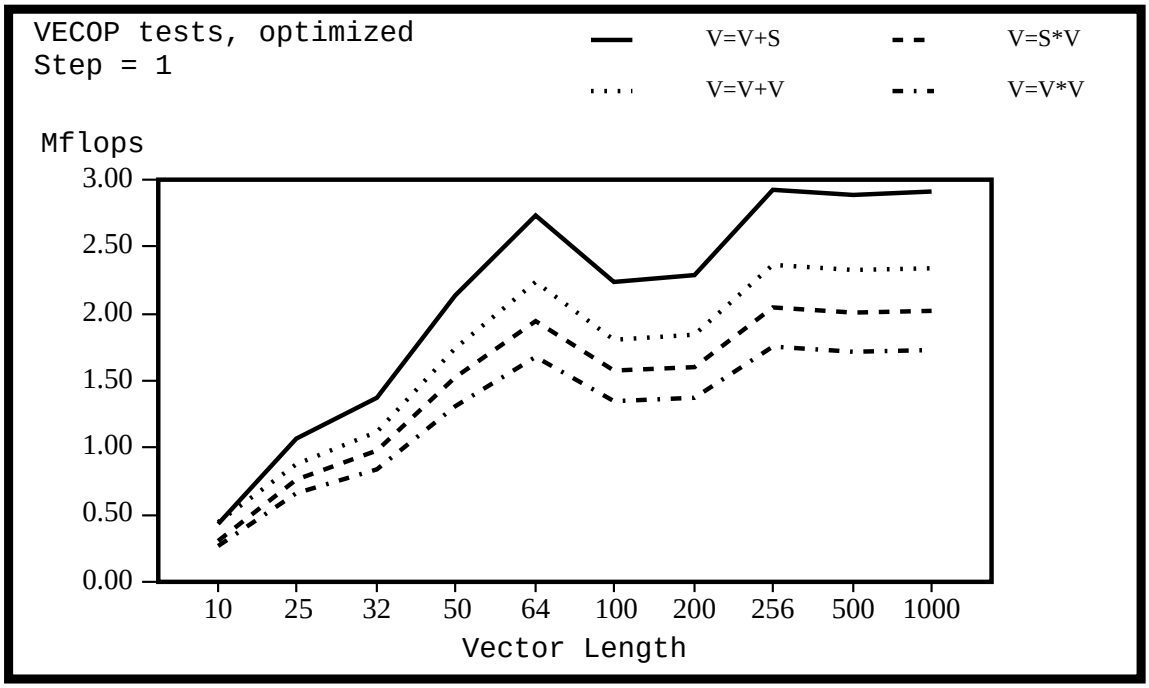


Figure 5-4 VECOP Results, Optimized, for Step = 1

processing elements to execute during a given time period, leading to higher performance rates. The performance peaks when the vector length (n) is a multiple of 64 since all processing elements are busy all the time. When n>64, the multiprocessor must ‘‘strip mine’’, or process data in sets of 64. The performance decreases if during some time period not all processing elements have data to process. This occurs whenever the vector length is not a multiple of 64.

- The PS-2100 has very weak performance on scatter-gather operations. These operations involve indirect addressing of memory locations. A vector contains addresses which are accessed to load (gather) or store (scatter) data values.

Weaknesses of the PS-2100. While a considerable improvement over the PS-2000, the PS-2100 still suffers from a number of weaknesses [Iten90]:

- Scalar processing. A SIMD machine, the PS-2100 naturally exhibits higher performance rates when multiple processing elements operate simultaneously. Not only do scalar operations leave many processing elements idle, but the base performance of each processor is not fast compared with world standards. The LINPACK benchmarks involve significant amounts of scalar computation. The lack of powerful scalar capabilities makes the PS-2100 particularly susceptible to Amdahl's Law.

  Amdahl's Law states that the achievable performance of a computer that has a parallel component and a scalar component is a function of the speed of the scalar unit and the fraction of the computation performed on it, and that the degree of speedup is dominated by the scalar portion. Specifically, Amdahl's Law states that if R is the ratio of the parallel processing rate to scalar processing rate, then the final speedup, S, of a process that is P percent vectorized is

$$\frac{1}{\left(\left[1-P\right]+P/R\right)}$$

  In the PS-2100, scalar computation can be performed in one of two ways. The host computer can execute the source program, with only the parallel routines being off-loaded onto the parallel processor. Thus the host computer can execute the non-parallel portions of the code. On a PS-1001 host, single-precision fixed-point addition requires 980 nsec (1.02 MIPS) [Impu89b, 5]. The execution time on a single PS-2100 processing unit is 420 nsec (2.38 MIPS). Thus the ratio R of parallel execution rates to scalar is (64 x 2.38)/1.02 = 149. If a process is 90% parallelized, the total speed-up would be only 9.4, or 6.3% of

the theoretical maximum speed-up of 149 which would occur if the process were 100% parallelizable.

Alternatively, scalar computation can be performed on a single processing element. In this case, the ratio R=64, and the total speed-up for the example above would be 8.8, or 13.8% of the maximum speed-up of 64.

- Data transmission system. Although the PS-2100 is an improvement over the PS-2000, the system still lacks a truly high-speed communications system for transmitting data between processing elements.
- Memory bandwidth. In each memory cycle, only one 16-bit word can be read from the processing element.
- Floating-point and double-precision computation. The PS-2100 does not incorporate high-performance hardware floating-point units. Floating-point and double-precision computations require a significantly greater number of machine cycles than fixed-point and single-precision operations.
- I/O remains one of the greatest bottlenecks. According to Itenberg, because of this bottleneck, one is able to attain little more than 10% of the theoretical peak performance of the system. While the system was designed for use with 317.5 Mbyte drives, in practice these drives have not been available. PS-2100 configurations are often equipped with imported Winchester drives made for personal computers [Impu91, 9].

Benchmark data must be handled carefully. Although they are often designed to give results which are representative of performance on a particular class of data, the degree to which they succeed in this has been a popular point of discussion and dispute among computer users and designers for years. While results for the PS-2100 may be considered low in comparison with many commercially available computers in the West, there is at

least one example of how such an architecture (in this case a dual-configuration PS-2000) can be used to obtain results which are surprisingly competitive.

In September, 1990, a group from Sandia National Laboratories and Lawrence Livermore National Laboratory visited the Applied Physics Institute in Novosibirsk, which was solving high-speed projectile impact problems on a dual PS-2000 configuration. During the visit, the US group was asked to define a problem to be run on the PS-2000. One of the members of the delegation, Jim Asay, set up a two-dimensional hypervelocity impact problem which involved the impact of an aluminum pellet into a double-walled shield followed by an aluminum target. The impact of a high-speed object on a satellite would be a similar problem. The description was entered into the computer through a reportedly user-friendly interface. The computation was performed on a coarse grid and was solved in about 12 minutes.

After returning to the US, the scientists reproduced the results on a Cray Y-MP using one processor. They tried two different approaches, giving processing times of 4.1 minutes and 14.25 minutes respectively. The latter used a method that was not vectorized.

The level of resolution was not nearly as fine as on the Cray, and the application was undoubtedly highly optimized (coded in either ASPS or, perhaps, in microcode) for this particular problem. Nevertheless, the machine gave useful results for a computationally intensive problem.

5.4 A Period of Change

In the years following Gorbachev's appointment, changes in legislation and policy unleashed forces which quickly snowballed, causing dramatic changes in the economic, social, and political environments within which organizations like NPO Impul's functioned. Change occured in the legislative environment, in the PS-2x00 market, in rela-

tionships with R&D sponsors, suppliers, customers, the ministry, factories, and foreign organizations. The changes placed extraordinary pressures on NPO Impul's, created unprecedented opportunities and degrees of freedom to adapt to these changes. We discuss here those changes which appear to have had the strongest impact on the Problem-oriented Computing Systems division in terms of its structure and activities, and the PS-2x00 computers.

5.4.1 Relationships with Suppliers

The development and manufacture of the PS-2x00 line is clearly dependent on the ability to acquire the necessary inputs. These include everything from subsystems to components to cabling to packing crates. During the first few years after the introduction of the Law on State Enterprises (Associations) in 1987 NPO Impul's had relatively few problems acquiring the necessary component base. Here the nature of the component base, the structure of the market for NPO Impul's machines, the existing relationships with components manufacturers, and the timing of development played helpful roles.

The PS-2000 had been built almost entirely with chips in series production at the time the system was developed. Even acquiring such chips during the late 1970s was not easy, for Minelektronprom was overloaded with its own tasks and problems at the time. Significant assistance had to be given to NPO Impul's by the USSR Council of Ministers to establish the necessary relationship with Minelektronprom. Once the link to Minelektronprom factories had been established, however, subsequent interactions could take place largely without the involvement of the higher levels of government. NPO Elektronika in Voronezh' had not developed gate arrays specifically because of needs at NPO Impul's; semi-custom gate arrays can easily serve as building blocks for a large number of different computers. Given existing production of such chips and the establishment of relations between NPO Impul's and NPO Elektronika, however, low-level work-

ing contacts between Itenberg's engineers and their counterparts in Voronezh' were, for the most part, sufficient to work out the preparation of gate arrays tailored to the PS-2100.

When the PS-2100 was completed, The Elektronika Plant contracted with Impul's to move forward and develop the next generation of chips. It is unlikely that NPO Impul's was the only customer for such chips, although the funding for development came primarily from the financing provided to NIIUVM for development of the PS-2300 (discussed below). Although NPO Elektronika was reportedly operating under 100% state orders (*goszakaz*) in 1989, there was at this time much talk about the level of state orders dropping significantly in the coming year or two. NPO Elektronika therefore felt that it was in its interests to work on the next generation of gate arrays, a chip with wide potential applicability and a proven market.

We speculate that timing factors also played a role in NIIUVM's ability to acquire the chips necessary for the PS-2100 and also the PS-2300. The chips for the PS-2100 had already reached the production stage by the time the new economic mechanisms introduced by the Law on State Enterprises (Associations) began to take effect in Minelektronprom, in 1988-1989. When the contract between NIIUVM and NPO Elektronika to work together on chips for the next generation PS-2x00 was agreed upon, probably in 1988, the prospects for the machines looked good. What would have happened if the PS-2100 prototype had not been developed until 1991 or 1992? A drop-off in demand (described below) for these and other machines using gate arrays would surely have been felt and NPO Elektronika would by this time have been exercising much greater autonomy in determining its product mix. Would it have elected to proceed with development of the new gate arrays? We cannot be certain, but there is a non-trivial probability that they would not have, leaving NIIUVM stranded.

In 1990, the principal shortages were not in components, but in other, less exotic materials. Regarding components, S. N. Krashenko, deputy chief-engineer for new technology at SPZ, commented that, ‘‘[i]t is now possible to obtain everything, but the question is how much it will cost’’ [Kras90, 2]. In shorter supply than components were machine-tool steel, quality wire, good insulation, glass-cloth-base laminate for multilayer boards, and lumber for packaging. ‘‘The component base is right now, I’d say, tolerable’’ said Krashenko. According to Itenberg in September, 1991, the first gate arrays with 17,000 gates were scheduled to be delivered towards the end of that month.

5.4.2 Demand for PS-2x00 Computers

Under the reforms introduced by the Law on State Enterprises (Associations), two changes in particular had a significant impact on factories and, in turn, on the relationship between those factories and their customers and the R&D organizations which developed new technology. First, the reforms greatly increased the ability and need for factories to expand the so-called horizontal links, the links with their customers and suppliers. Second, the system of state planning which dictated what items were to be manufactured, who would manufacture them, where the inputs were to come from and where the outputs were to go, began to recede during the 12th (and final) Five Year Plan, giving way to the system of state orders (*goszakaz)*. While in many respects the state orders served the same sort of coordinating and supply functions as the annual and Five Year Plans (FYP) the total volume of state orders was decreased during the years following their introduction. This was both because of a conscious effort to move away from a plan-based system to a system based more on direct contracts between suppliers and customers, and because of a decrease in military orders stemming in part from efforts to convert military production to civilian.

While state orders were decreasing, direct contracts were not fully replacing orders that earlier had been incorporated into the Plan. As PS-2x00 customers were given greater responsibility for their own financial status and the uncertainty of the economy increased, they became more cautious in their expenditures, reducing the number of orders which they were submitting. While under the former state budget allocations money was allocated specifically for the purchase of computers and could be spent only on computers, under the new economic mechanisms such purchases had to come out of an organization's income. At the same time, the money was no longer ear-marked specifically for the purchase of one type of equipment or another. Computers had to compete with building apartments, providing social services for workers, the purchase of production inputs, etc.

As a result, NPO Impul's experienced a considerable financial crisis during 1988 and 1989 [Shka88, 78]. The market for the high-performance computers was weakening for two additional reasons. First, the PS-2000 was being phased out of production. Organizations planning to buy such a machine would do better to wait for series production of the PS-2100. Second, the accounting practices of Soviet organizations served to reduce demand among organizations that were simply planning to upgrade from a PS-2000 to a PS-2100. The planned amortization period for the PS-2000 was initially set at ten years. An enterprise which, for example, purchased a unit in 1985 would be required to use the system through 1995. It could sell the system, but not simply throw it away. While not, strictly speaking, limiting the purchase of a new machine, it was still legally difficult to replace a PS-2000 with a newer model.

In 1990, many of the traditional customers began placing orders. By September, 1990, orders for about 150 PS-2100 base modules reportedly had been received. Itenberg interpreted this as an indication that customers who, a year or two earlier, had been post-

poning purchases of new technology were recognizing the benefit of the computers to their operations and deciding to make purchases in spite of the economic turmoil.

In spite of an apparent genuine desire to purchase the machines, most of these customers were not able to carry through their orders. The increasingly chaotic economic conditions proved too strong. Even in September, 1990, Itenberg realized the difference between an order on paper and a purchase. He stated that an actual purchase was dependent on three things: a) the customer actually having the money to purchase a machine, b) the customer having the desire to purchase, and c) the factory's being able to acquire the components necessary to construct the machine. During 1988 and 1989, a significant number of customers who had planned to purchase a computer changed their plans, deciding to purchase the computer not immediately, but a year or two later.

Itenberg's fears about the orders were realized towards the end of 1990. The Soviet financial year coincides with the calendar year, and many of those who had placed orders had done so in anticipation of funds being available in the 1991 budget for their purchase. In a large number of cases, this proved not to be the case. Money was not available, contract prices for a many products were rising, and the potential customers once again became circumspect in their purchasing, waiting to see what the future would bring. Impul's' market was being dragged down by the general economic crisis. Production levels throughout the country were falling precipitously and the level of debt between enterprises was growing rapidly. Few organizations had money available to purchase the new computers, or chose to purchase the fashionable Western personal computers.

A final reason for the drop in demand, not easily measured, is the influx of Western technology. According to A. S. Nabatov, given the opportunity to travel abroad to sign a deal to acquire a piece of Western technology, many potential Impul's customers would

purchase the foreign technology rather than buy the domestic product. Of course, in industries where hard currency is not available, this factor plays a smaller role.

By September, 1991, approximately forty PS-2100 base modules had been manufactured but only approximately 20-30 had been sold [Tcha92, 33]. The number of customers willing and able to pay for real machines had shrunk from several tens, to single digits. Particularly serious was the fact that a couple of customers who had ordered fifteen base modules between them had canceled their orders after the machines had entered production.

Although the Law on State Enterprises (Associations) gave factories greater freedom in establishing their own contractual prices, NPO Impul's has kept prices for the PS-2100 at a constant level, in effect reducing their relative price as inflation causes price increases throughout the economy. The cost of a PS-2100 base module plus peripherals was 500,000 rubles in 1989, of which approximately 2-300,000 rubles was for the base module itself [Iten89, 3]. This figure was the same in 1990. In 1991, a base module alone was still being sold for 200,000 rubles. The sale price did not exceed the official ''limit price,'' set under the state price formation formulas while the machine was being designed, primarily because raising the price would hurt an already very weak market.

At the same time as demand for the PS-2100 was dropping off, financing for the next generation, the PS-2300, remained firm, at least for the time being. The Ministry of Geology remained the primary sponsor, supplying roughly five million rubles per year. According to Itenberg, through 1991 funding was sufficient for R&D work and the manufacture of the number of machines needed by the sponsor. Unlike many other divisions at NIIUVM, Itenberg's division was receiving by some accounts 95% of that which was needed to carry on development of the PS-2300. For this reason, Problem-oriented Computing Systems division did not, as of 1990, have to draw on credit to support the work

although NPO Impul's as a whole had had to. Many of the other NIIUVM divisions which had received financing through the Ministry of Electronic Equipment and Instrument-building (Minelektrotekhpribor), the successor to Minpribor, were in catastrophic shape as the ministry failed to obtain the usual funds; some of these were obtaining only 10-15% of what they needed to carry out their research. Thus Itenberg's division was in better shape than most.

The prospects for development were highly uncertain, however. The demand for the current generation and research support for the next are inter-related. If branch organizations decided, for example, to purchase IBM workstations rather than PS-2100s, or otherwise refuse to purchase the PS-2100 computers, the ministry is not likely to continue funding development of such machines.

5.4.3 Relationship with the Factories

NPO Impul's includes two factories, a prototype development facility and the Severodonetsk Instrument-building Plant for series production. NIIUVM has had a close working relationship with both of these units for many years, greatly facilitating the introduction of new machines into production.

During the scientific-research (NIR) and prototype-development (OKR) phases, both the series production and prototype development factories–but particularly the latter–are actively involved in the project. Technicians from the prototype development factory participate in the actual construction of prototypes, but representatives from the series production plant are also involved to make sure that the capabilities of the factory are taken into account. Decisions about the types of cabinets to use, the number of layers in the boards, the types of interconnects, and the precision class of the boards are examples of decisions which depended on the factory. The prototype development team uses the same components, materials, and construction tools as the series production plant. The docu-

mentation for series production is written during the prototype construction phase at the prototype development facility.

When a few prototype units have been completed and the production documentation is finished, the documentation is delivered to SPZ where the process of assimilation into production begins. In the case of the PS-2000 this process took a year. For the PS-2100, it took closer to two years. During the assimilation period the series production facilities are set up, modifications are made to the design to make it better suited to series production or to correct weaknesses discovered during the construction of the prototype.

The relationships between NIIUVM and SPZ have been strong, given that they were physically close together and both subordinate to the NPO Impul's General Director and scientific-council. Production orders for SPZ were worked out not only with NPO Impul's planners but also with the higher level ministerial and state planning organizations. Getting the factory to manufacture new machines was not always a smooth process, but through the NPO Impul's ties, NIIUVM was able to work more closely with the factory and have greater influence over its production schedule than could, for example, research institutes in the Academy of Sciences which had to rely on ministerial factory which were, administratively, far removed.

The R&D efforts at NIIUVM were not funded directly from the sales of the products manufactured by SPZ. One or more primary sponsors–in this case the Ministry of Geology–who had a strong need for a given technology funded the entire research and development process. In turn, these sponsors had the right to dictate major systems requirements and receive the first, prototype, units. The cost of development was determined through a series of state normatives regarding the wages to be paid, contributions to other funds, the costs of materials and other inputs, etc.

When the prototype development phase ended, the production documentation was ‘‘thrown over the wall,’’ gratis, to the series production plant which assimilated production. The cost of the series produced units was calculated according to state norms for the cost of materials, equipment, wages, various factory funds including a technological retooling fund, and profit, etc. The profit is used for various expenses such as construction of housing for workers, construction of new buildings, equipment acquisition, etc. These figures are worked out in the Laboratory of Price Formation which establishes a so-called ‘‘limit price’’ for each product. Before the reforms, a system could not be priced above the limit price.

The cost of production did not include R&D costs, however. The principal sponsors of R&D carried this cost themselves, gaining the right to obtain the first units produced. Subsequent customers pay only for the hardware and software; no further payment for the R&D costs of that machine were made. As a result, primary sponsors paid considerably more than subsequent customers.

The significance is that the R&D facility does not profit monetarily from the sale of systems in series production. Although profits from the sale of computers were absorbed into the NPO Impul’s and Impul’s in turn provided money for the budgets for its constituent parts, NIIUVM would receive not an additional kopek if 100 systems were manufactured rather than 50. While NPO Impul’s has a certain fund available to sponsor in-house R&D, it is not enough to support the development of systems like the PS-2x00 machines. Funding for the research and development of the next generation system would again have to take the form of specific, ear-marked funding from a sponsor rather than from proceeds of the sale of previous generation systems. In this respect Impul’s differs considerably from Western corporations which rely heavily on income from sales to fund their R&D programs. Dependent on R&D funding from sponsors or the Ministry,

NIIUVM has little access to hard currency, further limiting its ability to acquire foreign technology.

In 1990 serious discussions were held about the possibilities of altering the financial relationship between NIIUVM and SZP such that documentation developed by NIIUVM was sold to SPZ. No concrete changes were made as a result of these discussions. A stated reason was that the accountants at Impul's had not learned to properly evaluate the value of intellectual labor such as prototype development. A more likely reason is that the factory's commitment to many of the new NIIUVM systems was marginal enough that requiring the factory to purchase production documentation could have jeopardized entire classes of machines, such as the PS-2x00.

As the constituent parts of NPO Impul's gained greater autonomy and responsibility for their own affairs, the SPZ naturally was increasingly concerned that there be a market for whatever it was manufacturing and that the manufacture be profitable. No longer was the production schedule determined exclusively by the Plan. Except for orders which were part of state orders, SPZ had to find customers itself. In general, the greatest profits were gained from machines which had been in production for more than three years. The PS-2000 and SM-2M were both in this category. In 1990 it was too early to determine the profitability of the PS-1001 and the PS-2100. Production of the PS-1001 peaked in 1991 at 100 units, declining to 70 in 1992.

The PS-2100 was not the main SPZ product, accounting for only about 2-3% of SPZ's annual gross income. Nevertheless, such orders were not to be dismissed lightly, especially in times when the overall number of orders for the factory for all products was decreasing. But the future of the PS-2100 was not assured and there was a significant amount of pressure on developers to prove its worth to SPZ, to prove that there were customers for the system. When the orders for 15 base modules were cancelled, Itenberg felt

obligated to purchase these units from the factory to prevent it from shutting down production completely. The total cost of these base modules, at 200,000 rubles each, was three million rubles, a heavy debt for Impul's and, in particular, for the Problem-oriented Computing Systems division.

As profit grew as a motivating factor for SPZ, it was less inclined to engage in marginally profitable activities. One example of this was the factory's efforts to provide systems maintenance and support for installations in the field. Providing these services required a highly-trained, highly qualified staff. Itenberg's team and engineers at the prototype development factory wrote the production documentation, so SPZ engineers did not develop the expertise necessary to tune and service the machines. The factory did not prove willing to train individuals, since this would require a significant investment of time and resources. Several of the SPZ workers who knew the system best reportedly left the factory to start their own small enterprise.

In short, the relationship between NIIUVM and SPZ through 1991 was stable, but uncertainty about its future was growing. SPZ appeared willing to continue production of NIIUVM machines already in production, but lacked the internal expertise to develop new machines itself.

In 1992, because of a lack of orders for the PS-2100, SPZ ceased production and reoriented much of its capacity to consumer goods for which there was a market, such as telephones, watches and washing machines. While in the past computers constituted 80% of SPZ's production, but the end of 1992 the percentage had dropped to 20%. SPZ became completely independent of NPO Impul's and had no administration in common with NIIUVM.

5.4.4 Relationship With The Ministry

Prior to the perestroika reforms, the relationship between NPO Impul's and its ministry, Minpribor, was similar to that elsewhere in the economy. The ministry owned all Impul's facilities, established the production and development Plans and defined the indicators and forms associated with the planning process, provided funding for operations, and absorbed income generated through the sale of Impul's products. Changes to Impul's structure or activities had to be approved at higher levels in the ministry.

Following a poor performance during the 11th Five Year Plan, Minpribor in 1986 underwent a reorganization [Pano85; Prav860709; Shka86]. In an effort to tighten the link between R&D and production facilities, nearly all research facilities not already part of a production (PO) or scientific-production association (NPO) were incorporated into one [Shka86; Shka86b]. Since NIIUVM was already a part of NPO Impul's, these changes had minimal impact on it.

One consequence of the *perestroika* reforms was that Minpribor began to play a smaller and smaller role in the life of Impul's, particularly in the day-to-day activities. On January 1, 1987, Minpribor became one of five ministries to implement full *khozraschet* in its enterprises and associations, anticipating the passage of the Law on State Enterprises (Associations) which went into full effect January 1, 1988 [Sukh87, 9; Rumy88]. The Ministry had made a transition to *khozraschet* on the sectorial level in 1970, but this was the first time this accounting method had been applied in its full form to individual associations and enterprises [Shka87].

In July 1989, Minpribor was dissolved in a major governmental reshuffling, at least in part to reduce administrative overhead [Gorb89, 1]. Minpribor organizations were predominantly absorbed into the Ministry of the Electronic Equipment Industry (Minelektrotekhprom) to form the Ministry of Electronic Equipment and Instrument-building Indus-

try (Minelektrotekhpribor), although some institutes and factories were absorbed into the Ministry of the Radio Industry (Minradioprom). Under these circumstances, the new ministry leadership played an even smaller role than that of Minpribor and was, according to Itenberg, ‘‘barely noticed’’ by those working in the high-performance systems division. The high-performance development was financed chiefly by the Ministry of Geology, but other NIIUVM research had been financed by Minpribor. Following the absorption of Minpribor into Minelektrotekhpribor the level of financing dropped however, and with it, the involvement of the ministry in the activities of NIIUVM.

5.5 The Response to Change

The changes described above created both significant crises and new degrees of freedom to shape the mission, structure, and technology at NPO Impul’s.

As the economic crisis deepened and demand for NPO Impul’s products dropped, survival became a basic goal. But the goal was more than just survival at any cost; survival meant preserving the capability of staying in the same principal niche–control and high-performance computing systems–that NPO Impul’s had occupied for decades. During the plenary session of the 35th Anniversary Jubilee Conference at NIIUVM, general-director V. G. Rakitin spelled out his goals in an address to Impul’s employees and users of Impul’s computers [Impu91, 1]:

> We are joined by one work. You are and will remain our users and technicians. We’ll retain our orientation, because our work is needed in our economy. Without electricity, chemistry, automation, the economy can’t function. Our task in these difficult times is to preserve the collective, preserve the direction. We will work together to raise the quality of work to a more competitive level.

In this section we examine efforts to preserve the collective, to preserve the direction of development with the Problem-oriented Computing Systems Division by focusing on

two areas of change: modifications in organizational structure, and changes in the high-performance computing systems.

5.5.1 Changes in Structure

5.5.1.1 Traditional NIIUVM Structure

Traditionally, scientific-production associations drew together under one administrative roof organizations which covered the entire product life cycle: R&D facilities such as a scientific-research institute (NII), design bureaus, prototype development facilities, and manufacturing plants. Additionally, as in the case of Impul's, the association could incorporate training centers, associated laboratories, and other organizations. Production associations in general were created for a number of reasons, including administrative economy, achievement of economies of scale, promotion of the status and power of managers of very large enterprises, and the consolidation of a given sector. Scientific production associations in particular were designed to forge a closer link between research and production facilities [Nove86, 69-70].

Throughout the Soviet Union the structure of industrial research institutes has been cast from the same mold: an NII consists of multiple divisions, each consisting of multiple laboratories. Each division is responsible for its own line of work. At NIIUVM, I. I. Itenberg is the head of the division of problem-oriented high-performance computer technology which consists of approximately 75 people, including systems engineers, software engineers, lay-out engineers, operators, and others.

The divisions are subordinate to a scientific-technical council, which consists of the division and sub-division heads and deputy-directors of the institute and is chaired by the general-director. The general director and the scientific-technical council form the com-

mon administrative point between the NII and the other component organizations of the NPO.

Deputy-directors, such as V. V. Rezanov, were responsible for coordinating the activities of the various divisions and for allocating resources to each. Rezanov was responsible for ensuring that the various divisions all adhered to the same interface standards so that their devices would be interoperable, in keeping with the ASVT philosophy. If Itenberg and his engineers wished to depart from such a standard, they would have to get permission from Rezanov and the scientific-technical council. He also distributed financing to each. Requests were, of course, submitted by the divisions themselves and refined by the institute's economists who saw that all figures adhered to the myriad of economic norms established by the government and ministerial planners. But the link between the request and what was allocated passed through Rezanov, who was a critical link in the financial chain between the sponsor and the R&D departments. Rezanov and the general-director were also key players whenever new ties were being established with a supplier organization, such as with Minelektronprom factories, but once the ties were established, most contact took place between technicians in the respective organizations.

Itenberg's division was subordinate to Rezanov, but from the outset it had greater independence than did many of the other divisions at NIIUVM. A principal reason was that the PS-2x00 computers represented a qualitatively new line of development from the traditional, control system research. As such, the research was more self-contained than in other divisions.

Prior to the *perestroika* reforms, this structure was virtually invariant. It was created according to the so-called state schedule (*struktura shtatnogo raspisaniya*). The creation of new divisions and laboratories had to be approved at ministerial levels, sometimes even at the level of the USSR Council of Ministers. Similarly, the appointment of indi-

viduals as directors, deputy-directors, heads of divisions and laboratories had to be approved by the ministry and, especially in the case of director and deputy-directors, by the Council of Ministers. The job titles and their associated pay levels were all fixed by the government.

The laboratory structure was virtually invariant. Once created, laboratories were seldom broken up. Once assigned to a laboratory, a researcher moved out of a laboratory only through promotion, dismissal, or voluntary departure; rarely did someone transfer laterally to another laboratory.

The head of each division was also the chief engineer for the work carried out. He had to adhere to technical policy set by the scientific-technical council and the general-director and deputy-directors.

The chief engineer had full responsibility for work carried out in his division, and was the final arbiter for technical and procedural questions which did not need approval at higher levels.

5.5.1.2 Changes to the Structure of the Problem-oriented Computing Systems Division

The reforms placed nearly complete authority over the structure of the association and the institute into the hands of the institute's directorate. No longer were approval or resolutions from the ministry or the Council of Ministers needed to create new laboratories or divisions, or alter the structure of the existing ones. With this freedom, this degree of authority, came the need to address the questions of what would be the best form of organization for NPO Impul's, NIIUVM, and the constituent divisions. While the basic laws allowing changes to the structure were passed in 1987 as part of the Law on State Enterprises (Associations), the existing system had considerable inertia, and it wasn't until 1990 that new organizational forms began to appear at NIIUVM.

The transition towards a market economy placed increasing financial pressure on NPO Impul's. All recognized that work would have to be carried out more effectively, and many viewed structural change as key to accomplishing this. Specifically, many felt that greater effectiveness could be achieved by increasing the independence of individual divisions and expanding the authority of the division heads over the scientific-technical and financial activities. This would, it was argued, increase the effectiveness of decision-making, increase personal responsibility for these activities, and reduce overhead.

Gradually, authority was decentralized. The first major change to NIIUVM following the reforms was the creation of so-called temporary scientific-technical collectives (VNTK). In contrast to laboratories which were long-term structural entities which carried out a variety of activities for many, many years at a time, the VNTK were task or program oriented and much more flexible. A group drawn from different laboratories, different divisions, or even different institutes would be created to address a given project, and that project alone. When the project was completed, the collective could disband and its members join new and different groups being formed to work on other projects. The VNTK organizational form dates back to an August, 1983 Council of Ministers resolution, but had been used only infrequently, for relatively large-scale programs such as the START new generation project discuessed in chapter 6 [Fort90, 112]; Itenberg sought to apply this form on a much more modest scale, to individual tasks.

Itenberg's division was the first at NPO Impul's to experiment with the VNTK, creating the initial one in November, 1990. The idea arose when tasks came up for which Itenberg's division did not have the necessary personnel and expertise. At the same time, some of the other divisions had the needed expertise and less than a full workload. A key enabling factor was the fact that responsibility for budgets was being pushed down into the divisions themselves as the *khozraschet* mechanisms took hold more fully. While in

the past laboratories and divisions had been very protective of their workers, they were now more willing to let them work on someone else's project and get paid through that project. As Itenberg stated, ''This allowed us to obtain a certain degree of independence. Now we ourselves set the rate of pay, we ourselves determine the incentives, we ourselves hire people, we ourselves can attract people from other quarters. It's more flexible.''

By September, 1991, Itenberg presided over nearly half a dozen VNTK. The principal ones were for the development of personal geophysics computing systems (PGVK), the development of software for the PGVK, and the installation and tuning of a PS-2100 system with ten base-modules. Additional VNTK were devoted to integrating a single base module with a personal computer and to developing systems software. A VNTK for installing and tuning a PS-2100 with six base modules had already completed its work and had been disbanded.

The organizational structure and the nature of some of the tasks carried out by Itenberg's workers changed in tandem. Tasks which could increase the income of the division were desirable. One change, for example, involved the services provided to users. Traditionally, once the prototype machines had been built and installed at the principal sponsors, the Itenberg's division turned over all responsibility for installation and maintenance to the manufacturer, SPZ. During 1990, two basic changes occurred which caused the Problem-oriented Computing Systems Division to become more heavily involved. First, several of the individuals at SPZ who had been primarily responsible for this work left the factory to start their own small enterprise. As a result, the factory was less capable of installing and maintaining the systems it was manufacturing. Second, Itenberg recognized that such work could be done by them on a contractual basis, bringing more income di-

rectly into his own division. Members of the division began working with the small enterprise on installation, debugging, and system verification projects.

After the directorate had given Itenberg permission to form a VNTK, others followed suite. By September, 1991, VNTK had spread throughout NIIUVM, to the extent that the division-laboratory structure had been replaced in practice by the VNTK as the dominant organizational form. The former divisions and laboratories still existed administratively, but the work was being carried out in VNTK. To be sure, some divisions like that working on the PS-3x00 made the transition by transforming an entire division into a single VNTK. But the principles of flexibility and fluidity increasingly dominated thinking about organizational structure and even these were considered temporary in nature. The only two divisions which continued to operate under the old structure were the scientific-technical information and standards divisions where the need for flexible working groups was less pronounced.

The issue of which organizational form would be the most appropriate was still under discussion in September, 1991. Since the creation of the VNTK, in August, 1990, a law was passed allowing the creation of a new organizational form, the small enterprise. In the months following its passage, a growing body of discussion and experience had been disseminated through the mass media. Soviet economists also had been promoting such organizations. In additional, several small enterprises were created in Severodonetsk by people who formerly worked at Impul's, with good results. Inspired by these examples, individuals in Itenberg's division–and throughout NIIUVM–grew more convinced that the old system had to be replaced by a more progressive one in which wages were more closely tied to the amount of work and the profitability of the organization. The risk that such efforts could fail was acknowledged, but many felt that the old system with its de-

pendence on centralized authorities would not serve them well in the future. The possibility of creating small enterprises at NIIUVM was first raised in the beginning of 1991.

An arrangement in which independent collectives existed under the roof of the old organization had benefits for both. The independent collectives could use the parent organization's name recognition to acquire contracts more easily, would have greater access to production facilities than if they functioned in complete isolation, had a chance of getting internal or government credits, and could use the same material base as before. The parent firm would profit because the collectives, fighting for survival, would bring in profitable orders, they would pay rent to the parent organization, and the directorate of the parent organization would not have to concern itself with management or personnel issues within the collectives [Pivo92].

Many questions remained to be solved, however. The precise nature of the small enterprise had to be established. At least two types of small enterprise could be used. One variety could employ 50 people or fewer. Another, called a scientific small enterprise, could employ up to 100 people, or up to 250 if it were involved in production [Ezh9008]. Second, the precise nature of the relationship between the small enterprise and Impul's was still unclear. For example, if the small enterprise were a separate organization with no possessions of its own, renting facilities and equipment from Impul's, who would provide such social services as a kindergarten for employees' children? If an employee of Impul's had been waiting ten years for an apartment, would his or her status be jeopardized by joining the small enterprise? Many such questions had to be addressed.

A more fundamental issue had not been resolved, however. While decentralizing the structure of NIIUVM could lead to more effective work, it also threatened to destroy the integration between divisions necessary to carry out the core activities of the institute: the development of complete process control and high-performance data processing systems.

Allowing each division to drift along its own path threatened to undercut the core capability of the institute. For this reason, NIIUVM leadership was resistant to decentralization, but felt the financial pressures to do so accutely.

During 1991 and 1992 recommendations were made to the Directorate to implement a large measure of decentralization in the form of rental collectives (*arendnyy kolektiv*) without granting complete independence to individual divisions [Cher92]. Still a part of NIIUVM, such collectives enjoy increased financial independence, and have a formal agreement with the institute to rent facilities and equipment. They have their own bank accounts so that income received from customers would pass directly to the collective rather than through the institute accounts. They have control over wages, but must earn the money to pay them. They also have the right to fire workers. In non-financial respects, however, the workers continue to operate as employees of NIIUVM. Heads of the collectives are still subordinate to the director, and decisions of the scientific council were binding on them. In August, 1992, an order was issued to implement a number of rental collectives [Cher91]. By the end of 1992 there were five: high-performance computing, general-purpose systems on the basis of the PS-1001, controllers, publishing, and CAD systems. These constituted the core of the institute. Small enterprises did exist, but for peripheral tasks such as engaging in commercial ventures to supply workers with foodstuffs. The institute had a dual structure.

However, there was sufficient discontent with this order that a commission was created to resolve the issue of institute structure. The commission could not resolve differences of opinion among its members and instead of one compromise recommendation, several were offered. These ranged from a centralized structure in which the director had unified control over property, finances, and management to a loosely-coupled collection of units, each with full legal status and control over financial and management issues.

Separately, Rezanov argued passionately that the only way NPO Impul's could survive was if the association achieved a level of self-financing in which the sales of products to customers would generate a profit sufficient to cover the costs of R&D and the operations of the firm. The products with the greatest earning potential were the integrated process control and data processing systems and associated services. The requirements of these systems should determine the structure of the association with production, not R&D, as the core activity. Given the decentralization that had already taken place, he proposed a hybrid structure in which each of the rental collectives worked both on tasks related to the integrated systems under the coordination of the association's leadership, and cultivated individual market niches in which they had complete control [Reza92].

The issue still not been resolved. At the time data collection for this study ended in December, 1992, the issue had yet to be put before the expanded NIIUVM scientific council for a final decision [Cher92].

The issues of decentralization and flexibility versus integration and maintenance of capability were pressing issues at all levels withing NPO Impul's. As greater autonomy and flexibility were given to the divisions and collectives within NIIUVM, Itenberg's primary goals remained close to those stated by the general-director: to preserve the collective, to keep the PS-2x00 development team intact so that the line of development itself could be preserved. He used a three-prong strategy: incorporate flexible teams to address the division's tasks, retain leadership over the work of the division as a whole, and pursue opportunities for earning money outside the division's traditional sphere of activities.

Facing weakening demand for his machines, low profit margins on those few units which could be sold and growing uncertainties about future research funding, Itenberg had to try to find ways to keep his development team intact. Key to keeping his principal

engineers was finding enough money to pay them a wage which could support them and their families.

A possible solution was to try to put non-core individuals to work on projects which would earn sufficient money for the division to support the engineers working on high-performance systems. In particular, if they could start up the manufacture of low-cost, low-overhead consumer products which enjoyed a large market, the twin goals of providing work for all and supporting the key lines of work could be achieved. The list of possible products is broad. They would be willing to manufacture anything from medical equipment to electronic games to coffee grinders to door-handles for cars. Any product would be considered, provided it would enjoy an extensive market and could be manufactured at NIIUVM in conjunction with the prototype development factory.

To cultivate the market for the division's products and explore possibilities for new products Itenberg therefore chose to do more of his own marketing. The chief changes in ''marketing'' involved advertisement and pricing. Promotional literature (prospectuses) had always been printed, but they were now being printed earlier in the product development cycle. Itenberg gained much greater flexibility in setting prices. He sought to preserve and expand the customer base by 1) maintaining close contact with current and past customers, providing new equipment and maintenance, 2) identifying new customers through communications with old customers, 3) participating in conferences where he could both publicize the machines and learn of new applications to run on them. With new applications come new potential customers.

The plan favored by Itenberg and others in 1991 was to create a small enterprise within the framework of NPO Impul's specifically for the production of consumer goods. Ideally, this enterprise would establish a joint operation with a foreign firm which would provide investment funds, or simply technical licenses or know-how to expedite the pro-

duction of specific goods. Because the prefered organizational form at NIIUVM during 1992 was the rental collective and many issues surrounding the role of small enterprises had not been adequately resolved, Itenberg's division was transformed into the former.

The VNTK structures within the division changed little during this year, although their tasks changed as funding for PS-2300 development ended. The team doing hardware development began working on a new generation of control computers, based on Western microprocessors. The software development team worked on applications for existing PS-2100 users.

5.5.2 Changes in Technology

In spite of the many changes precipitated by the *perestroika* reforms, the core trajectory of the high-performance machines has remained very constant. Besides basic improvements in the traditional performance, reliability, and functionality, changes to the PS-2100 and its successor have been relatively small-scale, focusing on making the machines more suitable to a broader circle of users, while remaining within the framework dictated by the nature of their relationship with the principal sponsor and their traditional customer base.

5.5.2.1 The PS-2100

Modifications were made to the PS-2100 configurations to improve the user interface and adapt the machine for use by a wider circle of potential customers. Using a PC AT bus–IUS (''universal systems interface'') adapter, a personal computer could be used as the primary user console. One such configuration, called a personal geophysics computing system, consisted of an IBM PC-compatible personal computer used as a host machine attached via the bus adapter to a PS-2100 base module, a PS-1001 monitor subsystem, and a plotter [Iten91b, 10]. The same adapter could be used in larger configurations,

attaching directly to multiple PS-2100 base modules, the external exchange switch, magnetic disks and tapes, and a number of additional personal computer user consoles [Iten91b, 11]. Other developments under way at the time of this study include the incorporation of standard Token Ring and Ethernet networking protocols.

The PS-2100 was being adapted to other host computers as well. The geophysicists at NIIGeofizika had talked to representatives from IBM who wanted to expand IBM's operations in the Russian oil industries. During the course of such discussions, the question arose about whether or not it was possible to use the RS/6000 workstation as a host to the PS-2100 multiprocessor. It was felt that this would be a way of both selling IBM workstations and the PS-2100 in Russia [Tcha92, 33]. Initial discussions took place during the summer of 1991 and in September A. S. Nabatov visited IBM's Norwegian subsidiary to explore the issue further. These discussions ultimately proved unfruitful.

To increase the range of machines available, a small, desk-side version with 32 processing elements was developed.

5.5.2.2 The PS-2300

Shortly after the PS-2100 prototypes were completed, work began on successor models. Designers explored the possibility of overcoming some technological limitations by building a machine consisting solely of imported components. When it became clear that such a project would not be feasible because of the difficulty of getting Western components and information about them, work began on an indigenous successor to the PS-2100. The aborted project was called the PS-2200 even though it never passed the design phase; efforts were redirected towards the creation of the PS-2300.

As had been the case with the PS-2100, the PS-2300 project began with an analysis of the weaknesses of the PS-2100, the requirements of users–principally of the geophysicists

	PS-2300/2	PS-2300	PS-2300/1	PS-2300 full configuration
Processing elements	16	64	256	2560
Theoretical Peak Performance (Mflops)	60	250	1000	10,000
Main Memory (Mbytes)	2	40	96	1000
Housing	desk top	desk side	cabinet	10 cabinets

Table 5-6 PS-2300 Configurations

who remained the primary sponsors–and the components and other technologies projected to be in production by the time the PS-2300 was to be completed.

Since the principal sponsors and applications remained the same as for the PS-2100, the basic requirements for the PS-2300 were simply to improve the basic operating parameters: increase the processing rate, improve reliability, increase the amount of internal and external memory, provide faster I/O, reduce the physical space, etc. Although not required by the principal sponsors, several measures were being taken to increase the potential market. These included development of a family of PS-2300 configurations, shown in table 5-6,ranging from a desk-top model to a full-scale multicabinet model, the incorporation of non-indigenous computers such as MS-DOS based personal computers and Unix based workstations as the host computers, the incorporation of standard network protocols such as the Token Ring and Ethernet and the use of the IEEE 754 floating-point format for data compatibility with other computers. According to Itenberg, ‘‘There was no such requirement [for using the IEEE 754 standard]. We simply are aware of the state of affairs in the development of computer technology and don’t want to be

left without in our country, although the implementation of this is a bit more complicated..." [Iten91, 8].

The practice of providing compatibility at the level of ASPS (but not at the binary level) was continued. As had been the case with the PS-2100, this would involve recoding the software libraries and some of the systems software, but relatively little additional effort on the part of user applications developers.

A key development making such configurations possible was the development of an improved gate array component base. Gate arrays had been a boon to the PS-2100 project for they enabled designers to customize chips on a base of series production chips with far less effort–both technical and political–than would have been required to develop fully-custom chips. The CMOS gate arrays projected to be available in series production around 1993, the anticipated year of completion for the PS-2300 prototype, have 17,000 gates, over 40 times more than those used in the PS-2100 [Impu91, 9; Impu91b]. With such chips, an entire processing element could be put on a single chip, and four chips could be put on one board. Development of gate arrays funded through NIIUVM took place at the NPO Elektronika in Voronezh', Russia. The first 17,000 gate chips were to be delivered during the fall, 1991.

In designing chips based on these gate arrays, NIIUVM developers were able to integrate 64-bit operations more fully into the processing elements, such that 32- and 64-bit floating-point operations both execute in the same number of cycles [Impu91b, 1]:

	PS-2100 PE	PS-2300 PE
Number of PEs per board	1	4
Peak processing rate (Mflops) on reg.-reg.		
32-bit floating-point addition	1.0	4.0
32-bit floating-point multiply	0.5	4.0
64-bit floating-point addition	0.12	4.0
64-bit floating-point multiply	0.07	0.5

In addition, the PS-2300 will incorporate 128K-512K bytes (depending on whether 256 Kbit or 1 Mbit memory chips are used) of buffer memory per chip to facilitate I/O operations.

As with the PS-2100, designers reconsidered the question of modifying the processing element interconnect schema. Once again, as the design progressed, the decision was made to continue with the original linear-ring interconnect. The new gate arrays provided 128 data outputs per chip, too few to incorporate four-way interconnects easily.

The shift to incorporating the MS-DOS and Unix operating systems is the result of a strategy to use systems more common than the PS-1001 as the host computer. According to V. A. Largin, it is not clear what the future of the PS-1001 will be. Personal computers are widely available and the ability to attach a PS-2x00 parallel processor to IBM PC compatible machines would increase the potential user base. The MS-DOS machines are only single-user, single-task computers, however. Some sort of multi-user, multi-task operating system should be used. The natural choice is a widely-used operating system which runs on commonly available workstations. Unix is the natural choice.

The PS-2300 is an incremental evolution of the PS-2100. The basic requirements and target applications remained intact, and, at least through 1991, funding through NIIGeofizika, the principal sponsor, remained stable at levels consistent with prior years. The basic architectural decisions remained unchanged, while the operating and performance characteristics were improved, chiefly through the incorporation of an improved component base and construction technologies.

As has been noted, several measures were being taken to improve the ability to adapt a configuration to the needs and technology of the widest selection of users possible. The need to improve the marketability of the system was becoming more urgent at the end of

the 1980s and early 1990s, but the basic philosophy was a continuation of that developed in the ASVT and earlier PS programs.

At the end of 1991, work on the PS-2300 came to a hault. The geophysicists who had been funding research did not have the money to continue support. Consequently, NIIUVM did not have the funding necessary to support the development of the new generation of gate arrays at NPO Elektronika. At the time of this writing, efforts were being made to secure some support for the PS-2300 through the Ukrainian government, but it was not clear whether these would be successful. Even if they were, there was a strong possibility that the engineers at NPO Elektronika would have found other work and would no longer be interested in working on PS-2300 chips.

A complicating factor was that Voronezh' is in Russia, now a different state. Thanks to difficult relations between Ukraine and Russia, movement of capital between the two countries is currently difficult and re-established working relations with NPO Elektronika could be problematic. There are no firms in Ukraine which can manufacture chips with the necessary levels of integration, and the PS-2300 depends on being able to place an entire processor on a single chip. Consequently, Itenberg's engineers were considering using general-purpose Western chips instead, should the PS-2300 project be restarted. Because of the specifics of the PS-2x00 processing elements a Western microprocessor would have to be augmented with a set of customized chips, but the latter could have a lower level of integration and, in principle, be manufactured domestically.

5.6 Discussion

The Problem-oriented Computer Systems Divisions within the Scientific Research Institute of Control Computers (NIIUVM) has developed high-performance computing systems with original designs for over a decade and a half. Although not the most powerful Soviet machines, the PS-2x00 computers throughout the 1980s constituted perhaps the

most successful line of Soviet high-performance computers, enjoying high production rates, relatively short R&D cycles, considerable popularity among users, and generally good operating characteristics. The PS-2x00 are an exception within Soviet HPC, having R&D cycles considerably shorter than a decade. The PS-2000, prototyped in four years and assimilated into series production a year later, is unique among Soviet high-performance computing efforts in the speed with which it was brought to fruition. A comparison between this machine and others is instructive in identifying factors which impact the nature and speed of development within the Soviet context.

The third generation of PS-2x00 systems was under development through the end of 1992. Over the years, this family has grown steadily in performance and functionality, but has maintained a high degree of continuity and focus in underlying design and construction from one generation to the next, even during the turbulent *perestroika* era. In this section we summarize the many factors which have shaped development of these computers.

During the reform period, the structures of the Problem-oriented Computer Systems Division, NIIUVM, and NPO Impul's as a whole have undergone considerable change. Organizational structures became significantly more fluid, changing frequently, and the ability to make decisions about the structure, operations, and finances to a considerable degree was decentralized, placing much more authority and responsibility in the hands of the division heads. However, the issue of organizational structure has been a point of considerable debate and had, at the time of this writing, not been settle conclusively. In this section we summarize the factors affecting technological advance and organizational structure within the Problem-oriented Computing Systems division and, to the extent that it affects HPC, within NIIUVM as a whole from the perspective of the conceptual frame-

work introduced in chapter 2. We will analyze the impact of change since the start of the *perestroika* reforms and discuss options for the future of HPC at NIIUVM.

5.6.1 The Technology

Table 5-7 gives an outline of some of the many factors shaping the evolution of the PS-2x00 series. The PS-2x00 family evolved within the context of a belief system–a set of guiding principles–which was in part shared by all R&D divisions of NIIUVM and in part unique to the Problem-oriented Computing Systems Division. A foundational belief was that the machines should be very strongly oriented towards industry, towards actual use in existing applications, rather than as a test-bed for theoretical ideas. The implication was that the machines should be developed as a system, with an integrated, complete set of hardware, software and peripherals, and should be developed and manufactured in a timely manner; simply developing an interesting and complex computational engine with good theoretical parameters would not be acceptable. A second belief characterizing NIIUVM development as a whole was that the systems should serve as broad a customer base as possible. In Western capitalist economic systems such beliefs are driven by basic profit motives: the more you sell the more you earn. Under the Soviet system, an R&D facility like NIIUVM would not earn more from increased sales of a machine, but would increase its stature and political and financial support base. Since users of NIIUVM machines had a wide variety of special-purpose control and computational applications, the strategy which evolved was to build modular systems which could be configured in a wide variety of ways, according to the specific needs of a given user. This strategy eventually became part of the philosophy of machine design and modularity, with its requirements for uniform internal and external hardware and software interfaces, is a strong characteristic of nearly all NIIUVM computers. As the centralized directive

<table>
<tr>
<td>Environment
Directive form of economic management
Increasingly market driven
Monopolistic infrastructure
High-level pressure to develop PS-2000 quickly
Extensive set of upstream industries
Indicator-based insentives for upstream enterprises
Incentives based on market signals
Stable user requirements:
High performance for core set of data parallel tasks, real-time capability, high reliability, moderate computational precision, upward compatibility between generations
Stable funding for R&D (until 1992)
Close links with supporting industries, especially with production facilities. Links based on administrative ties, NPO structure
Weakening links with supporting industries, increasingly based on mutual interest and persuasion
Strong market
Very weak market
USSR integrated politically
Ukraine-Russia transactions problematic</td>
<td>Technology
SIMD-oriented architecture,
Reconfigurable bus interconnect,
Autonomous indexing,
Predicate processor,
Distributed memory,
Share secondary storage, etc.

Organizational structure
(NIIUVM) Traditional NII division-oriented structure
(HPC Division) Traditional structure based on laboratories
Flexible Structure based on VNTK, rental collectives, with shared organizational services
Largely functional division based on core systems, subsystems, tasks
Structure oriented towards new tasks</td>
</tr>
<tr>
<td>Technological availability
Examples and ideas from West: ILLIAC IV, STARAN
Reliance on components and subsystems in series production. Generally available in sufficient quantities.
Extensive computer development experience
Some reliance on new technologies (i.e. gate arrays)
Expanded opportunities for use of Western technology. Still difficult to acquire.</td>
<td>Beliefs (design principles)
Build fastest machines possible
Develop systems for real users in industry
As much as possible, use availble technologies, and limit use of immature technologies (Don’t seek to be an industry driver)
Develop systems as quickly as possible
Serve broad customer base
Don’t duplicate technology of other NIIUVM divisions</td>
</tr>
<tr>
<td>Organizational slack
Steady funding through 1991
Decreasing/terminated funding for HPC
Integrated funding stream for HPC
Greater reliance on small-scale/contract work</td>
<td>Strategy
Employ many moderately powerful processing elements in SIMD-based arrangement
Build modular systems
Incorporate existing hardware/software to degree possible
Seek out a variety of customers</td>
</tr>
</table>

Table 5-7 Factors Influencing PS-2x00 Evolution

system gave way to a more market-oriented system, serving a broad user base still served the institute’s interests.

A third basic belief, specific to the special-purpose computing division, was that high performance was a major design goal, and that high performance should be achieved through the use of parallelism, specifically through a single-instruction multiple-data (SIMD) approach.

Clearly there were some trade-offs to be made to accomodate the three pillars of the design philosophy. Achieving higher performance often comes at the cost of lengthened development cycles. But in the early days of development, several environmental factors strongly influenced the development strategy. The combination of an intense national energy campaign and Western restrictions on the export of computers to the Soviet Union increased the pressure to build a high-performance computer for seismic applications in as short a time as possible. Key strategic decisions about development arose out of the guiding principles and these environmental factors in particular. First, the PS-2000 would not serve as a driving force for support industries such as Minelektronprom; it would be built using components already in existence in order to speed up development and reduce the time, effort, and expense needed to develop or acquire customized chips and materials. Second, the machine would be designed as a problem-oriented parallel processor attached to a general-purpose host, thus reducing the hardware and systems software development effort, preserving much of the operating environment familiar to users of NIIUVM products, and making it possible to use the machine within the framework of the aggregate systems of computer technology and software (ASVT and ASPO).

A third environmental factor which strongly influenced the speed with which the PS-2000 could be brought into production was the relatively close relationship between NIIUVM, the prototyping factory, and series production facilities at SPZ. The ‘‘administrative gap’’ between these entities was small, they were geographically proximate, and the channels of communication and cooperation were in place long before the PS-2000

went into series production. The NIIUVM projects reflect the benefits and difficulties of conducting research and development within an important organizational form in ministerial branch science, the scientific-production association. They demonstrate the benefits of a relatively close, long-term association between R&D and production facilities and a strong industrial orientation, but also reveal some of the problems inherent in the ‘‘union of independent organizations,’’ particularly during a period of greater decentralization.

The primary sponsor for the machines, NIIGeofizika, was a strong source of environmental influence over the PS-2x00 machines. Its applications determined the basic set of requirements and the SIMD approach was verified as very applicable to many seismic applications. One of the key design decisions–the use of a reconfigurable bus interconnect–was agreed upon only after specialists had confirmed that it would support the primary applications.

NIIGeofizika remained the primary sponsor through the end of 1991 and its requirements and funding provided a stable, long-term foundation for development. Its on-going sponsorship was one of the important factors underlying a key characteristic of the PS-2x00 family: the continuity in architecture and design from one generation to the next. Improvements were largely incremental, directed at increasing processing speed, improving computational precision, providing greater amounts of primary and secondary storage, releaving bottlenecks such as I/O throughput, improving applications and systems software, etc. Even the shift to a MIMD/SIMD architecture in the PS-2100 simply reflects the aggregation of multiple basic SIMD modules to achieve higher total performance. The basic architecture proved satisfactory for the primary applications; there was little need to alter the fundamental approach. A related environmental factor was the need to maintain a level of software compatibility between one generation and the next because of the growing installed base of PS-2000 computers.

During the late 1980s and early 1990s when demand for the PS-2100 grew quite weak, developers were nevertheless focusing on developing yet another generation, more expensive, with much greater memory and numbers of processing elements, etc. Supporting this development was not investment by NPO Impul's with the prospect of a large market for such machines but the continuing support of the primary sponsor.

Seismic applications were not the only ones to influence development of the PS-2x00 line. A strategy stemming from the desire to cultivate as broad a market as possible was to incorporate features which would be useful to users outside of geophysics if they did not conflict with NIIGeofizika requirements. The atomic energy industry in particular had strong requirements for real-time features and reliability as did the space industry which used the systems for real-time processing of satellite data and control. Other potential users (as well as the geophysicists) desired greater computational precision than the 24/48-bit PS-2000 formats allowed.

The ability to develop all the PS-2x00 computers was strongly influenced by the availability of know-how, components and other supplies, and development tools. The Western experience with the ILLIAC IV and STARAN computers gave Soviet developers considerably inspiration. These machines pointed out a possible development path, and their existence encouraged developers to think about how machines of this nature could be developed in the Soviet Union. Much of the knowledge critical to building a usable, industrial machine had been built up during more than a decade of real-world computer development at NIIUVM.

The actual implementations were sharply constrained by the component technology available, however. Some design features, associative memory in particular, were judged not feasible given the state of Soviet microelectronics. The decisions to use existing components and production technology had a number of implications. The size of the

components determined machine size, causing designers to limit the number of processing elements to 64 in the PS-2000 and the word-length to 24-bits. Similarly, a desire to keep the size of the machine to a reasonable level forced the use of a reconfigurable bus interconnect. Although it was clear that this interconnect was less desirable than, for example, a NEWS interconnect, the technology available for each generation of PS-2x00 computers convinced developers to continue using it. As more functionality was packaged into smaller components, it became feasible to increase the word-length, the number of processors, the amount of memory, etc.

The impact of component availability can be seen clearly when the PS-2100 and PS-2000 are compared. The PS-2100 development time was at least two years greater than that of the PS-2000 in spite of the fact that developers had accumulated a great amount of experience building the PS-2000 and the basic architecture of the PS-2100 base modules was very similar to that of the PS-2000. The principal delay factor was the lack of appropriate components (gate arrays) to construct the machine, together with a lack of experience in customizing them for the PS-2100, and only marginally adequate design tools.

The PS-2x00 themselves and the inherent know-how, architecture, and construction also played an important role. Because the PS-2000 user base was quite extensive, developers felt compelled to preserve the basic architecture and assembly-level instruction set. But beyond this, the basic SIMD-oriented approach was suitable enough for the core applications that developers did not feel the need to explore radically new architectures. Developers prefered to apply existing knowledge of this approach to a new generation of similar machines rather than explore significantly different approaches and face steep new learning curves and protracted development times. The next generation would look similar to the current one in large part because this was the technology that the developers were familiar with.

The PS-2000, -2100, and -2300 mark points on a ‘‘technological trajectory’’ which has been remarkably even. Each generation represented an incremental extension of its predecessor: key parameters were improved, functionality was increased in a cumulative manner, the underlying architecture was retained, and compatibility with previous generations was largely preserved. Until funding ended, the direction of the technological trajectory did not change in any essential manner.

We can postulate the existence of a technological paradigm embodying elements already discussed within which the technological trajectory progressed. Elements of the paradigm include achieving high performance and reliability through parallelism and modularity, the use of a SIMD-based architectural approach with independent memory indexing and activation processors within the processing elements, the incorporation of a linear, segmented-bus interconnect system, construction using standard components and subsystems, and other elements. In other words, the machines already built served as the models or patterns on which further developments were based.

Underlying this consistency were a number of factors which remained largely invarient during the development of these three machines. Table 5-7 shows little variation in user requirements, funding stability, design principles, and strategy. The existing approach met development goals adequately and was reinforced by the need for intergenerational compatibility, and technology was (or became) available to support a strategy of continuity. Thus the ‘‘selection environment’’ which gave developers signals about which developments were possible and beneficial strongly favored the continuation of the existing trajectory.

5.6.2 The Organization

The traditional structure of NIIUVM and the prevalence of similar structures throughout the USSR was discussed in section 5.5.1.1. The basic structure, with the vari-

ous levels of job specifications, was established by the central authorities. The creation of new laboratories and divisions also had to be approved by higher-level authorities. As a result, while research projects naturally differed greatly from one organization to another, their basic structures were very similar throughout the country.

The specific laboratories and their research domains was determined largely by the nature of the technology being developed and the existing organizational structure. When a new task or development project arose, it was usually assigned to one or more existing laboratories/divisions; if the work was a significant enough departure in nature or scale from existing work, a new laboratory or division could conceivably be created, but only after multiple levels of approval from higher authorities.

During the reform period, the organizational structure of NIIUVM and the problem-oriented Computing Division changed in significant ways, leading to more flexible and autonomous structures. The factors influencing organizational structure most strongly are summarized in Table 5-8. These changes reflected the new opportunities created by changes in legislation, the growing financial crisis for NPO Impul's as a whole, and sometimes conflicting views on which organizational structures served best served the needs of the association, NIIUVM, and its constituent divisions.

Changes in the legal environment of state enterprises and associations had a significant enabling impact on organizational structure. The June, 1987 Law on State Enterprises (Associations) gave the leadership of the NPO the authority to alter the internal structure. Other laws defined the conditions for creating other organizational forms, including small enterprises, temporary scientific-technical collectives, rental collectives and cooperatives. While they did not explicitly mandate structural change, they a) made it possible to create a variety of new, legally recognized organizational forms, and b) gave the organization itself much greater authority to make such changes. As other organiza-

Environment	Technology
Laws establishing norms for job titles, wage levels Involvement of higher-level officials in approving changes to organizational structures **Legislation allowing alternative organizational forms** **Legislation giving individual institutes the authority to determine their own structure** **Legislation implementing *khozraschet* principles at institute and sub-institute levels** Strong market for NPO Impul's products **Rapidly declining market for NPO Impul's computer products, especially HPC**	Functional tasks of R&D programs **Some movement away from large-scale projects to smaller scale projects, contract work**
	Organizational Structure
	Traditional NPO, institute, division structures oriented towards basic R&D, production tasks. **Weakening link between R&D and production** **Appearance of more flexible and autonomous VNTK, rental collectives, small enterprises**
Technological Availability	**Beliefs**
Examples of organizational structure and structural change at other organizations	**Key to survival is maintenance of integrated structures** **Key to survival is giving organizational components greater responsibility to fend for themselves** **Key to survival is retaining core personnel**
Organizational Slack	**Strategy**
Stable funding for R&D at NIIUVM **Decrease in government funding for R&D** **Implementation of *khozraschet* principles**	**Maintain integrated divisional and institute structures** **Create flexible, autonomous organizational structures** **Seek foreign partners**

Table 5-8 Factors Influencing Organizational Structure within NIIUVM, Problem-oriented Computer Systems Division

tions throughout the country took advantage of the changes in legislation they became examples to decision-makers at NPO Impul's.

The actual impetus to experiment with new organizational forms came from other quarters. The deepening financial crisis at NPO Impul's forced management at all levels to search for was of improving the efficiency of work, find alternative funding sources, and increase the ability to apply existing resources in more effective ways. The VNTK, rental collectives, and small enterprises, taking responsibility for their own financial status helped accomplish these goals through their flexible organization, the strong incen-

tives they had for finding contracts, and regulations which permitted greater freedom in setting wage levels, etc.

The creation of these new organizational forms required a shift in beliefs about the most appropriate organizational forms, a shift which has by no means been made willingly by all at NIIUVM. In spite of its inflexibility, the traditional NII structure had served the NPO Impul's quite well over the years, especially in terms of integrating the many development projects carried out through the institute. As the economic crisis grew in magnitude, the primary goal increasingly became survival, and the preservation of the collectives. The strategy of greater decentralization and autonomy of the low-level organizational structures was not adopted easily by high-level management which recognized that the cost of decentralization could very well be a resulting fragmentation of the research of the institute as a whole.

The choice was a difficult one. The nature of the market (or market prospects) for large-scale products which required inter-divisional coordination was not strong enough to alleviate the pressures on the individual organizational structures to find whatever sources of revenue they could. To some, the administrative burden of centralized coordination was an unwelcome overhead, reducing the individual structures' viability. To others, the ability to maintain the integration between NIIUVM divisions was the key to future survival. An important, but unknown variable for NIIUVM was how long it would be before the market for the integrated hardware/software process control and data processing systems recovered. The more appropriate organizational structure to no small degree depends on the answer to this question.

The Problem-oriented Computer Systems Division was very much affected by the forces just described at two levels. As a division, this organization was one of the constitutent parts of NIIUVM and felt the tradeoffs between operating as an autonomous unit

finding and executing its own contracts, and remaining a contributing element of the broader NIIUVM research program. The same forces were operating at the sub-division level as well. As the need to find additional revenue sources grew and funding for the PS-2300 grew inadequate, the individual VNTK had to work on whatever projects they could find which would bring in revenue. However, the likelihood of fragmentation to the point were future work on an HPC project is hindered by intra-divisional organizational factors is less than for the institute as a whole. The division (the rental collective) not the constitutent VNTK has its own back account. Although diverging (the software collective continues to develop PS-2100 applications which the architecture collective is working on industrial computers based on the Intel family of microprocessors), the work of each VNTK is closely monitored and coordinated by Itenberg. The core set of PS-2x00 engineers, although working in different VNTK, is small and their experience in working together is great enough that it is likely that will continue to coordinate their efforts and seek projects which will benefit them as a team. Should funding for HPC reappear, it is still possible to reconstitute the core development team.

5.6.3 Prospects

What are the implications of these changes on the future of HPC activities at NIIUVM? The reforms and their consequences have brought about some improvements in the manner in which R&D is conducted, but these have been overshadowed by developments which have made it extreme difficult to continue development of the PS-2x00 line. Without a market and now without funding for HPC development, the prospects for HPC development are, at best, dismal.

Nevertheless, the reforms have brought some improvement in the manner in which R&D is conducted which are likely to be beneficial in the future. The shift away from a centralized planning and supply system and the very real possibility that traditional fund-

ing sources would dry up forced people like Itenberg to spend more effort trying to find and cultivate potential supporters and customers. These efforts include rudimentary marketing, more extensive and intensive meetings with those who have applications potentially suitable to PS- style high-performance computers, publicizing at conferences, etc. Similar activities have taken place for years as NIIUVM sought to expand its market, but they have taken on new urgency and developers are becoming each of these remains under the firm administrative control of Itenberg. Only the rental collective (i.e. the division) has its own bank account. much more sensitive to the real needs of actual and potential users. This, plus growing contact with the West has the potential for increasing the rate of idea generation.

Some aspects of ‘‘coalition-building’’–gathering the support necessary to obtain financing, materials, tools, and know-how–has become easier. It is no longer necessary to navigate through long chains of Ministry and government planning officials. Most series produced material inputs and components, either domestic or foreign, are available at a price. Negotiations can be carried out directly between suppliers and customers who now deal on the basis of self-interest rather than centralized directives.

In the short term, however, low organizational slack still make it very difficult to acquire inputs, and the poor state of the economy make finding contracts and other support a very time-consuming process. The geo-political distance from former customers and suppliers in Russia also severely hinder activities. Although NIIUVM does have a pilot production plant which can carry out low-volume production and generate income, it lacks the large-scale production facilities that used to be provided by SPZ and has never had discretionary use of the funds generated through the sale of SPZ products. Without the potential for generating income, NIIUVM will have a difficult time attracting investment and be forced to rely on piece-meal contract work.

Although NIIUVM in principle has creater opportunity to forge ties with the West and acquire foreign inputs, it is still difficult to do so. The information and financial flows between the West and Severodonetsk, located in a remote area of Eastern Ukraine, still move predominantly through middlemen in Russia. Consequently, acquiring information about Western products and companies and making the exchange from the Ukrainian kharbovantsy to the Russian rule to foreign currencies is time consuming, costly, and uncertain.

Some developments have had a positive impact on the product development cycle at NIIUVM. The introduction of temporary, flexible teams has enabled Itenberg to customize teams for the tasks at hand and in principle draw human resources from all parts of NPO Impul's. The concentration of the appropriate human resources is likely to help reduce R&D cycles. In addition, a stronger contract orientation is likely also to shorten development cycles by forcing engineers to meet the terms of contracts, or forfeit payment.

In the short term, however, these improvements will often be overshadowed by the amount of time engineers spend search for contracts or needed supplies, or simply caring for their families.

In sum, although seeds have been laid for successful R&D in the future, the prospects for HPC development are very poor. Successful development in this domain clearly depends on developing an HPC product for which there is a market. The development of HPC systems which are competitive with Western systems now available will require the use of components which, like the 17,000 gate CMOS chips, are currently not available from the domestic industry. To regain competitiveness, Itenberg may have to find specialized niches and increasingly incorporate Western components.

CHAPTER 6. THE MARS PROJECTS

6.1 Introduction

In this chapter we study the MARS computers and the organizations involved in their development. The MARS (Modular, Asynchronous, Extendable Systems) multiprocessor computers were part of the Soviet Union's START program, created in part as the Soviet answer to the Japanese Fifth Generation efforts. Centered in Novosibirsk, the work represents a very high-profile project carried out within the USSR Academy of Sciences (AN SSSR). Carried out under the patronage of G. I. Marchuk, chairman of the State Committee on Science and Technology (GKNT) and later President of the USSR Academy of Sciences, the START program as a whole and the MARS project in particular enjoyed some of the most favorable conditions within the Academy of Sciences–save perhaps in institutes with dual Academy/ministerial subordination–for developing new computers. While not necessarily typical of Academy efforts in computer development, therefore, the MARS project does represent something of a best-case. Many of the problems experienced in MARS development, particularly those regarding environmental factors, were shared by other Academy projects, but to an even greater degree.

Like most Academy of Sciences projects, the MARS computers incorporate a strong research component. The machines were viewed not only as vehicles for providing the scientific community with high-performance computing capability, but also as a means of experimenting with new architectural approaches. We therefore expect to see a different set of factors guiding this research than work done chiefly for industrial use. At the same time, being applied research, the work should strongly reflect the opportunities and constraints of the technological, organizational, and political context within which it was carried out. The MARS project can contribute to our understanding of high-technology R&D within the Academy of Sciences both prior to and during the reform period.

The MARS research took place within a changing organizational environment. The groundwork was laid within the Computing Center of the Siberian Department of the USSR Academy of Sciences (VTs SO AN SSSR). Much of the implementation work was carried out in the same facilities, but within a new organization, the START temporary scientific-technical collective (VNTK). Following the termination of START, a new–although more traditional–Academy of Sciences institute, the Institute of Informatics Systems (ISI), was created based on many of the Novosibirsk laboratories which had participated in START. Additional organizations reflecting a variety of organizational types were created as the *perestroika* reforms progressed. The organizational evolution surrounding the MARS project provides insights into the forces influencing organizational development within the Academy of Sciences during the Soviet Union's last decade and the relationship between technologies and the organizations within which they are developed.

This chapter is organized around the phases of development of the organizations involved in MARS research. Following a brief history of research at the Computing Center of the SO AN SSSR, we examine both organizational and technological developments during START's formation, the years of its existence, and during the period following its termination through the dissolution of the Soviet Union.

6.2 History of VTs SO AN SSSR Research

The Siberian Department of the USSR Academy of Sciences, the first territorial department of the AN SSSR, was established in 1957 to provide a strong research base for the rapid development of Siberia. All Academy of Sciences research institutes east of the Urals were subordinated to the SO AN SSSR, and many new institutes were created. From 1957 to 1975 the chairman of the SO AN SSSR was M. A. Lavrent'yev; from 1975 to 1980, G. I. Marchuk; and from 1980 to the present, V. A. Koptyug.

Half of the scientific personnel and research institutes are concentrated in the Akademic City *(Akademgorodok)* in Novosibirsk. The *Akademgorodok* is home to over twenty Academy research institutes, over 100 subsidiaries and design bureaus of branch scientific institutes, and–during the 1980s–a combined staff of over 70,000 individuals. The accumulation over the years of organizations associated with branch science has come to be known as the ‘‘implementation belt.’’ Its purpose is to facilitate the transfer of scientific results into production [Soan82].

Under the leadership of Marchuk and his successor, A. S. Alekseyev, the Computer Center of the SO AN SSSR became one of the major computer science research facilities in the Soviet Union. Marchuk, Alekseyev and others such as A. P. Yershov (one of the fathers of Soviet computer science) and V. Ye. Kotov had important voices in high-level computer-related policy commissions and committees within the Academy of Sciences and the State Committee on Science and Technology.

The principal areas of research of the Computer Center were the development of mathematical models of the atmosphere and oceans; methods of numerical analysis; computer applications for control processes in enterprise and territorial automated management systems (ASU); research and development of theoretical foundations for information processing; systems programming; development of techniques for mathematical computer-aided simulation; and the development of collective-use computing facilities shared by numerous *Akademgorodok* institutes. The methods and models enabled research in weather forecasting, general circulation of the atmosphere and oceans, climate theory, nuclear reactor analysis, and especially seismology [Soan87].

Until the late 1970s, virtually no work was done on designing and building computers. Much theoretical work, such as that of Vadim Ye. Kotov on the modeling of concurrent systems, and systems software development was carried out. The latter included

automated programming systems (Al'pha), multi-language translation systems (BETA), the first Soviet experimental time-sharing systems (AIST-0), and early translators for such languages as Algol-60, Al'fa-6, EPSILON, ALMO [Metl73; Yers80b]. During the 1970s and 1980s the VTs SO AN SSSR was the principal organization in an effort to develop the *Sibir'* regional network, part of the larger academic network *Akademset',* and became very involved in the development of such automated management systems (ASU) as Barnaul and view/Sigma [Alek83; Bobk78; Eko79; Mche85].

Prior to the late 1970s, the laboratories which were to become involved in START had done little applied work, and almost none in the area of high-performance computing. Following a change in the USSR Academy of Sciences' charter in 1977 encouraging more applied research [Fort90, 50], G. I. Marchuk began to conceive of his organization's becoming a pioneer in high-performance computing. To proceed past a paper design, projects within the Academy of Sciences had to gain technical, financial, and political support. Marchuk had close ties with V. S. Burtsev, the director of ITMVT at that time, as did Kotov. The three of them had served together on an Academy of Sciences commission which addressed supercomputing issues. Over the course of many conversations with Marchuk, Kotov, and others, Burtsev agreed to provide some technical support for a development team from Novosibirsk. As Marchuk moved up through the scientific bureaucracy, from director of the Computing Center of the SO AN SSSR to president of the SO AN SSSR (1975) and chairman of the GKNT (1980), he became increasingly able to push proposals into the top levels of government. As a result, it became increasingly feasible to carry out applied high-performance development in Novosibirsk.

In 1978, Marchuk and Kotov published a conceptual framework for a modular, asynchronous, extendable system (MARS) computer. The projects with origins in this work, discussed in section 6.6.2, marked the Computer Center's first serious attempts at applied

computer development and formed one of the cornerstones of the START program discussed below.

6.3 The MARS Conception

The basic principles of MARS and some of the reasoning behind them were first spelled out in [Marc78; Marc78b] and were repeated in various forms over many years in publications about MARS research. A recent article published in the West explaining the origins of the MARS philosophy and design goals is [Koto91].

In [Marc78], Marchuk and Kotov presented their analysis of key trends in the development of computer technology, and identified one possible approach to the development of computers of the next generation. The principal trend, in their view, was a broadening of the sphere of application of computer technology, the compound or systemic character of the problems to be solved, and, consequently, the transition from single units of machines with a traditional (Von Neumann) architecture to computing systems with a variety of configurations and a wide range of capabilities and purposes [Marc78, 4,9,12]. Given this trend, a very relevant problem was the architecture of systems oriented towards multiple modes of use, from time-sharing to real-time processing, to information retrieval, etc. [Marc78, 14]. Extendable systems which could be adapted to a variety of application domains were of particular interest [Koto86b, 277; Koto91].

Providing specialization and adaptability to application domains would require ‘‘the selection of a base set of specialized units for information storage and processing, the creation of subsystems and devices which are programmable to specific operation modes and algorithms, and the development of the principle of reconfigurability of the system as a whole’’ [Marc78, 15]. At the basis of the architecture should be a theory of the analysis and synthesis of computational structures.

A second major trend was the miniaturization of the component base, making it possible to incorporate multiple microprocessors as a basic component of a computing system. It was felt that this trend would also result in fundamentally new means of system design, as the design of such components increasingly took on the characteristics of the design of large and complex systems [Marc78, 11].

A third trend was the stready improvement of the man-machine interface through higher-level programming languages.

Kotov and Marchuk felt that the architecture of promising computers of the future would be a logical extension of those ideas and principles which to one degree or another were already to be found in machines in existence at that time. Section 2 of [Marc78] identifies a host of Soviet and Western machines which are grouped according to architectural features relevant to the discussion. Their survey touches on general-purpose multiprocessors (Burroughs 7700, UNIVAC 1108, El'brus), vector-pipeline systems (Star-100, ASC), SIMD architectures (SOLOMON, ILLIAC-IV, PEPE, STARAN), and homogeneous computing systems with programmable interconnect systems. They discuss distributed memory and associative memory approaches, and asynchronous computational methods such as data flow.

The MARS Conception reflects an effort to fuse many of these ideas into a unified computing system, maximizing the strengths of each approach and minimizing the weaknesses. The key architectural principles, discussed in greater detail below, were [Marc78, 5,33-41]:

- parallelism (both in processing, data access, and control);
- decentralization of information processing and data flow;
- asynchronous interaction of devices and processes;

- hierarchical structure (both functionally and in terms of the components), with multiple virtual layers in the system;
- specialized systems components, hardware implementation of complex data processing functions;
- self-identification of data and processes (tagged architecture);
- modularity, reconfigurability.

Parallelism. Collectively, the machines surveyed exhibit parallelism at a number of different levels: parallel processing, parallel access to memory, and parallel control. Parallel processing can take the form of parallelism of iterations, independent operations on a set of objects, parallel processing of interacting branches, asynchronous parallelism, etc. Kotov and Marchuk tried to fuse many of the ideas into a unified computing system; a goal of the MARS project was to incorporate many of these kinds of parallelism in the same system, in an integrated fashion.

Decentralization. To accommodate growth in the number of processors or other devices without overloading certain system resources, some form of decentralized processing would be necessary, enabling subsystems to function autonomously on local data and communications. Marchuk and Kotov felt that the optimal variation would be the combination of the principles of centralized information processing with the capability of widespread decentralization [Marc78, 34].

Asynchronous control. They felt that asynchronous control was ‘‘necessary’’ for implementing the complex interaction of a highly parallel and (partially) decentralized system. Components would interact with each other through shared resources such as memory, buffers, communciations channels, etc., but would not have direct control over each other. Kotov’s earlier work [Koto66] had developed ideas of a centralized asynchronous computation with shared memory. Other researchers, including Torgashev and My-

asnikov in the Soviet Union, and Miller, Dennis, and Rumbaugh in the West had worked on ideas of decentralized asynchronous processing in their work on recursive and data flow machines [Glus74; Denn68; Rumb75]. Seeking again to fuse many different threads of research into a single system, Marchuk and Kotov suggested that the optimal solution would incorporate the strengths of the centralized and decentralized approaches, minimizing their differences. As a result, the MARS model combined various asynchronous computational models.

Hierarchical structure. The computational models in effect at different levels differ in the nature of the activation conditions used. In particular, control mechanisms could be divided into three categories [Marc78b, 19-20]: unconditional, conditional, and data flow. These control mechanisms could be modeled using an extended Petri net notation[1] and trigger functions. Trigger functions are boolean expressions indicating readiness of some system component, be it an operation, expression, module, etc. [Koto66]. At the level of expressions with scalars and vectors, the data flow model is used. At the level of larger fragments (statements, program modules) which share common memory, an asynchronous control-driven mechanism based on trigger functions is used [Koto86b, 279].

Marchuk and Kotov divided the world of computational processes into two categories: user, or application processes, and systems processes. To handle the complexity of the contemporary systems used to run these processes better, they felt it useful to view a system as a series of nested ‘‘virtual machines’’ in which each layer represents a virtual

[1] Petri nets are a formalism for representing systems with concurrency or parallelism. A Petri net is a graph with two types of nodes–places and transitions. Places are drawn as circles while transitions are drawn as bars (or rectangles in [Marc78b]). Directed arcs (arrows) connect places to transitions and transitions to places. For each transition, the directed arcs define its input places (arc from place to transition) and its output places (arc from transition to place). A Petri net is executed by defining a marking and the firing transitions. A marking is a distribution of tokens to the places of the Petri net. A transition is enabled whenever all of its input places have one or more tokens. A transition fires by removing one token from each of its input places and adding one token to each of its output places.

machine which ‘‘runs’’ on a lower level virtual machine. They drew this concept from the work of Dijkstra on the THE multiprogramming system [Dijk68], and pointed to the IBM VM (Virtual Machine) operating systems as another example.

The functional hierarchy could also be reflected in the hardware itself, however. There were a number of examples of such systems. The central processors of the CDC 6600 and the Burroughs B6700 consisted of a number of subprocessors and registers. The ILLIAC IV design originally called for four array processors, each consisting of 64 processing elements. Marchuk and Kotov felt that some form of hierarchical organization was necessary to implement highly parallel computation most effectively.

Specialized systems components. Before the introduction of RISC concepts, the trend in instruction sets had been towards increasing numbers of instructions of increasing complexity. Marchuk and Kotov also felt that the most appropriate way to adapt a computer to specialized application domains was through the creation of application-specific, complex operations implemented either in hardware or in microcode. These measures, it was felt, would raise the level of the machine language, increasing programming effectiveness and reliability; decrease the portion of computation executed in software, improving performance; and increase the reliability and modularity of the entire system. Marchuk and Kotov point to a number of systems with hardware or microcode implementation of high-level operations, including the Burroughs B5700/B6700, FORTRAN and SNOBOL machines, and machines including hardware support for operating systems and database operations [Marc78, 38-39].

Tagged architectures. Tagged architectures, allowing low-level specification of data types, information about origins of operations, labels, state of readiness of data, etc. could be used to provide considerable control at low levels in the architecture. They

could provide greater structural flexibility and simplify programming and process control. These ideas had already been implemented in the Burroughs and El'brus machines.

Modularity and reconfigurability. To maximize the average performance of the system across a wide spectrum of applications, Marchuk and Kotov held that a rigid, uniform architecture could not be used. Such systems, while achieving high performance on certain kinds of problems, ran slowly when there was a mismatch between the nature of the problem and the architecture of the machine. To overcome this difficulty, Marchuk and Kotov would rely on the principles of modularity and reconfigurability. Rather than rely on one type of specialized system, a full MARS configuration would include different types of subsystems, each oriented towards a different class of problem. Furthermore, it should be possible to modify the structure of systems to match the characteristics of a given application. The challenge, of course, was not to lose performance overall as a result of this flexibility. The key was the integration of the core MARS principles in a manner that was without internal contradictions. They felt that these conventions should be reflected not only in the hardware and architecture, but in the software as well.

The belief was that the conventions not only were compatible, but could all be integrated into a coherent whole, multiplying the effectiveness of each [Koto91, 44]. Three basic principles which would facilitate this were [Marc78b, 6-7]: (1) a unified set of rules and means of composing systems (and programs for them) from modules of various levels; (2) evolution of the functional capabilities of the system and its ability to adapt through hardware constructs, virtualization, and specialized modules; and (3) a unified principle of asynchronicity for organizing the base system, and the computational processes and programs.

MARS represents a series of experiments seeking to further the knowledge of concurrency and the assumptions of the design philosophy through the construction of a number

of artifacts exhibiting different designs and application paradigms. In the MARS projects, in keeping with the basic philosophies, a multiprocessor is viewed as a ‘‘(statically) reconfigurable structure of loosely and/or tightly coupled based processing modules’’ [Koto91, 37]. Two key projects were the MARS-M and the MARS-T. In addition to the overarching philosophies just mentioned, each of these projects exhibited a number of more specific design goals.

Kotov, G. I. Marchuk, and Yu. L. Vishnevskiy decided in 1980 to start an implementation of the MARS ideas. They felt the basic ideas had been sufficiently worked out that they could begin designing a concrete architecture. Having persuaded V. S. Burtsev to agree to support the project by making ITMVT facilities and tools available, they felt such a project was feasible. Largely thanks to Marchuk’s efforts the GKNT agreed to fund initial design work which was carried out between 1980 and 1985.

6.4 The Pre-START Years (1983-1985)

6.4.1 Formation of START

The START program and the research carried out within it were born out of the confluence of three major streams of activity: research in concurrent architectures and artificial intelligence in progress in the Soviet Union during the early 1980s, the Japanese Fifth Generation Project, and changes in Soviet legislation which made it possible to organize research and development in new ways. In the years following the announcement of the Japanese Fifth Generation Project in 1981, Soviet researchers, particularly in the area of artificial intelligence, began discussing ways of incorporating their efforts into a Soviet counterpart to the Japanese program. A new form of organization called a ‘‘Temporary Scientific-Technical Collective’’ (VNTK), legalized in 1983 while such discus-

sions were taking place, became the vehicle in which the ‘‘Soviet Fifth Generation Project’’ was carried out.

The Japanese Fifth Generation Project placed artificial intelligence research at the core of a program oriented towards the development of a new generation of computers. Anticipating that information processing systems of the future would play crucial roles in increasing the productivity of office workers, cultivating information as a national resource comparable to food and energy, assisting in saving energy and resources, and coping with an aging society, the Japanese initiated a program to develop computers with the following characteristics: increased intelligence and ease of use to better assist man; the ability to process information conversationally using everyday language; the ability to store knowledge and carry out learning, association, and inference functions; etc. [Moto82]. The Japanese announced their plans in October, 1981, at the International Conference on Fifth Generation Computing Systems in Tokyo. The Institute for New Generation Computer Technology (ICOT), formed on April 14, 1982, was at the core of the project [Feig84, 10]. Only about forty researchers worked at ICOT itself, but over 150 others in Japanese industry worked under their direction [Feig84, 27]. The anticipated total budget for the ten-year project was $850 million [Feig84, 115].

Some of the earliest discussions about the creation of a comparable Soviet program took place between Aleksandr S. Narin’yani, Viktor M. Bryabrin and Enn Kh. Tyugu early in 1983. Each of these individuals had been engaged in AI research for many years, and felt that it was necessary to ‘‘do something,’’ to organize a higher-level, focused program of research in AI. Narin’yani, Bryabrin, and Tyugu worked together on preliminary drafts for the formation of a new organization, but drew Vadim Ye. Kotov into the discussions early on. Unlike the other three, Kotov had strong ties to Guriy I. Marchuk, who at this time was chairman of the GKNT.

Kotov proposed emphasizing the architectural aspects of the program and using artificial intelligence problems as benchmarks. He approached Marchuk who, after some discussion, agreed that the proposal was worth trying to push though policy-making channels [Afan87]. Kotov saw an opportunity to acquire additional support for MARS and worked this project into the proposal. Narin'yani, Bryabrin, Tyugu and Kotov spent much of the summer of 1983 in the GKNT headquarters in Moscow drafting a more formal proposal for higher-level authorities.

Within the Soviet science and technology community and in bodies such as the Coordinating Committee on Computer Technology[2] of which Marchuk was chairman there was considerable discussion about possible responses to the creation of national computing projects in Japan and Western Europe. While some proposed large-scale, long-term projects with massive government funding, others, including the founders of START, proposed more modest approaches for a number of reasons. Large-scale computing programs in the past, such as that to develop a national system of automated management systems (ASU) had not lived up to the initial promises [Mche85], and many had very legitimate doubts that a large-scale program would succeed this time. During the early 1980s the Soviet economy was experiencing steady decline and policy makers were less eager to commit to large, long-term investments in computing. Furthermore, Soviet industry was not capable of producing the advanced components, packaging, and cooling systems needed to compete in the race to develop the next generation of supercomputers [Koto91b, 30].

The founders of START wanted to create an efficient, controllable program that would build on what strengths the Soviet computing community had. In contrast, a large-

[2] Created in 1979, this body of the USSR Academy of Sciences was tasked with coordinating funding of large-scale academic projects and establishing and supervising links with industrial ministries [Petr79].

scale project would have to rely on funding from the military, a traditional sponsor of computer-science research. The military's involvement, however, would mean greatly increased bureaucratic overhead and reduced freedom in managing the project. The involvement of many institutions would also greatly increase the management difficulties. This was to be avoided, if possible.

Marchuk did not fully share this aversion to large-scale programs, however, and made some effort to expand the project. He invited the directors of other computing institutes such as V. S. Burtsev from ITMVT and V. V. Przhiyalkovskiy from the Scientific-Research Center for Electronic Computer Technology (NITsEVT) together with other officials, including some from the Military-Industrial Commission, to participate in a committee to develop a formal proposal. The final product consisted of an industrial portion and a fundamental research component. In an effort to return to the notion of a smaller, more flexible, less encumbered approach, Kotov, Narin'yani, Tyugu, and Bryabrin decided to continue pushing the smaller, fundamental research portion, leaving the others to promote the industrial component, if they desired. Apparently the industrial component never became viable, even though it officially existed.

The laboratories of the four founders–the Computer Center of the Siberian Department of the USSR Academy of Sciences in Novosibirsk, the Institute of Cybernetics in Tallinn (IK AN ESSR), and the Computer Center of the USSR Academy of Sciences Moscow (VTs AN SSSR)–formed the core of the project. A Novosibirsk laboratory of the Impul's Scientific Research Association in Severodonetsk, Ukraine was also incorporated as the sole formal industrial participant. The Novosibirsk laboratory was selected largely because of geographic proximity and existing working relationships. Although connections between Impul's and the other START organizations were rather weak, some industrial participation was helpful because of the hardware and construction expertise

that it could provide. The subsidiary of NPO Impul's and the prototype development and design bureaus associated directly with the Computer Center would not be able to provide all the industrial capability necessary to build MARS hardware prototypes, however, and relationships with industry factories outside of Novosibirsk would have a strong impact on the projects' progress.

As word about the new project spread through the Soviet computing community others sought to join but were turned down. The founders felt that the participation of many organizations would be too difficult to control. By keeping the proposal modest, the founders could exist solely on support obtainable by Marchuk through the GKNT and the Academy of Sciences, and not involve the military.

In August 1983, the USSR Council of Ministers passed a resolution ''On Measures for Accelerating Scientific-Technical Progress'' instructing various ministries to establish Temporary Scientific-Technical Collectives (VNTK) and defined in detail the characteristics of such organizations [Ntr85]. The original resolution was rather vague, but stressed that research and development teams, created for limited time periods and pooling the expertise of individuals in multiple branches of science and industry, should be able to conduct R&D in a more efficient manner than had been the case earlier. Such a resolution was made at precisely the time when START organizers were considering such questions, and they adopted the VNTK label and actually helped define what such organizations should be like. In later years they occasionally consulted for other groups trying to organize similar teams.

A number of measures were taken to stimulate productivity. First, the basic nature of an inter-branch, temporary collective served to bring together young, qualified researchers with complementary expertise into a single, high-intensity, high-visibility program. Second, although imposed by law, the three-year duration of the program forced re-

searchers to produce results within a compressed timeframe. Third, a bonus averaging 3,000 rubles was promised each participant if the project goals were achieved within the three years [Koto91b, 31; Manu88].

Although a major goal was the creation of an efficient, controllable, moderate-scale program, the effort to create a new type of structure actually added complexity in the initial phase. The creation of a research organization, particularly one with a new structure required the approval of several ministries at all levels. In spite of efforts to minimize the number of participants, the involvement of multiple ministries was unavoidable. The primary work was being carried out within the Academy of Sciences. Equipment was needed from the Ministry of Instrument-Building, Means of Automation and Control Systems (Minpribor). Financing was to come from the GKNT and the Academy of Sciences. Because START was a new organizational form, the Ministry of Justice, the Ministry of Labor and Wages, Trade Unions, and the Military-Industrial Commission all had to approve the proposal. In each ministry, one had to collect signatures from the lowest levels to the highest, even from ministers themselves. When objections were raised, they had to be addressed and approved by all the other organizations. The final document contained 121 signatures.

The turmoil in leadership of the country further delayed the process. The first draft proposals were written while Yuriy Andropov was General Secretary. Little progress was made while Chernenko was in office. Only in January, 1985 did the USSR State Committee on Labor and Wages and the All-Union Central Council of Trade Unions resolve the practical matters of wages for members of VNTK in the resolution ‘‘On Procedures for the Payment of Wages and Bonuses for Workers of Temporary Teams’’ [Ntr85]. START was finally organized on April 1, 1985.

As the START proposal was being pushed through the Soviet bureaucratic labyrinth, organizers found it very useful to draw strong analogies to the Japanese Fifth Generation project. START was not on the same scale as the Japanese effort in terms of duration or financing, but casting START as a response to the Japanese program both validated START goals in the eyes of policy-makers, and provided a national program which leaders could use to try to demonstrate to themselves and the world that the Soviet Union was a serious participant in leading computer science research at a time when a number of national and international programs such as Alvy and ESPRIT were being created. Phrases such as ‘‘The purpose of START is to perfect and test fifth generation computers...’’ were common in press reports about the new organization [Zakh85; Ntr85; Favo86; Afan87; Koto87]. Not surprisingly, once in existence, START continued to be promoted using *perestroika* terminology adopted by Gorbachev. The project was a means of promoting the ‘‘acceleration of scientific and technical progress and intensification of the national economy’’ [Koto87; Gorb87].

6.4.2 Nature of the Research Plan

Although the founders desired to keep the scale of the project managable, they felt it necessary to pursue advances on many fronts. As Kotov said, ‘‘The transition to the fifth generation is impossible without radical improvement of all the things that make up computer technology. Everything must be updated: the component base, means of communication, software, and primarily machine architecture’’ [Ntr85]. Not only are new components and subsystems required, it was argued, but progress in these areas had to be integrated to a much greater extent than was the case in the traditional ministerial, command-economic structure with its compartmentalization of development [Koto87]. This argument was used to justify the creation of a VNTK form of management in which researchers in different administrative entities could much more easily work with each

other. Furthermore, while critical of the results of the Soviet computer industry, Kotov consistently argued that the Soviet computing community needed to conduct at least *some* original research in all these fields to preserve its intellectual capital [Koto87]. This led to considerable breadth in START-related research, although individual projects did not depend strongly on each other.

The founders initially had wanted to tie all the threads of their existing research into a single, tightly-coupled project. Early descriptions of START speak of the development of the MARS computer, which would integrate a wide range of functionality, from AI to database management, to networks, to high-speed scientific computation. There were discussions about establishing a common representation for all the structures used in the AI components. When all was said and done, however, the degree of actual overlap between projects was slight. As explained below, the difficulty in bringing about this integration was greater than had been anticipated, and the final results of START would be better characterized as an aggregation of individual projects than true joint efforts.

The component research projects were, for the most part, extensions of research which had been in progress for several years and for which some preliminary results had been achieved. Tyugu had begun development on the PRIZ program synthesis system during the early 1970s [Tyug70]; Kotov and Narin'yani had published on asynchronous concurrency in 1966 [Koto66], and Kotov and Marchuk had first published MARS ideas in 1978; Narin'yani had conducted research in various areas of artificial intelligence, natural language processing in particular, during the 1970s [Jako85; Nari85]. It could be argued that this approach had to be taken to produce the necessary results within the three-year limit imposed by the VNTK legislation. As we shall see, the MARS-M could hardly have been built from scratch in three years.

6.5 The START Years (1985-1988)

6.5.1 MARS Research

START-related research was broad-based. At the Computer Center of the SO AN SSSR, four laboratories were involved. Under Yuriy L. Vishnevskiy, the parallel systems laboratory worked on creating the MARS-M computer. Vadim Ye. Kotov's laboratory worked on the development of two parallel languages, BARS and Pol'yar. Aleksandr G. Marchuk's laboratory focused primarily on developing the 32-bit Kronos microprocessor and the MARS-T parallel system incorporating the Kronos. Marchuk's laboratory also developed a computer-aided design system for designing VLSI chips, the Kronos in particular. Aleksandr S. Narin'yani was the head of a laboratory involved in artificial intelligence research. In Moscow at the Computing Center of the USSR Academy of Sciences, Yu. G. Yevtushenko and Viktor M. Bryabrin developed systems and applications software, in large part for personal computers. At the Institute of Cybernetics in Tallinn, researchers in the Systems Software Department under Enn Kh. Tyugu developed object-oriented software development systems, program synthesis systems, and an object-oriented workstation based on the Kronos processor. In this section we discuss the MARS hardware projects, MARS-M and MARS-T.

6.5.1.1 MARS-M

The 1978 preprints by Kotov and Marchuk presented the high-level principles which defined the MARS architecture. In particular, they discussed dividing a computing process into four different kinds: control, memory access, execution, and communication. The MARS-M was the primary effort to implement these principles. The original name, ''Mini-MARS,'' refered to the fact that this machine was conceived to be a small portion, the numerical processor, of a larger MARS configuration [Koto86b, 280-281]. Besides

being an effort to implement the principles outlined in the Conception, the MARS-M represents a practical effort to explore issues of parallelism at multiple levels within a single architecture. This involved the development of architectural support for parallelism at each level and language facilities for expressing parallelism at the corresponding levels.

The MARS-M incorporated the key principles outlined in section 6.3 (with the exception of hardware tags, which were deemed unnecessary as development progressed), but on a smaller scale than outlined by the Conception. In particular, while the Conception called for execution systems integrating a variety of special-purpose processors (general-purpose arithmetic, vector, associative, symbolic, etc.), a memory system incorporating addressable, associative, and orthogonal memory modules, etc., the MARS-M was limited to a single type of memory, and an execution subsystem incorporating control, execution, and addressing processors.

Kotov and Marchuk communicated these ideas to Yuri Vishnevskiy during the late 1970s and 1980 as they discussed the possibilities of creating a machine based on these principles. Kotov and Marchuk played only a minor role in deciding *how* to implement such an architecture, however.

Following a brief overview of the architecture and construction of the MARS-M prototype, we present a chronology of development, and analyze the factors influencing the development of architectural ideas, and their realization. A more extensive description of the MARS-M can be found in [Doro92].

Architecture. The MARS-M is a shared-memory heterogeneous multiprocessor having a control processor, a central processing unit, a peripheral subsystem, a memory management unit, and multiported main memory. The control processor consists of eight (virtual) systems processors, and the central processing unit consists of a control subsystem,

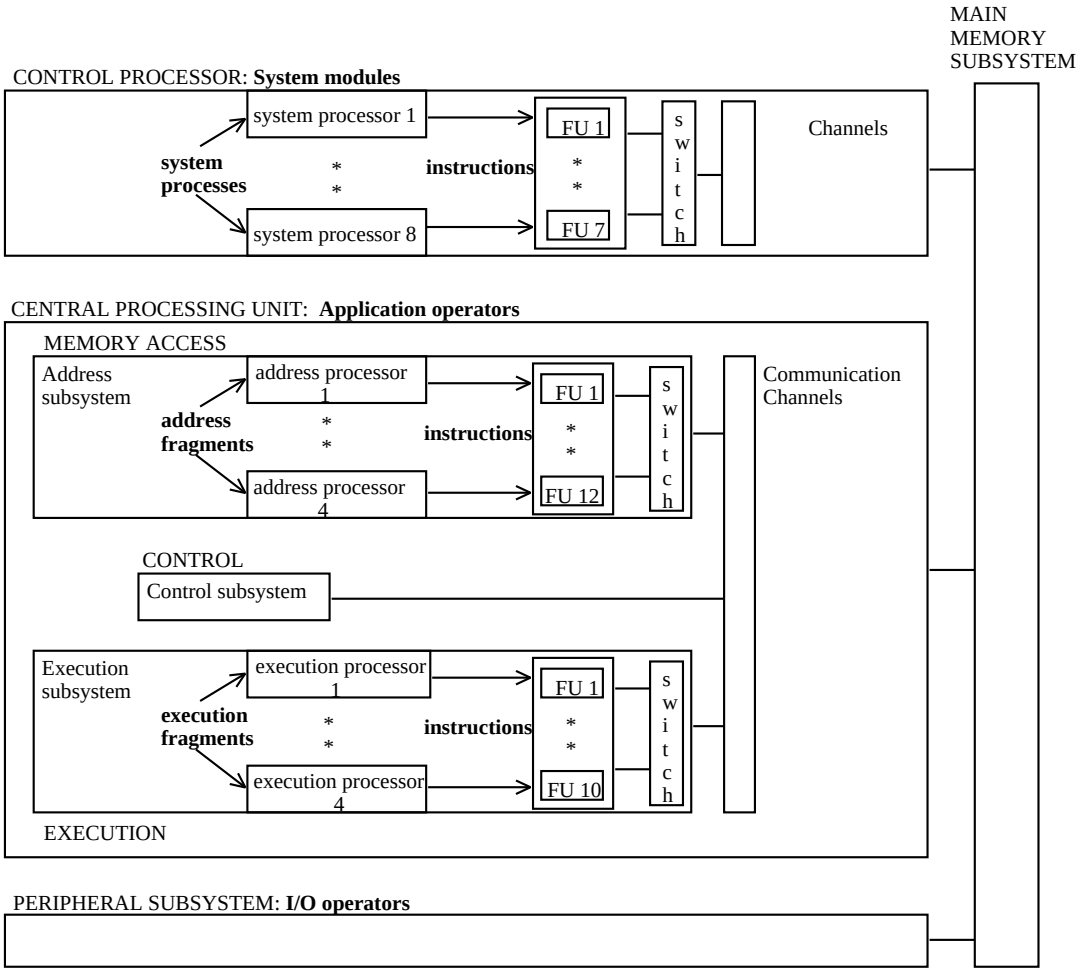


Figure 6-1 MARS-M Logical Structure

four address and four execution processors. These elements constitute the fixed ‘‘skeleton.’’ Figure 6-1 shows the logical structure of the MARS-M.

The MARS-M has the following principal characteristics:

- hierarchical organization of data processing;
- multiprocessing;
- multi-pipelined organization of each processor;

Level	Model of Computation	Language Objects	Execution of the Objects
System	Virtual heterogeneous multiprocessor	Modules and I/O operators	by the control processor and the peripheral subsystem
Application	Decoupled architecture	Application operators	by the central processing unit
Functional	VLIW architecture	Address and execution fragments	by functional units of the address and execution processors

Table 6-1 MARS-M Levels and Their Features

- asynchronous interaction of processors with the help of uni-directional buffer data and logical value transmission channels;
- powerful descriptor mechanism for working with compound data objects like vectors, arrays, etc.;
- adaptation of the MARS-M architecture for solving a specific class of problems by means of a selection of commands for the formulation of representations and processing of compound data objects.

The MARS-M has three architectural levels: systems, application, and functional. Each level has a corresponding computation model, language objects processed at that level, and hardware elements to execute those objects. Table 6-1 shows the levels, together with the computation models and associated language objects. The system and application levels were proposed in the MARS Conception. The third, functional level, represents a lower level needed to support the execution of application operators in the central processing unit.

System level. Computation within modules is defined by so-called control expressions. These contain instructions which call modules, operators, other control expres-

sions, etc. The execution of control expressions is called a system thread. A new system thread can be generated both explicitly by a parallel call operation invoking a control expression and implicitly with the help of a token control mechanism.

In general, this control mechanism is similar to macro (procedure-level) data flow mechanisms. Its specifics lie in the objects whose execution is to be dynamically scheduled: the MARS-M's modules, operators, control expressions, and system operations.

Application level. Like the central processing unit which supports it, the computation model for the application operators is based on a decoupled multiprocessor architecture. Decoupled architectures have two main elements [Smit86]. They have two separate instruction sets and concurrent processing of at least two instruction streams, one for accessing memory and one for performing function execution; and communication between the memory access and the execution processes takes place via architectural queues. Operators in the MARS-M are built from three types of operations: control, address, and execution. The latter two are called address and execution fragments. They are functions for performing complex computation and memory access operations that are typical for an application field. Different applications can have different sets of ''elementary'' address and execution operations. The execution of an address fragment by an address processor, an execution fragment by an execution processor, and control instructions by the control subsystem are called address, execution and control threads, respectively. Up to four address, four execution and one control threads can run simultaneously at this level. These executing threads communicate via queues of different types.

There are two asynchronous mechanisms to schedule parallel processing within an operator: data flow and control token. The data flow mechanism, hidden from the programmer, is built into hardware to synchronize communication between fragments through any queues. The control token mechanism is used by the programmer to impose

additional synchronization control on decoupled multi-thread processing. Here, thread exchange is enforced via special control messages sent through branch queues.

Functional level. At the functional level, the computational model is very-long-instruction-word execution of address and execution fragments. The control subsystem splits an instruction stream generated from application operators (from the *application level)* so some of the instructions are passed to free address and execution processors, while others are executed by the control subsystem.

Each instruction sent to the processors contains a fragment call operation. Fragments are independent program blocks which can consist of multiple basic blocks and a control part containing branches, loop-counters, literals, and other operations. A basic block is a sequence of code that can only be entered at the top and exited at the bottom. Conditional branches and conditional branch targets typically delineate basic blocks.

Four address and four execution processors execute up to eight fragments in parallel. Address fragments compute addresses of elements within structured objects such as arrays or vectors which either have to be fetched from memory into queues, or have to be stored into memory from queues. Execution fragments perform their operations on data being fetched from memory or computed and passed from address fragments via queues external to the subsystem and/or computed and passed by other fragments via the subsystem's internal queues. Instructions are issued strictly in program order but can begin and complete their execution out of order. The reason for this out-of-order execution is the asynchronous communication between processors and memory via channels. A data flow mechanism is used to initiate or suspend the execution of a fragment, depending on the availability of operands in its channel queues.

Both address and execution processors have a horizontal architecture with multiple pipelined functional units. The execution and address subsystems both consist of four

processors sharing a set of functional units. A single very-long-instruction-word contains multiple operations and in each cycle operations can be issued to all of the functional units simultaneously. A unique feature of the MARS-M is that it uses a combination of static and dynamic scheduling for forming and issuing very-long-instruction-words. A so-called fragment compiler schedules individual fragments statically, forming very-long-instruction-words to be executed on individual address or execution processors. These instruction words are combined at run-time by a fragment dispatcher to run concurrently on the functional units.

Since the MARS-M computer is oriented towards scientific computation, the effective processing of arrays and vectors was a primary goal. A number of address and execution fragments implementing typical array/vector operations were developed. In many cases, array-subscript expressions can be rather complex and contain conditional branches. To provide continuous address generation for such expressions, special array address generators were incorporated into the address subsystem, making it possible to generate an address in each clock cycle.

The MARS-M could be adapted for specific classes of application programs by creating specialized problem-oriented fragments. This library-oriented approach was similar in concept to that used in many attached array processors which provided a library of routines accessible by users. The library-oriented approach for fragments made it possible to load the code for all fragments into the distributed instruction memory before a program was initiated and fetch it locally and independently by multiple sequencers at run time. The compiler specified a timetable of all instruction streams to be issued by the control modules in executing any fragment.

MARS-M Performance. Investigations of MARS-M performance showed some weakness in the pure queue-based architecture of the address and execution subsystems

in scalar processing. This was due to a lack of general-purpose registers for storing local data of the operators being executed within these subsystems, resulting in the use of the complex dynamic resource allocation mechanism to perform simple scalar operations. To resolve this problem, the local data memory of each subsystem was divided into two parts: one for storing the constants of all the fragments being loaded into the subsystems, and the other for storing operators' local data. After redesigning some mechanisms, it became possible for fragments to take/pass their input/output parameters through the general-purpose part of the local data memories. This implementation significantly improved MARS-M scalar processing performance [Vish84].

The aggregate peak performance of the MARS-M prototype was 18.5 Mflops or 240.5 MIPS, which is the sum of the following components [Doro92]:

- execution subsystem: 18.5 Mflops or 92.5 MIPS
- address subsystem: 111.0 MIPS
- control subsystem: 9.25 MIPS
- control processor: 27.75 MIPS

MARS-M Chronology. Kotov, G. I. Marchuk, and Vishnevskiy made the decision to start implementing some of the MARS ideas in 1980. They felt that the basic ideas had been sufficiently worked out that they could begin designing a concrete architecture. They had gotten enough support from V. S. Burtsev that they felt that such a project was feasible. The GKNT agreed to help finance the project, but demanded a target performance rate of 50 Mflops. Kotov felt this was rather unrealistic and proposed 10 Mflops. As a compromise, a figure of 20 Mflops was agreed upon. The first MARS-M design was developed during 1980-1981. At this time certain technical parameters were established in conjunction with ITMVT. For example, the system would use ECL chips with 3

nsec/gate delays, water cooling, would have a maximum of 100 ICs/per board, would fit into standard El'brus cabinets, etc.

In 1982 a second version of the design was initiated. Here the number of functional units and the clock period were established. At the same time, the logic design of subsystems and the development of a fragment translator and simulator was begun. In 1983, further modifications were made to the instruction sets of the arithmetic and address subsystems, and the clock period had to be increased to 92 nsec. Design of the memory subsystem was completed in 1984. In 1985, the year START was formally created, design work on the MARS-M was completed, and prototype assembly began at the VEM Factory in Penza. In this year the clock period had to be increased to 100 nsec. In 1986, when prototype development at Penza was delayed, the Novosibirsk team developed the concept of a distributed operating system kernel, designed and built a simulation model of the control processor, and began design of the parallel structure machine language (KOKOS). In 1987, further software design was carried out, a debugging subsystem was created, and the clock period once again had to be increased, to 108 nsec. The prototype was finally completed in 1988 and installed at the Novosibirsk Subsidiary of ITMVT. For the next two years, until funding was terminated, it underwent testing [Doro92b, 2].

Construction, and influences on the design. The construction of the MARS-M was heavily influenced by the participation of ITMVT. Through ITMVT, the MARS-M developers gained access to components, subsystems, and CAD systems. ITMVT provided technical advice, logic design and a board layout system. The institute also produced production documentation for the factory. While the development of a MARS-M prototype would not have been possible without such assistance, these elements constrained development. Only by conforming to the technology available at ITMVT could the machine be built, however.

When the decision was made to use El'brus-related parts and systems to build the MARS-M, the intention was to integrate the MARS-M into an El'brus configuration as a special-purpose CPU [Golo86]. By not having to develop the entire I/O system, and being able to use many of the materials and facilities available through ITMVT, the MARS-M developers could save considerable time, money, and effort [Doro92b, 1]. Developers could use the El'brus racks, boards, memory, chips, power supplies, cooling system, and the entire El'brus I/O system. The system incorporated Soviet analogs of the Motorola's MECL 10K chips, the IS-100 series with a minimum delay of 2.5 nsec. These medium scale integration ECL chips were used in the El'brus-2 and Soviet high-end mainframes through the mid-1980s, when ECL gate-arrays were introduced [Anto81; Smol83; Loef83; Kuch85; Lomo87].

The MARS-M had to fit into the three racks constituting a single El'brus CPU. It was not possible to add a fourth rack because the racks were designed to fit in a three-spoke configuration, and the water cooling system and power supply were also designed for three racks, not four. The need to build the MARS-M small enough to fit into the racks constrained the implementation in a number of ways [Doro92b, 1]. First, the number of functional units and processors had to be limited. While the Conception called for multiple execution, address and special-purpose subsystems, the implementation was limited to one CPU with one address and one execution subsystem [Doro92]. Second, the designers were forced to reduce the number of functional units. There was room in the prototype for only one floating-point addition unit, and one floating-point multiply [Doro92]. Third, designers had to decrease the word-length to 48-bits, rather than use the standard 64. Fourth, the number of memory address channels and memory ports had to be reduced from eight each to six.

Main memory and the memory management unit occupied one of the three water-cooled racks. The 2M-word memory used 16-Kbit ECL DRAMs.

The designers also were forced to accomodate the El'brus system by choosing a clock period which could be easily synchronized with that of the El'brus-2. Their original choice, 88 nsec, was, in the early 1980s, exactly twice the planned clock period of the El'brus-2. Timing issues were never fully resolved, however. As they constructed the MARS-M, developers found that they actually needed a clock period of 108 nsec, more than twice the clock period of the El'brus-2, even though the latter ended up with a clock period of 47 nsec. Second, the host El'brus available to them in Novosibirsk was an El'brus-1, housed at the Novosibirsk Subsidiary of ITMVT located in an adjacent building. Not only did this machine have a different clock period, but the connection to the El'brus exceeded the six meters maximum called for in the design specifications. This too threw off the timing.

During the project developers realized that they would need to use conventional sequential languages. They planned to develop FORTRAN and C compilers at a later stage and move from library-oriented to conventional high-level language programming. These compilers would have to build a data flow dependency graph and find independent fragments of a program which could be executed in parallel by address and execution processors communicating with each other. FORTRAN and C compilers were not implemented before the project ended, however.

Relationship to Western developments. The MARS-M was a unique computer which combined an unusually large number of different architectural concepts. As we have seen, at the conceptual level, Kotov and Marchuk began with a survey of Western machines and selectively drew some basic principals from Western and Soviet research. At the implementation level, it is difficult to trace the origins of many MARS-M ideas.

The final design contained a number of features which are similar to some Western efforts, but in many cases the Soviet work pre-dated the Western or at least represented an instance of parallel development.

Perhaps the strongest Western influence on implementation were the horizontal architecture ideas which had been incorporated into the AP-120B attached array processor by Floating-Point Systems, Inc. This machine was first delivered in 1976 [Hock88, 206]. It used instructions which were 64-bits wide, and each instruction controlled the operations of all units in the machine [Hock88, 212]. This arrangement was called 'horizontal microcode.' One instruction could be issued per clock period, but since each instruction controled multiple operations, the aggregate performance was much higher than for a machine with a comparable clock time, but only one functional unit. This approach inspired the use of very-long-instruction-words at the MARS-M functional level, even though actual implementation was unique.

A number of Western projects have characteristics which resemble portions of the MARS-M. The MARS-M library-oriented approach for fragments made it possible to load the code for all fragments into the distributed instruction memory before a program was initiated and fetch it locally and independently using multiple sequencers at run time. The compiler specified a timetable of all instruction streams to be issued by the control modules in executing any fragment. A similar extended VLIW approach, called XIMD, incorporating multiple sequencers and homogeneous functional units was proposed in [Wolf91].

Ideas similar to the MARS-M virtual multiprocessing scheme were implemented in the HEP-computer [Smit78]. Both systems use dynamic sharing of common hardware between multiple processes, performing processor switching when memory access operations are issued.

The ZS-1 and the PIPE computers are examples of a recently developed computers with decoupled architecture [Smit89; Farr91]. The MARS-M differs from these and earlier efforts in its use of multiprocessing to perform both memory access and execution tasks, and dynamic hardware allocation of both address/execution processors and communication queues.

The idea of virtual multiprocessing is not new. For example, the CDC 6600 used virtual multiprocessing for its peripheral processors in 1964. The approaches of the MARS-M and the CDC 6600 differ in key respects, however. Only two of the CDC 6600 virtual processors executed operating system functions; the others were used for communication with input/output devices. Also, the CDC 6600 peripheral processors were switched in equal time periods, in contrast to those of the MARS-M, which are switched dynamically.

Most implementation ideas originated with the Russians, however. Vishnevskiy liked to work in isolation of other work being done in the world, prefering to think up his own ideas. While working within the high-level framework established by Marchuk and Kotov, he made decisions about implementation not from the perspective of the MARS ideology, but from a more pragmatic viewpoint. He knew about the horizontal architecture of the FPS, and adopted this approach, but in general had limited access to foreign developments. He reportedly does not read English very well, so it was difficult to absorb Western ideas. In [Vish85, 79] he attributed some influence to ideas about parallel processing, pipeline processing, and distributed memory conceived by S. A. Lebedev, the Father of Soviet computing.

Dorozhevets, the principal designer of the control processor, was considerably better informed about Western developments. The MARS-M team did not appropriate Western ideas directly, but occasionally a Western development would generate a new idea, or reinforce some design decision already made. The ideas implemented were ultimately the

outcome of a complex interaction between foreign ideas, indigenous ideas, and untraceable trains of thought. Even in retrospect, it is difficult for designers themselves to trace the origins of ideas. Dorozhevets comments [Doro92c, 7]:

> Maybe I had known about distributed operating systems, but I'm not sure now. It's difficult to remember why you proposed this.... I've read so many papers at that time that it was useful for me to see links, the similarities between ours [and foreign developments] and this coincidence sometimes resulted in new ideas how to take the next step. It was a complex of reasons. I knew about horizontal architectures. I knew about all the supercomputers in the USA at that time. But our information is sometimes very basic, not very deep. Sometimes it was not clear. So maybe, to compare our approach...it was not a direct compilation of ideas. That much is clear.

Although it never became a viable machine, the MARS-M was a useful research vehicle. Perhaps overly complex, it nevertheless demonstrated the possibility of integrating multiple architectural approaches such as data flow, decoupled heterogeneous processors, hierarchical systems, and VLIW scheduling. It also explored possibilities for combining static and dynamic scheduling in a VLIW implementation.

6.5.1.2 MARS-T

The MARS-T was intended to be a testbed for experimentation with a variety of concurrent structures, communications methods, and architecture-algorithm relationships. It represented an integration of the MARS philosophy including earlier research on asynchronous processes by Kotov, Narin'yani, and others, and two bodies of Western research: transputer architectures and the associated Communicating Sequential Processes (CSP) computational model of C.A.R. Hoare, and Niklaus Wirth's Modula-2 research and the Lilith Modula-2 processor.

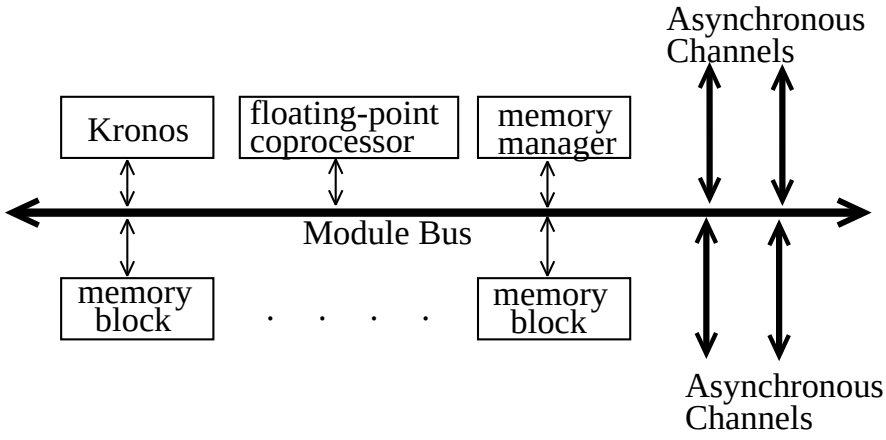


Figure 6-2 MARS-T Basic Module
Source: [Koto91, 38]

The MARS-T contained one or more processing modules, shown in Figure 6-2, each consisting of at least a processor (called the Kronos), a memory block, a memory manager, a floating-point coprocessor, and asynchronous channels. In keeping with the philosophy of modularity and expandability, the MARS-T configuration could be increased either by adding additional processors, additional memory, a memory manager, or specialized processors such as an arithmetic co-processor to any module; or, by adding additional modules. Each module was considered a node, and nodes could be grouped in a variety of configurations such as a regular network or a hierarchy of nodal clusters. By varying the intra- and inter-nodal configurations, the system as a whole could be tailored to specific applications. One example, in this case the MARS-T prototype model, is shown in Figure 6-3.

MARS-T designers chose to base their system on C.A.R. Hoare's CSP model primarily because it is simple [Koto91, 37]. Hoare's contribution was to suggest that input and output are basic primitives of programming, that parallel composition of communicating

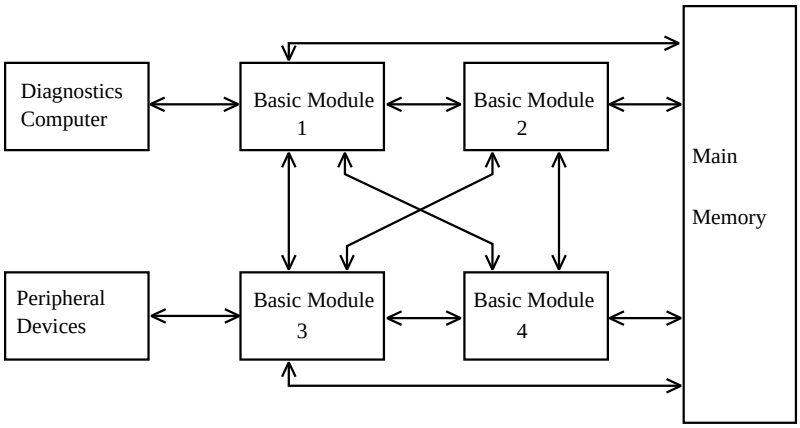


Figure 6-3 MARS-T Logical Structure

sequential processes is a fundamental structuring method, and that such communication can be expressed simply and usefully. CSP is based on three components: (1) Dijkstra's guarded commands which are used as a sequential control structure and are the sole means of introducing and controlling nondeterminism; (2) a parallel command to specify concurrent execution of constituent sequential processes; and (3) simple forms of input and output commands which are used for communication between concurrent processes [Hoar78]. CSP is the computational model underlying the Transputer and its native programming language, OCCAM.

The MARS-T computational model differs from CSP principally in two respects: the channels are asynchronous, being represented as first-in-first-out (FIFO) buffers, and processes are created dynamically with recursive creation allowed [Koto86, 11; Koto91, 37]. Processes can access a queue whenever it is ready to send or receive data. More details on the physical implementation of such queues are given in [Kuzn86, 16-17; Koto91, 38]. Modifications introduced in the implementation to improve efficiency in-

cluded: (1) a prioritization of processes accessing channels; (2) a time-limit on how long a (physical) processor must wait either to take data off of a channel or to put it on; and (3) a protection scheme to prevent unauthorized use of channels. The prioritization of processors accessing a channel does not change the FIFO prioritization of data already *in* a channel [Koto91, 37].

Hoare considered the alternative of making the basic communication process asynchronous. He deliberately rejected this alternative for two reasons: ‘‘(1) It is less realistic to implement in multiple disjoint processors, and (2) when buffering is required on a particular channel, it can readily be specified using the given primitives’’ [Hoar78]. The Novosibirsk designers had different reasoning. The asynchronous channels were more faithful to the overall MARS philosophy, and the data-to-process synchronization being implemented was viewed as simpler and more efficient than the process-to-process synchronization required by the CSP model [Kuzn86, 19].

Hoare also deliberately disallowed recursion. In explaining the rather static nature of the model, he explains that it was intended to be implementable both by a conventional computer with a single main store, and by a fixed network of processors connected by input/output channels [Hoar78]. Simplicity was a key consideration. Considerations of simplicity certainly played a role in the design of the MARS-T, as evidenced by the selection of CSP, but it was not part of the fundamental design philosophy.

The basic processor within a MARS-T is the Kronos. The Kronos originated as a student’s master’s project. Ye. V. Tarasov, D. N. Kuznetsov, and others developed a one-board processor for the DVK[3] computer, finishing at the end of 1983. Aleksandr G. Marchuk, head of the MARS-T project, was not involved with the project at this time.

[3]The DVK family of personal computers has an instruction set based on that of DEC’s PDP series.

In 1983, A. G. Marchuk began efforts to start a multiprocessor project and needed a basic computing element to serve as a processing module. He first considered using the MARS-M control processor, redesigning it on the basis of the Soviet analog of the Am2900 bit-sliced processor, the K1804. This would have required that Dorozhevets leave Vishnevskiy's laboratory to work for Marchuk. Vishnevskiy did not agree, so using the Kronos processor was one of the only options open to Marchuk. At this time there were no other 32-bit processors available in the Soviet Union. Marchuk adopted the Kronos and included the project into START. It grew to become the major START project, consuming more resources than any other.

In developing the Kronos, developers drew many ideas from the Lilith processor developed by Niklaus Wirth at the Swiss Federal Institute of Technology. Like the Lilith, the Kronos is a processor designed for fast and efficient implementation of Modula-2. Three fundamental elements of the Lilith design incorporated into the Kronos were: an instruction set carefully selected to map neatly to Modula-2 constructs; a stack-based architecture which is known to support the implementation of high-level procedural languages rather well [Bulm77]; and an addressing scheme relying, in general, on short addresses.

The Lilith used six instruction formats ranging in length from one byte to three. These were designed so that the most frequently used instructions had the shortest length. The stack-based architecture enabled the Lilith to use relative rather than absolute addressing, greatly conserving the space that would otherwise have been used for addresses [Ohra84]. The Kronos incorporated all these features, except that the Kronos instructions were one-, two-, or four-bytes in length [Kuzn89]. In both processors most of the commands are single-byte.

A number of other low-level design features were shared by the Kronos and the Lilith processors: the use of implicit, immediate, local, global, indirect, and external addressing; an immediate format using a four-bit opcode plus a four-bit constant, a feature previously unique to the Lilith; strict control over the over/underflow of the expression stack by the compiler [Ohra84; Kuzn86]; a limit of 256 local variables within a procedure, although this is a limitation imposed by the compiler rather than the hardware [Kuzn89, 2]; and indirect referencing of the values of structured variables–arrays and records–, so more than 256 variables can be accomodated if they are stored within such structures.

Both processors supported a process-switching command. The Lilith, designed to be a single-user machine, uses this command for co-routines. The Kronos, designed to be incorporated into a multi-process environment, uses the command for multiprocessing [Kuzn89, 3]. Like the Lilith, the Kronos (operating system) supports dynamic binding of separately compiled routines at run-time [Kuzn89, 3].

The Kronos processors were developed entirely using indigenous series production 1802 8n bit-sliced processors, manufactured using Schottkey TTL technology [Schl83; Goly83; Stap88; Kuzn89,3; Kron89]. The Lilith was built using the Am2901 bit-sliced processors, and the Kronoses were built using

The primary differences between the Lilith and the Kronos were that the Kronos was a 32-bit processor, compared with the 16-bit Lilith, and the Kronos incorporated a number of extensions to the instruction set to accomodate the communications necessary in the MARS-T multiprocessor [Koto86]. Besides increasing the size of the operands, the increase in word size permitted an expanded address space. The original Lilith permitted 16-bit addressing; the maximum length of an instruction was three bytes: one for the operand, and two for an address. It had 64 Kbytes of RAM although with a four-bit shift it could access 256K. Later models had double-word addressing, and was built with eight

Mbytes of RAM. The Kronos design permitted instructions of up to four bytes (one word) in one word, providing up to 24 bits of addressing. The amount of main memory included in the processors was, due to space considerations, limited to 512K-2 Mbytes, depending on the size of the RAM chips used [Kuzn89, 3].

The ‘‘signal’’ instruction activating the first of possibly multiple processes waiting for a channel was added to the instruction set to help support inter-process communication. Signals were also used to manage the queue of processes waiting to be executed by a given processor. The addressing scheme of the Kronos was extended to accomodate multiprocessor configurations. The local memories of each processor constituted a global address space accessible by each processor. The global physical memory consisted of a node number and a local memory address. The link controller of each node uses a mapping table with information about the configuration to route data from one link to another [Koto91, 39].

Apart from these primary differences, there were numerous minor differences. For example, in the Lilith the instruction fetch unit performed a prefetch of 64 bits of instructions (four words), while the Kronos permited only 32 bits of instruction (one word) to be read at a time [Ohra84, 190; Kuzn86, 16]. Designed to handle large high-resolution bit-mapped displays, the Lilith incorporated a broad, 64-bit memory read bandwidth, a more conventional 16-bit write bandwidth, and a special shifting device to accomplish the shifting and masking needed for image movement on the screen [Ohra84, 188-190]. The Kronos did not have these features. The expression stack of the Kronos was seven words deep, while that of the Lilith was sixteen [Kuzn89, 2]. These differences reflect different goals for the machine (the Kronos workstation was not designed to be a graphics workstation) and the nature of the component base which limited the amount of functionality which could be placed on a single board.

Modula-2 was selected in part because it is quite amenable to the basic design philosophy of MARS. It is fundamentally modular, allowing programs to be partitioned into units with relatively well defined interfaces. These interfaces, enabling the specification of an object to be separated from its implementation, promoted information hiding (the control of the visibility of objects to other modules), and separate compilation of modules [Wirt84].

The extensibility of MARS-T software was also facilitated by the modularity of Modula-2. Information hiding allowed software building blocks to be constructed, compiled, and executed in a rather independent manner. The module specifications shielded the outside world from the internal details, providing a concise interface between modules. When modules were created and compiled, they could be linked dynamically at runtime. This obviated the need to maintain multiple copies of a module, reducing the size of the executable code, and speeding up the editing-compiling-execution cycle.

Multiple versions of the Kronos were implemented. The first processor was the Kronos 2.2 [Kuzn89, 3]. Designed for installation into the Soviet-made DVK microcomputers, this processor used Digital Equipment Corporation's Q-bus. It used 32-bit operands, but since the bus was 16-bit, two bus accesses were needed to fetch a word of data from memory. To enable the entire processor to fit on a single board, the ALU was only 20-bit. At least two clock cycles were therefore required for each data arithmetic operation. The processor included up to four Mbytes of directly-addressable memory and ran at a frequency of 4 MHz.

The Kronos 2.5 was the first full 32-bit processor, using a 32-bit ALU and interfacing to the Multibus-1 32-bit bus from Intel [Kuzn86, 15; Kuzn89, 3]. Completed around 1985, this two-board processor was installed in the Labtam computer, a 32-bit machine manufactured in Australia which had been sold to numerous Soviet research institutes, in-

cluding the Institute of Cybernetics in Tallinn, and the VTs SO AN SSSR. The Labtams were being used at the Institute of Cybernetics for software development, and the incorporation of the Kronos into a Labtam system in place of the native processor was one means by which the Tallinn group could contribute software to the Kronos.

The Kronos 2.5 was used in the MARS-T prototypes. Configurations with four processors were built, but plans called for eight processors [Kuzn86, 15; Koto91, 39]. In a further departure from the Transputer architecture, the MARS-T configuration included, besides 512K - 2 MBytes of main memory local to each processor, a common memory store using the four-port memory of the PS-3000 multiprocessor developed by NPO Impul's and described in section 7.5.1. The design of a full-function MARS-T system would also incorporate a diagnostics/console computer, a network interface, access to secondary storage, and possibly specialized processors for scientific computation such as the ES-2706 array processor (described in section 7.12.5) or perhaps even the MARS-M. Full configurations were never developed, however.

Early design specifications proposed a clock period of 200 ns (5 MHz) which, given an average 0.75 period instruction access time for a total instruction execution time of 350 nsec, would give an expected performance of 2.86 MIPS [Kuzn86, 16]. Unlike the Kronos 2.2 CPU, the Kronos 2.5 CPU consisted of two boards, however. To accomodate the inter-board propogation times, the clock period had to be increased to 330 nsec (3 MHz), reducing the expected performance to 1.5 MIPS [Kuzn89, 3].

The Kronos 2.6 was a version designed as the engine for a single-processor workstation. The architecture and chips used to implement it were the same as for the Kronos 2.5. The principal differences were that the Kronos 2.6 used a native bus rather than Multibus-1, and was constructed using boards conforming to the European E2 standard

[Kuzn89, 3]. Like the Kronos 2.5, the Kronos 2.6 had a clock frequency of 3 MHz and a peak performance of 1.5 MIPS [Kuzn89, 3; Kron89].

The Kronos 2.6WS was a workstation based on the Kronos 2.6. Another workstation built on the basis of the Kronos was the PIRS object-oriented workstation developed at the Institute of Cybernetics in Tallinn [Kyae86; Koto91, 39]. The PIRS workstation was designed to be an object-oriented system, designed to support the NUT (New UTopist) software development environment also developed at the IK AN ESSR. It was distinguished from other workstations, both Kronos-based and otherwise, by the design of a specialized object server which was linked to the Kronos processor and memory via a systems bus.

The decision to build the PIRS dual-processor system was based on at least two considerations. First, it represented one of the few examples of an implementation of the MARS ideology of incorporating special-purpose processors into a single, multi-purpose system. Second, the developers wished to design a system which would, on the one hand, be able to take advantage of the rapid progress being made in conventional, sequential processor technology, and on the other, provide high performance in a specialized problem domain, in this case the management of objects. The desired flexibility was made possible by using a modular and open architecture [Tamm88, 459-461].

To implement PIRS, the Tallinn group obtained basic Kronos design documentation from Novosibirsk, built the processor, and added a board to implement the object server functions. A prototype unit was completed in March, 1988, shortly before the termination of START.

Kronos design documentation was sent to a number of other organizations which manufactured some units of their own. At least four industrial organizations, the KamAZ Automotive Factory, the Institute of Atomic Energy, and the ELAS research institute in

Zelenograd (under Minelektronprom), and the Scientific Research Institute of the Aviation Industry decided to use the Kronos design to manufacture processors for themselves.

Each organization built its own Kronos, and made minor modifications to the design to suite their own purposes. For the PIRS workstation, the Institute of Cybernetics added an additional board to handle graphics functions [Tamm88]. The ELAS institute changed the instruction set considerably. In short, there was no ‘‘standard’’ Kronos. Marchuk’s group continually improved the processor and among other things adapted it to run with three different bus architectures, the Q-bus, Multibus, and Multibus-II.

Two operating systems were developed to run on the Kronos. Excelsior, supporting a Unix-like interface in a multiprocessor environment, was written in Modula-2 for more efficient execution [Kuzn86, 15; Kuzn89, 3]. Although providing a similar interface to the user, the Unix operating system was also implemented. The primary reason for this was that Tyugu’s team, developing the NUT environment in C rather than in Modula-2, and the developers in Moscow prefered an operating system based on C. Preferences in both camps were strong, and rather than reconciling the differences and settling on one operating system, both were supported.

6.5.2 START Operation

Whatever the earliest intentions may have been for close cooperation, START was realized as a loosely-coupled set of complementary research projects under a unified funding umbrella. Contact between the groups consisted primarily of one conference per year, held at an Academy of Sciences *dacha* outside of Moscow. Each January, representatives from each laboratory met with representatives from the Academy of Sciences, the GKNT, industry, and others for approximately ten days to discuss organizational, financial, and technical issues. Discussions here were intense, but individual groups operated in a much more isolated fashion during the remainder of the year.

Tyugu and Narin'yani had worked together in the past on artificial intelligence problems, but other than the conferences, the main point of overlap between the participating laboratories was to have been the Kronos. Developed in Novosibirsk and scheduled for production in Kiev, the Kronos was slated to serve as the hardware platform both for Tyugu's NUT object-oriented software development environment and Bryabrin's applications software. But even this level of overlap was not fully realized, largely due to delays in manufacturing the processor and the lack of the systems software–specifically C compilers–needed by the Tallinn and Moscow groups. Porting the existing applications to the Kronos would not have been difficult had standard C compilers existed. Although projects to develop such a compiler were initiated in Marchuk's laboratory and in Tallinn, they were never fully completed.

In spite of administrative efforts, the technical characteristics of a project sometimes reinforced the technical isolation between groups. For example, while most technical decisions in the MARS-M project were made within Vishnevskiy's laboratory, decisions about the structure of the system kernel for the control processor involved Kotov and Marchuk. Kotov, Marchuk, and Vishnevskiy felt that a C or Modula-2 compiler should be built for the control processor and that this work could be done outside of Vishnevskiy's laboratory. In this manner, the MARS-M project could be more tightly integrated with other START projects at the Computing Center. Mikhail Dorozhevets, chief designer of the control processor, knew that the hardware and software issues for the machine were too tightly coupled to be developed by different groups. The synchronous scheduling of instructions and other hardware issues were very close to software issues and would have been very difficult to develop separately. Dorozhevets was eventually given a year to develop the system kernel which he did as part of his Ph. D. *(kandidat)* dissertation.

Within the Computing Center of the SO AN SSSR, two over-lapping organizational structures were in effect: the traditional laboratory-based structure and the newer ‘‘working-brigade’’ structure. The traditional structure of Academy research institutes was based on a rigid structure of laboratories. Researchers were assigned to laboratories permanently. Once established, laboratories were seldom changed; once assigned to a laboratory, researchers rarely moved to another.

The working-brigade arrangement, one of the first like it in the Soviet Union, arose out of the need to carry out so many different projects within a limited number of laboratories. There were some projects which required the participation of people in different laboratories, and there were too many projects to be accomodated within the four laboratories involved in START. Some means were needed to improve communication and management, and to accomodate all the necessary projects. The urgency of the START program made it possible to overcome the resistance of the traditional laboratory structure. In effect, all laboratories were made temporary and new brigades were created. Two brigades were involved in MARS-M development, and three were involved in the Kronos. These brigades contained researchers from more than one laboratory. There were no brigades which included individuals from more than one institute, however. When START ended, researchers returned to their original laboratories.

Although the VNTK increased organizational flexibility, it was not completely free of bureaucratic entanglements. In particular, the mechanism provided few incentives for industrial organizations to participate, and many procedures still had to be approved by a number of state agencies. Shortly after START had been formally initiated, Kotov enumerated some of the difficulties remaining [Ntr85]:

> Naturally, everything is not going as smoothly as of now. Experience shows that directives alone are not sufficient for the practical participation of a base organization in the work of the VMNTK.[4] It is evident that certain changes in the economic mechanism are needed to ensure the interest of organizations in the progress and results of the work of the temporary team. Moreover, the system of managing the VMNTK still is not flexible enough: it is difficult to maneuver the staff of specialists, the wage fund, and subcontracting and specialized work. For example, in the course of research, as some problems are solved and others arise, there may be needs for new specialists and for changing the professional structure of the team. But it is difficult to transfer headquarters from one organization to another, because this requires agreement by many parties.

Funding for START was in some respects unconventional as well. Usually, proposals for projects were submitted to a single organization–the GKNT, the Academy of Sciences, a Ministry, etc. In the case of START, several sources of money were combined into a single project, primarily from the GKNT and the Academy of Sciences. Although this added to the complexity of the proposal, the fact that G. I. Marchuk at this time both was chairman of the GKNT and had close ties to the Academy of Sciences counterbalanced the additional complexity. During the three years of its existence, START received on the order of five million rubles per year.

6.6 The Post-START Years (1988-1991)

As planned, prototypes of three types of computers and many software systems were completed by the time START formally ended on April 1, 1988. In accordance with the laws on VNTK, START was disbanded. An interadministrative commission consisting of representatives from the GKNT, the Academy of Sciences, and the State Committee on

[4] VMNTK, elsewhere refered to as VNTK, stands for ‘‘temporary *inter-branch* scientific-technical collective.’’

Computer Technology and Informatics (GKVTI) signed an act for the acceptance of the hardware and software systems, which included the MARS-T parallel processor, the MARS-M scientific processor, and a family of Kronos processors [Manu88; Sots880515]. Public statements by participants and policy makers and press reports had commented favorably about START throughout its existence, and with the stated goals met, praise continued [Manu88; Agam91]. Besides praising the specific research achievements, reports placed considerable significance on the fact that the work had been accomplished in a shorter timeframe than would otherwise have been possible. START's deputy director Yevgeniy P. Kuznetsov commented that ''Over 200 scientists, engineers, and programmers from Novosibirsk, Moscow, Tallinn, Kiev, and Severodonetsk accomplished in three years a research program which under traditional approaches would have required not less than five years'' [Sots880515].

Several factors contributed to START's success. First, the participants were, for the most part, young researchers eager to spend long hours working on interesting projects. The fact that START was unique in its organization and had a high profile and high-level support increased motivation. The promise of a 3,000 ruble bonus (roughly equivalent to ten months' wages) upon completion of the work added a concrete financial stimulus. The nature of START financing further differed slightly from past projects in that START leadership was given more freedom in how to spend the money it received. Another contributing factor was the fact that work had been organized in brigades. Although not necessarily a primary reason for START's success, this organizational form did facilitate communication and management. Under the conventional management structure which includes laboratories, sectors, and divisions, much time is spent acquiring approvals at the different levels for the next stage of the work [Manu88].

While work proceeded faster than it might have in projects with a more conventional organization, one should keep in mind that the START achievements were the product of more than three years of work. The MARS-M had been under active development for five years before START was created. Similarly, Tyugu's PRIZ system and Narin'yani's AI systems had been first prototyped several years before START.

In spite of the stated success of START, the enthusiasm of researchers and policy makers, and the interest in furthering the work expressed by a number of industrial organizations, the transformation of START into a new form was not smooth. The formation of START's successor from 1988 onward was strongly influence by the changing and uncertain environment, the beliefs and strategies of the participants, and the nature of the technologies. The evolution of START's successors is characterized by a measure of casting about–given the opportunities and constraints of *perestroika* and the Soviet economy–for an appropriate structure, domain of activity, and stream of resources to keep the organization and the research alive. In particular, it reflects a tension between commercial and academic activities, between applied and fundamental research.

6.6.1 Organizational Transformation

The post-START developments were characterized by a fragmentation of efforts. A unified successor to START, incorporating industrial enterprises together with the basic START laboratories was never created. Instead, the START constitutents each pursued further development and commercialization of their work independently, experimenting with a number of new organizational forms over time. When START was terminated in April, 1988, the Impul's Scientific Production Association also ended its involvement.

Many factors played a role in these developments. With the end of the START program, new sources of funding had to be found. The GKNT and the Academy of Sciences were not as willing to fund the research past the prototype development stage. Large-

scale funding from a few sources became difficult to obtain. It could be argued that multiple, smaller projects pursuing their own funding would have better chances of success. The organization of a unified program incorporating multiple ministries and administrations would be, for reasons given above, difficult to carry out, especially during times when laws on organizations were changing regularly, causing an overall slow-down in the bureaucratic approval processes. At the same time, laws enabling research institutes to enter into joint ventures with foreign firms, establish cooperatives, and engage in more direct contract work with industrial organizations opened new possibilities for attracting funds. In the balance, the forces pushing towards a smaller, more decentralized arrangement proved stronger than the desires to maintain a unified, centralized program.

Kuznetsov placed considerable emphasis on transfering the results of research into commercial production. While this has always been a stated goal in the Soviet Union, and START had been oriented towards applied research from the beginning, the intensified demands of *khozraschet* made the production of a marketable product a greater necessity [Manu88, 21].

START's termination was followed by the creation of a host of organizations and organizations within organizations which carried out a spectrum of activities, from purely commerical trading, to contract work, to applied and fundamental research financed through state funding: academic institutes, joint ventures, and cooperatives.

6.6.1.1 Industrial START

As START neared its termination, preliminary plans for a continuation of the work where being made by Kotov, Kuznetsov, and others. A primary goal of the plans and strategies was to keep the basic goals of the research, but also to commercialize the results of the START program. Preliminary plans called for using the ‘‘implementation belt’’ of industrial enterprises created around a core of academic and branch science re-

search institutes [Manu88; Sots880515]. Industrial organizations had been involved in some aspects of the research, but only for the purpose of facilitating the construction of prototypes. A natural extension of START was to establish a new, more permanent structure with a much greater industrial component which could take the research results and put them into series production.

Industrial organizations would provide more than facilities. A factor contributing to the success of START was the high motivation of the young researchers who were able to work on interesting and important projects. The conversion of the results of a research project into a commercial product involves a great deal of ‘‘uninteresting’’ work with low scientific content, such as writing device drivers, compilers, editors, etc. A new organizational arrangement was needed which would enlist technicians to carry out the work necessary to prepare a commercial product, and enable researchers to continue to work on ‘‘interesting problems.’’ As Aleksander Marchuk stated, ‘‘We don’t feel it is our job to develop [basic systems software]; our job is the scientific work.’’

In 1988, it was not unreasonable to expect that such a structure could be organized. START enjoyed considerable good-will among policy-makers. A number of Soviet and foreign industrial organizations had expressed an interest in the work [Manu88, 21; Sots880515]. Nevertheless, changes in funding levels, relationships with industry, and the overall legal and economic environments favored alternative arrangements, discussed in the following sections.

6.6.1.2 Joint Ventures

After the law on joint ventures was passed in January, 1987, Kotov’s group worked on establishing the POLSIB Polish-Soviet joint venture between the Siberian Department of the AN SSSR and several Polish electronics industry enterprises [Tryb88; Alek88, 3]. One of the Polish organizations, Metronex, had been selling computer systems in Siberia

since the mid-1970s, so existing business contacts made the Novosibirsk-Poland link a natural one.

The joint venture was formally established on June 21, 1988 [Tryb88b]. The Polish enterprises were to develop monitors, printers, and other peripheral equipment, and the SO AN SSSR would develop software products [Alek88; Tryb88]. Together, the partners were to expand distribution networks within the Soviet Union and Poland, providing not only hardware, but also custom-designed hardware/software systems, installation, training, and support services [Groc88; Tryb88b]. The initial plans called for the commercialization of three projects: the MRAMOR publishing workstation, the Kronos processor, and the Alisa local area network. The MRAMOR workstation facilitates the typesetting of newspaper text by supporting text input, editing, formating, and font and pitch selection [Yers86; Valk87]. The Alisa is a token-based network supporting SM and Elektronika computers [Psas87; Kata88; Kuch90]. At the time of this writing, only the Alisa project has survived, although development and marketing operations for it have been moved to Moscow to bring them closer to the market. Several tens of MRAMOR workstations were manufactured, including some for the newspaper Pravda, but they were unable to compete with the desktop publishing systems imported from the West. The Kronos processors never entered production because the Ministry of the Electronics Industry (Minelektronprom) refused to sell the necessary chips to Poland.

From 1988 onward, Kotov, Tyugu, and Narin'yani made on-going efforts to find partners for joint ventures to refine and market their projects. Kotov visited the United States during August, 1988 with others from Soviet policy making, academic, and industrial bodies involved in computing. They visited numerous companies along Boston's Route 128 and in Silicon Valley [Netm88; Meye88]. On a trip to Silicon Valley in 1988, Gorbachev's chief economic advisor, Abel Aganbegyan, searched for American software

companies which would be willing to establish joint ventures with the Academy of Sciences. The MASTER integrated spreadsheet, wordprocessing, and database package, developed under the START program in Moscow, was among the examples of Soviet expertise which he demonstrated [Sese88]. Neither of these trips produced concrete results, however.

In 1989, A. G. Marchuk and two other Kronos engineers visited individuals in Utah who were also involved in the design of Modula-2 hardware and software systems. Although there was considerable professional overlap between the two groups, collaboration was impossible because of government restrictions. Because the VLSI CAD system used by the Americans had been developed using Department of Defense funding, the Russians were not allowed to see it.

6.6.1.3 Institute of Informatics Systems (ISI)

The organizations formed to carry on START work reflect the desire and need to maintain both fundamental and applied research, further the research already done, and to draw on as many sources of funding as possible. Kotov's efforts were primarily oriented towards the creation of a new institute, ultimately named the Institute of Informatics Systems. Called informally ''The Siberian Program'' or ''START-II'' before it was officially created, the institute was to employ 200-300 individuals, including both START participants and some personnel from an artificial intelligence center in Novosibirsk. Officially a research institute of the USSR (Russian) Academy of Sciences, ISI is involved in both government-sponsored fundamental research and applied research supported through contracts with industry.

ISI was the culmination of a drawn-out, stop-and-go effort which had begun in the late 1970s to create a new institute. The details of this effort are not clear, but a major element was the desire to achieve a greater measure of financial and directional inde-

pendence from the Computer Center of the SO AN SSSR. Serving the entire *Academgorodok,* the Computer Center served a large and varied constituency. The allocation of resources to meet the needs of all was fraught with inequities and political agendas. The laboratories saw a considerable fraction of the funds allocated to them drawn off into the overhead of the Computer Center. Such practices were, apparently, not uncommon nationwide. After Marchuk left to become chairman of the GKNT, a decision was made not to form a new insitute, but to establish something like a special design bureau (SKB) which would operate on *khozraschet.* The drawback of such an arrangement, however, was that the SKB would not be able to carry out fundamental research–very important to Kotov and the other researchers–so no action was taken, and discussions continued. START offered a measure of independence, but it was not permanent.

The design of ISI evolved in response to changing conditions. Although the structure was to be more traditional than the original START, requiring a more typical approval process, early plans already reflected the new economic and organizational opportunities. For example, in August, 1988, Kotov anticipated that ISI would consist of a number of small cooperatives which would help provide maintenance and software development services, as well as more traditional laboratories. The exact form would depend on the interaction between local participants and higher-level organizations. The latter would have to approve the overall structure and participating organizations, while the former would have the freedom to specify which cooperatives would be formed and what their missions would be. A small cooperative, Prolog, was in fact formed, but it proved not to be profitable [Kron89]. It is not clear how many cooperatives were formed by individuals based at ISI or the Computing Center. There were few formal relationships since such linkages were apparently viewed with suspicion by the Presidium of the Siberian Department of the Academy of Sciences.

The bureaucratic process of forming the new organization took a full two years, for a number of reasons. The institute had a more traditional form than START and the approval process could therefore follow a well established path through the bureaucracy. However, because 1987 and 1988 were years of considerable change in the legal framework for organizations, approving bodies such as the GKNT were postponing decisions in anticipation of a change in the regulations and laws, slowing the progress of traditional requests. Once the proposal for a START successor entered the pipeline, it took longer to gain approval than would have been the case under other circumstances. Nevertheless, the approval process both for START and for START's successor was simplified considerably by the fact that no new facilities needed to be built or acquired. In Novosibirsk, Tallinn, and Moscow the participating laboratories remained in their same buildings. When ISI was formed, it remained physically within the Computer Center of the SO AN SSSR on a rental basis.

A proposal to create ISI was initially submitted to the Presidium of the Siberian Department of the AN SSSR, then to the Presidium of the AN SSSR, then to the GKNT which had to approve the funding. Finally, resolutions needed to be issued and signed by the Council of Ministers of the RSFSR and the USSR Council of Ministers, signed by the USSR Prime Minister, Nikolay Ryzhkov. The proposal was submitted to the Presidium of the SO AN SSSR in 1988, and throughout most of 1989 and part of 1990 was slowly pushed through the bureaucracy. ISI was finally created on April 1, 1990. [Izv900606]. At that time, ISI had a budget of 3.65 million rubles, one million of which came from the GKNT for the development of the Kronos processor, plus a few hundred thousand rubles in contracts with industry . While START was funded entirely by the GKNT and the Academy of Sciences, ISI relied much more on industrial contracts. Through 1991, 100%

of ISI's wages from came from Academy allocations, but other budget items were financed through contracts with industry.

The proposal was not submitted until several months after START was terminated. Kotov and others felt that an interim period was necessary during which, they hoped, the changing legal and political environment would stablize and they would have a better understanding of the best way to proceed. As an interim solution, the laboratories involved extensively in the START project were grouped together into a formal division within the Computing Center. In contrast to their pre-START status, the division kept the financial independence gained under START. The new division was included as a separate line item *(''otdel'naya stroka'')* in the Computer Center's budget.

ISI gained not only the ability to determine more easily the direction of research, and control its finances, but also the ability to engage more easily in commercial activities and the freedom to pay employees higher wages than would be possible under state budget funding alone.

ISI was structured partially along traditional lines. Although non-traditional in some of its activities, the structure, and the people filling key posts, still had to be approved by the Presidium of the AN SSSR.

MARS research continued while deliberations to form a new institute were in progress. A sharper division of labor between those working on commerical products and those conducting research arose. The group working on the Kronos processor divided into two portions, one to continue to advance the processor, and one to focus solely on developing a prototype of a commerical Kronos-based workstation. Researchers from the local Special Design Bureau of Computer Technology were enlisted to participate in the hardware development, and some contacts were made with the Parma Scientific Production Association to develop the Unix-based systems software [Kron89]. Thus a loose co-

operation between factories, cooperatives, research institutes and design bureaus was established, even though no formal organization uniting them had been created.

When START was terminated, researchers returned to their original laboratories and the traditional laboratory again became the dominant structure. Some proposals were made again to base the work on a more flexible, task-oriented model, but these met with resistance. Dorozhevets proposed creating new groups to continue the MARS-M development, but this idea was at first not supported by the scientific council because it wasn't clear what the main duties of the heads of the laboratories would be if such groups had the freedom to set their own research agendas. Other unresolved issues were how finances were to be handled, and what kind of contact such groups would have with other organizations.

Attitudes changed when the budget tightened in 1991. Restricted resources forced each laboratory to think harder about how it was going to survive. Dorozhevets had ideas for finding work for about half of the members of the laboratory and was given permission to organize a group as a self-financing entity within the institute. This reduced the pressure on the laboratory's budget considerably. Other groups were formed and flexible, temporary groups soon became the dominant model of the institute. The institute leadership allowed groups to find their own contracts on the condition that they pay the institute a portion of their earnings to cover their use of institute factilities and resources.

The rigid laboratory structure continued to exist as the umbrella under which basic research was carried out. Research institutes within the Academy of Sciences could not be 100% involved in commercial or applied research. The laboratories were responsible for carrying out scientific research. They received money from the Academy of Sciences for carrying out basic research, and were required to give annual reports to the institute and the Academy Presidium about what had been accomplished over the year. The labo-

ratories continued to serve as the organizational construct through which the work was financed, carried out, and accounted for.

At the same time, the science budget was no longer sufficient to support such laboratories. By mutual agreement between institute leaders and laboratories, laboratories at ISI all operated on the principle of *khozraschet.* The goal, of course, was to make each laboratory financially self-sufficient and accountable. This reduced the administrative burden for the institute's leadership, and gave individual laboratories more control over their own finances and a greater share of the money from contracts which they found. The temporary working groups enabled laboratories to supplement their incomes. Out of the money they earned from contracts, they augmented researchers' wages, returned a percentage of their earnings to the laboratories, and gave another percentage to the institute. In the case of the MARS-M groups, researchers were spending 90% of their time on applied research, and 10% on basic research.

In practice this system worked, although a negative side was that the laboratories were more protective of their ideas and work than earlier. Formerly, laboratory members freely shared ideas, experiences and advice with other laboratories. Later, when the correlation between an idea and revenue became much tighter, informal sharing decreased. It also became more difficult to shift people from one laboratory to another because of the questions of who would bear the cost.

Although in principle it was possible for the working groups to include members of different laboratories, as of 1991 there was no cross-over between laboratories.

6.6.1.4 Commerical Start

Besides the creation of ISI, other avenues were pursued with at least partial success. Kotov and others created a strictly commercial company–also called Start–in an effort to promote the Kronos commercially and provide a source of income which could be used

to further work at ISI and pay the workers adequate wages. This organization, independent of ISI although Kotov and others served in the administration of both, was organized as a stock company. It was formed after the START project ended but before ISI was officially formed.

In practice, no actual development work was carried out in commercial Start. Organizationally, it consisted only of Kuznetsov, who became the director, and his book-keeper. Controlled by Kuznetsov, Kotov, and Marchuk, commercial Start became an umbrella under which a variety of commerical activities were carried out. The commercial Start obtained money through bank loans, contracts with the Computing Center (i.e. from the Computing Center's budget), and through other commercial activities.

Individuals from Computing Center laboratories were hired to continue Kronos development. Commerical Start became a mechanism through which the real salaries of researchers could be increased, since for the same work–the Kronos–they were receiving both their wages as employees of an Academy of Sciences research laboratory, and payment for commercial activities through Start. Commercial Start was also to serve as the clearinghouse for Kronos processors, when they finally entered series production. The Kristall factory, part of the Mikroprotsessor Scientific Production Association in Kiev which worked on constructing a Kronos chip set, would sell the processors to commercial Start, which would oversee the manufacture of Kronos workstations and retain the profit from their sale.

6.6.1.5 Other Developments

Narin'yani and Tyugu also created new organizations. Narin'yani created an organization called Intelligent Technologies to carry out the AI and software development research. Initially, Intelligent Technologies was loosely coupled with ISI. Although administratively distinct, Intelligent Technologies did have one department which was actually

a part of ISI. Later, however, this laboratory was moved to Moscow, the location of the rest of Intelligent Technologies, to be closer to the markets.

Additional ideas for keeping not only the START work but also the *Akademgorodok* viable were proposed. Discussions between *Akademgorodok* and Berkeley, California centered on the establishment of sister-city ties. Conversations were also held with representatives from UNESCO. The president of the SO AN SSSR was very interested in developing foreign contacts and Kotov, desiring to be a pioneer in these efforts, proposed a network of "villages" around *Akademgorodok*, linked with electronic networks, which would provide housing and laboratory space for groups of specialists from around the word. The focus was on creating small, flexible groups of people who could work on focused projects for a limited period of time, then disband and be replaced by others. These ideas never passed the discussion stage.

6.6.2 MARS Research

Following the termination of START, MARS-related research continued existing development lines. Although accepted by state commissions in 1988, further testing, refinement, and implementation of the projects had to be carried out.

6.6.2.1 MARS-M

The MARS-M underwent testing and debugging from 1988 to 1990. In 1990, all funding for the MARS-M ended and the approximately 15 individuals who were still working on it had to find other projects to work on. Fundamentally, the reason was a lack of state funds to continue the program. A contributing factor, however, was that the El'brus-2 program, into which the MARS-M had been incorporated, had also come to an end. The MARS-M could not be incorporated into the El'brus-3 program because a pol-

icy decision had been made at ITMVT not to incorporate any specialized processors into the new machine.

The Ministry of the Radio Industry (Minradioprom) expressed an interest in 1990 in the development of a specialized processor incorporating some (but not all) of the MARS-M ideas with a peak performance of 50-60 Mflops. The process of coming to agreement on the project involved multiple iterations of design proposals and corrections. A letter of intent, indicating funding on the order of 10-20 million rubles over five years, had been signed. This project was never initiated, however, because of a lack of funding.

6.6.2.2 MARS-T/Kronos

Considerable time and expense were spent during the post-START years developing the commercial Kronos workstation. The Kristall factory worked on manufacturing industrial quality Kronos processors, and systems and software development work for the machine continued in Novosibirsk.

Kotov had discussions with representatives of factories besides Kristall, including some from the United States, but apart from those with the ELAS Plant in Zelenograd, none produced any results. A Swedish firm trying to set up a joint venture with Minaviaprom (for which ELAS produced chips, in part for the space program) to manufacture embedded control systems was considering using the Kronos as the basic component. They were interested in processors that conformed to international standards, so Kotov was considering augmenting the Kronos to run under multiple modes, one Modula-2 based, and the other Unix based. This would not be difficult since some movement in this direct had already been made to accomodate those with a Unix/C preference within START.

The Swedes withdrew from this venture, and after this, ISI ceased working on Kronos hardware. The primary reason, according to Kotov, was that given the technological level

of Soviet industry, it was not possible to compete in hardware. Work on the conceptual and software components of the Kronos continued, however. The software platforms were ported to Intel-based machines.

6.6.2.3 Technological Base

Besides adequate financing and support from industry, a viable development program must have appropriate tools. To carry out world-class software and hardware development, software engineering environments and computer aided-design (CAD) systems running on workstations with powerful processors and much memory are highly desirable. With the exception of the BESTA workstations assembled at ZIL automobile plant in Moscow based on imported Motorola 68030 microprocessors, such machines were not manufactured in volume in the Soviet Union. The Kronos was too experimental to be used for true development work. CoCom restrictions and high costs made the acquisition of Western workstations prohibitive. The best tools obtainable at ISI were Western personal computers with 386 microprocessors.

From 1990 onward, the Computer Center and ISI lost a number of key people to cooperatives and joint ventures. ISI was able to pay its employees among the highest wages in the Academy of Sciences, but cooperatives and joint ventures were able to pay still more. By the end of 1991, a number of key researchers found positions in the West. Some in Novosibirsk have left ISI for other opportunities within Russia. By mid-1991, ISI had lost roughly 25% of its employees.

6.6.3 Relationships with Industry

The support of industry was critical to the success of applied hardware development. To be effective, support had to consist of both a willingness to participate in the program, and the technological ability to do so. Each hardware project–the Kronos processor and

the MARS-M–established relationships with factories, but the nature of these relationships varied. In both cases the relationships were an on-going source of frustration. In the case of the Kronos processor, the Kristall factory exhibited a willingness to participate (at least at the level of the scientific-production association and the factory), but lacked, or was not able to obtain, the necessary technological capability. In the case of the MARS-M, the VEM factory in Penza had the necessary technological capability, but lacked a strong willingness to participate.

6.6.3.1 Mikroprotsessor Scientific Production Association

While START was in progress, Kronos developers had reasonably good relationships with the Mikroprotsessor Scientific Production Association in Kiev which had been involved in the manufacture of functional duplicates of Intel microprocessors as well as some control computers in the Elektronika family [Bork91; Soko85]. START had been able to contribute considerable funds, on the order of 1-1.5 million rubles per year to Mikroprotsessor in exchange for their participation in the development of the Kronos. The main task at the first stage of the project was to design and manufacture a chip consisting of 30,000 transistors, using 3 micron technology, which was to contain the arithmetic-logic unit and register file for the Kronos. Researchers from Novosibirsk were to supply the logic design, but the implementation was to be worked out solely in Kiev. In principle, the division of labor was clear, but in practice, there was regular interaction between representatives of the Design Bureau at Kristall and Marchuk's laboratory to clarify inconsistencies and work out how the logic was to be separated out into the various processor chips. Relationships were best with the research and prototype development unit of Mikroprotsessor, but the Kristall factory was also interested in the Kronos.

Ultimately, Kristall was unsuccessful in developing the processors. After approximately four years of work, it did produce a version of the chip, but one which reportedly

contained many errors and was not usable. The reason for this was ultimately technical, although higher-level policy decisions were contributing factors.

Kristall had been one of the organizations which, during the early 1980s, had been trying to clone the Intel 8086 microprocessor. For reasons that are not entirely clear, it was not successful. Perhaps the participating engineers were not sufficiently skilled; perhaps there were problems with their development technology. The end result was that Kristall had gained a reputation for being a less than a first-tier chip manufacturing facility. It is likely that for this reason the factory was not as over-loaded with orders for ICs as leading fabrication plants and was eager for additional work, such as developing the Kronos. Kristall may have been given this task because it had development and production capacity available and the Kronos project would not crowd out projects deemed more important.

Bringing in over one million rubles a year, the Kronos project was rather profitable for Kristall. Some speculate that one reason Kristall never produced a fault-free processor was that this would have meant the end of the development project and the associated rubles. START was terminated in 1988 and ISI, its formal successor, was officially formed in 1990. In the intervening years, however, money was still allocated to the computing center to continue START-related work. The annual amount totaled approximately 3.5 million rubles, part of which was money from the GKNT to continue the Kronos work. A new contract was signed between the computing center and Kristall and roughly a million rubles a year went to Kristall in 1988 and 1989. Only in 1991 did investment in Kristall end, with 355,000 rubles contributed in that year. Until that point, Kristall had been eager to continue development work, if not production.

In principle, the *perestroika* reforms, with emphasis on direct relationships between organizations and self-management and cost-accounting could have facilitated the intro-

duction of the Kronos into series production. In practice, these efforts were hindered both by the ministerial structure in Minelektronprom which reformed very slowly as well as the new incentive structures. In spite of the changes in the laws, production decisions in practice still required multiple levels of approval within the ministry. Here the Kronos encountered officials reluctant to alter existing production lines to manufacture a processor for which demand was not clearly large. In general, although the ministries could produce items in small quanitities, there was little to be gained from this. They received little political credit for manufacturing small numbers of items designed by others, yet when production problems arose, they would have to bear the responsibility. At best, the scale of Kronos activities at Mikroprotsessor were modest; only about 10-15 individuals there were working on the project. Although approval was eventually obtained, retooling for the Kronos would not take place until 1990 at the earliest, when the new state orders were scheduled to begin.

Following START's termination, NPO Mikroprotsessor became less interested in the Kronos work for two reasons. First, less money was available from the Novosibirsk group. State funding for START had ended and although Kotov's laboratory was still obtaining resources through contract work and from the Academy of Sciences for research, overall funding was more fragmented and constrained. A more important reason, however, involved the changing nature of doing business at Mikroprotsessor. As institutes and enterprises were given more freedom and responsibility to become financially self-supporting, factories, with real production capacities, became the key economic units in associations such as Mikroprotsessor. While the Kristall factory had been supportive earlier, it increasingly found it more beneficial to manufacture simple components which could be exported for hard-currency than retool for a complex unit like the Kronos.

Speaking about his funding for Mikroprotsessor Kotov remarked, ‘‘It’s just money. It’s not enough.’’

6.6.3.2 Penza Electronic Computing Machines Factory

The MARS-M prototype was build in conjunction with the Penza Electronic Computing Machines Factory (VEM). This factory had participated in the production of a number of Minradioprom computers such as the Ural series, the ES-1052, the ES-1066, parts of the El’brus-2 computers, and others [Glus79; Vino80; Bbc86; Info88]. It was through their ties with ITMVT that the MARS-M developers established contacts with the VEM Factory. It is difficult to determine why, precisely, this factory was chosen and not another, but some suggest that MARS-M development here would be less disruptive to main ITMVT activities than at the principal El’brus factories closer to Moscow.

As in the case of Kristall, the Novosibirsk team developed the logical design, and the design bureau associated with the factory worked out the physical construction. After the logical design of boards was completed, however, the Novosibirsk team used a CAD system running at ITMVT to do routing. Sometimes the logical designs needed to be reworked to make them easier to route. When the routing was finished, the designs were sent to Penza.

Unlike Kristall the Penza VEM Plant usually had more orders than it could fulfill, given its production capacity. The five-year-plans were taut, but over the course of the five-year-plan the plant would receive additional, ‘‘priority’’ orders. The director made decisions about which orders to fulfill first. In the case of the MARS-M, constant pressure from researchers and policy-makers was needed to get the MARS-M prototype built on schedule. Representatives from START in Novosibirsk traveled to Penza monthly for nearly three years.

The visits from Novosibirsk also were to make sure that the system got built in a proper manner. For example, certain resistors known to be unreliable were used in the El'brus-2. MARS-M developers knew that if these ICs were used in the MARS-M, the system would have enormous problems. They themselves established contacts with other suppliers to ship higher quality chips to Penza. Without regular supervision by Novosibirsk researchers, such chips would have been used to relieve bottlenecks in other production schedules, not in the MARS-M.

It was not always clear what type of outside pressure would be most successful in pushing the project through. The original order to begin constructing a MARS-M was signed by the deputy-minister of Minradioprom, N. V. Gorshkov, in 1982, with scheduling revisions signed in 1983. However, little or now action was taken on this order through 1985. In the first half of 1985, G. I. Marchuk, frustrated with the lack of action now that the three-year clock for START was ticking, called local Communist Party leaders who gave the director of the Penza VEM Plant a stern talking-to. Offended, the director dropped the priority of the MARS-M and let the project go idle all together. Work proceeded only at the end of 1985, after Marchuk had talked the Minister of Minradioprom, P. S. Pleshakov, and persuaded him to issue an order.

Through 1989, constant pressure needed to be applied to the Penza VEM Plant to continue work on the MARS-M. In contrast to Kristall, the Penza VEM Plant had the technical capability to carry out the work, but not the incentive. This changed in 1988, after START was terminated, however. Thanks to the introduction of *khozraschet* and decreasing state orders, the Penza VEM Plant did develop excess capacity and it became legally possible for the plant to profit from the production of computers. It became very interested in producing the MARS-M, since the prototype had been completed and the production documentation had been finished. It asked the Novosibirsk developers to help

organize mass production. It was, however, still necessary to find money to invest in re-tooling the production line, and arrange for the needed components to be manufactured and shipped to Penza. Before such arrangements could be made, funding for further development of the MARS-M dried up, in 1990. It is further interesting to note that while the Penza VEM Plant was eager to manufacture the machines, it had little interested in doing the work necessary to find and acquire the necessary chips. This task was left to the developers in Novosibirsk.

6.6.4 Levels of Support

In spite of efforts to establish a viable organization, during 1990 and 1991 in particular, ISI's ability to function eroded. These years witnessed a decrease in research funding, waning support from industry, uncertain markets, the attrition of skilled researchers, and the a growing inability to acquire the tools necessary to carry out state-of-the-art research.

6.6.4.1 State Support

Because the Computer Center and ISI were offically part of the Academy of Sciences, they were recipients of Academy financing, and participated in competitions for Academy and GKNT grants for fundamental research which began in late 1988 [Nemo88; Veli89]. In 1989, the Computer Centers in Novosibirsk and Moscow received many millions of rubles. Funding for research in this year was higher than it had been in previous years, but in the years that followed budgets were cut, and the purchasing power of allocations decreased sharply in the face of high inflation rates and worsening shortages of all kinds of supplies throughout the Soviet Union. In 1990, by the time wages of all personnel had been paid, ISI had only on the order of one million rubles to spend on research. Most of this was paid to the Kristall factory for work on the Kronos. Funding throughout the Academy of Sciences became desparate in 1992. After not supporting Yel'tsin during

the abortive coup in August, 1991, the Academy lost much sympathy among the Russian leaders.

To compensate, ISI sought other resources. The commercial banks could provide a source of temporary funding, although at high interest rates. In 1990, ISI was able to borrow 5 million rubles from a variety of sources to help get the Kronos workstations into production. Technically, ISI assumed 3.5 million rubles of debt; and the commercial Start, 1.5 million rubles.

At the time of this writing, ISI had little hope of remaining viable conducting applied research. By the end of 1991, commercial Start had paid off its portion of the bank loan, 1.5 million rubles. ISI, on the other hand, continued to experience a deep financial crisis, since the interest payments alone on the remaining 3.5 million ruble principle was approaching one million rubles a year. ISI's annual budget decreased significantly in 1991 as well. In 1990 the money allocated by the GKNT for the Kronos had been added to the Academy of Sciences' allocation for scientific research, resulting in more money per capita in ISI than in any other institute in the Siberian Department of the Academy of Sciences. Leaders of the Siberian Department frowned on this practice and forced ISI to give 650,000 rubles to the Computing Center, reducing the budget to three million.

6.6.4.2 Non-State Support

Although ISI had originally been established to continue MARS-related work, other projects which could bring in resources had to be pursued. Maintaining sort viability became the highest priority. As Kotov said, ‘‘The most important thing is to survive.’’

In spite of Academy objection, commercial Start became heavily involved in the computer resale business. The basic budget of ISI in 1991 was three million rubles, but through commercial Start Kotov increased revenues by an additional several million ru-

bles. Through commerical Start Kotov obtained bank loans and dealt in trading computers, stocks, food, etc.

The weak support from NPO Mikroprotsessor was in part a result of a weak market for the Kronos workstations. By the end of 1990, tentative orders for 30-50 units at 100,000 rubles each had been placed. Under the volatile economic conditions of these years, however, the market did not remain stable. Great uncertainty remained about how many organizations would be willing to pay 100,000 rubles for the machine. But a greater concern, according to Kotov, was that development of the workstation would be retarded by unexpected delays. He felt that the current customers were not just buying the workstations for its own sake, but in anticipation of future versions. If new machines were not delivered within a reasonable timeframe, even these customers would discontinue their support. This is what happened. Less than ten units were actually sold, and the remaining units remained in a warehouse in Novosibirsk. In the final analysis, the barriers to successful development of the Kronos hardware were too high for continued work to be realistic.

A major barrier for further production of the MARS-M was the lack of customers. Although the MARS-M was cheaper than an El'brus, it still relied on the El'brus I/O capabilities. The potential market therefore was limited to El'brus users. The actual number of customers was much smaller. The Institute of Catalysis in Novosibirsk expressed an interest in the machine for running simulations of complex chemical reactions, but required that a FORTRAN compiler be developed so they could use existing applications code. Unfortunately, early design decisions about what a fragment was made it very difficult to use programs written in conventional languages without rewriting them to make calls of address and execution fragments explicit. When it became clear that this was too difficult to do in the time requested, the institute withdrew its support. Another institute

from the Ministry of the Atomic Energy Industry expressed an interest in the MARS-M for running simulations of atomic reactors. When this customer found out that the MARS-M was still under development and not ready for such applications, it too lost interest.

6.7 Discussion

MARS research was initiated at a time when emphasis on applied science was increasing with the Academy of Sciences. With the help of powerful patrons, the work obtained high-level attention and support as part of the Soviet answer to the Japanese Fifth Generation Program. It enjoyed many advantages over other Academy high-performance computing projects and was a better than average example of what could be achieved in applied computer development within the Academy.

Although applied projects, the MARS computers (the MARS-M in particular) had a very strong basic research orientation. The MARS-M had a unique architecture incorporating a large number of novel elements. The MARS-T showed a stronger influence of Western work, but was designed as a platform for research in parallel processing. The machines had long development cycles, and neither reached series production.

During the reform period, MARS-related research experienced enormous changes in levels and sources of support, relationships with industry, and opportunities for organizational transformation. The projects and the organizations associated with them provide a good lense through which to view the changes, their impact on research within the Academy of Sciences more generally, and responses within R&D facilities to those changes.

A variety of organizational forms–intra- and inter-organizational temporary working collectives, cooperatives, joint ventures, and commercial enterprises–were experimented with in an effort to find a combination of research orientation and organizational form

which would achieve the principal goal: preservation of the research collectives and the principal research directions. In spite of these efforts, many projects, the hardware development components of the MARS projects in particular, did not survive.

In this section we examine the principal factors influencing the development of the MARS technologies and the structure of the organizations within which they were developed. We conclude with some comments on likely prospects for R&D in the field of high performance computing at the Institute of Informatics Systems.

6.7.1 The Technology

The major factors influencing the development of the MARS-M and MARS-T are presented in table 6-2.

The MARS computers grew out of a desire to explore qualitatively new approaches to building computers. In contrast to other HPC efforts, such as the PS-2x00 machines which were geared toward industrial applications, the principal considerations of the MARS projects were research results–the generation, implementation, and verification of new ideas about machine architecture. In the case of the MARS-M in particular, this research philosophy resulted in a machine which had many interesting, novel, and even exotic features, but which was also very complex, incompatible with existing software, and not well suited for industrial use. The machine made contributions to the field of computer science by demonstrating the possibilities of integrating multiple architectural approaches in a single system, and combining static and dynamic scheduling in a VLIW implementation. It did not have the operational or price/performance qualities necessary to be a successful industrial machine, however.

The MARS Conception, embodying the principles of parallelism, modularity, asynchronicity, extendability, hierarchical organization, architectural integration of hardware

Environment	Technology
Growing allowance for applied research within the Academy of Sciences Existence of powerful patron (Marchuk) **Loss of powerful patrons** Laws on VNTK Three-year limit on VNTK Laws and conventions on structure of Academy research institutes **Laws introducing principles of *khozraschet* in Academy research institutes** Lack of specific industrial sponsors Non-existent market for MARS systems Japanese Fifth Generation Program Weak links with industry **Growing willingness of factories to consider production of (profitable) Academy innovations** **Trends away from funding large-scale hardware development projects**	Nature of El'brus construction technologies **Technological availability** Examples of Western research (including Lilith, transputer, horizontal architectures) ITMVT facilities and El'brus-2 construction technologies (established technology in mid-1980s) Difficulty acquiring necessary components Lack of prior experience in hardware development Inadequate facilities at Kristall factory Availability of 32-bit processors (early work on Kronos)
Organizational structure	**Beliefs (design principles)**
Traditional division/laboratory structure of Academy research institutes Fexible, temporary scientific-technical collectives starting with START **Coexisting structures including traditional laboratories, temporary collectives, cooperatives, joint ventures**	Mission is do make research contributions Explore qualitatively new approaches in a variety of research domains Demonstrate architectural principles through the construction of prototypes MARS Conception: parallelism, modularity, a-synchronicity, extendibility, hierarchical org-anization, integration of hardware and soft-ware, etc.
Organizational slack	**Strategy**
Generous funding under START, and through 1990 **Rapid decrease in state funding, especially for hardware development** **Inadequate income from commercial ventures, contract work**	Establish links with industrial R&D and production facilities Cast MARS in cloak of Japanese 5th Gen. Proj. Use existing technologies for prototype construction **Explore commerical opportunities**

Table 6-2 Factors Influencing MARS Projects

and software, etc. formed the framework for MARS-M development. These principles, with the exception of tagged architecture ideas, remained largely intact throughout the

implementation phase, even though they were not always implemented to the extent originally envisioned.

The MARS Conception played a significant, but weaker role in the design of the MARS-T. The four pillars of MARS–parallelism, modularity, asynchronicity, and extendability–remained in place, but a very different implementation path was chosen. The Kronos was initially developed outside of MARS research. When it was selected as a base component of the MARS-T, however, features such as the inter-processor communications channels were adapted to support the kinds of systems envisioned in the Conception. The goal of the MARS-T project was to develop a research vehicle, not an industrial machine. The MARS-T was designed to be a test-bed with which to explore various concurrent structures, communication methods and mechanisms, and the relationship between architecture and algorithms [Koto91, 37]. A guiding principle was that the MARS-T should provide the basic constructs for a wide range of potential structures. The same system should be able to be configured for coarse-grained parallelism as well as fine-grained; for shared-memory, and distributed memory, etc.

Although the MARS machines were primarily research vehicles, the systems did have to be built, both to satisfy research goals and to attract the needed political, financial and material support. A number of strategic decisions were made to facilitate development. One decision, more political than technical, was to wrap a number of projects into a politically attractive package–as an answer to the highly visible Japanese Fifth Generation Project. This strategy was successful and did help acquire much needed support. It did not, however, introduce any new technical requirements, other than those derived from a desire to improve the overlap between some of the START projects.

Another key strategic decision was to use existing technologies–components, tools, systems–to the degree possible. The MARS-M was built as an add-on to the El'brus sys-

tems and was built using El'brus components. The MARS-T incorporated memory from the PS-3000, standard peripherals, and series production chips. In can be argued that designers had little choice. Although the START program enjoyed considerable high-level support, the commissioning of new components and subsystems would have added new dimensions of complexity and delays that the project could ill afford. There were problems enough getting parts already in production.

Three of the most signficant environmental factors were the support of influencial patrons, the relationship with industrial factories and suppliers, and the lack of industrial sponsors. G. I. Marchuk was able to use his influence as Chairman of the SO AN SSSR and of the GKNT to promote MARS proposals within the various planning and funding agencies. The MARS projects reveal, however, the limits of influence even of high ranking officials, and sheds light on the nature of the gap between the Academy of Sciences and production facilities within the industrial ministries.

The selection of a factory for the MARS-M was based on existing working relationships between ITMVT and Minradioprom factories. ITMVT agreeded to provide development tools and facilities, forcing developers to use the component and technical base of the El'brus. If the factory had not already been using technology, components, and materials geared towards the El'brus line, implementation of the MARS-M would have been *very* problematic. By using ITMVT connections, the task of finding a factory to build the MARS-M was simplified.

The Computing Center of the SO AN SSSR, the Academy of Sciences, and even the GKNT had no direct authority and little influence over operations at the Penza VEM factory and NPO Mikroprotsessor. Marchuk had no ability to select the factories directly; the Ministries offered factories which suited their own purposes. Even after formal agreements had been made and resolutions signed, G. I. Marchuk could only influence a fac-

tory through persuasion. Much depended on the good will of Minradioprom officials, from the minister to the factory director. Steady pressure was required to get factories to fulfill their obligations.

Another characteristic of the environment was the lack of industrial sponsors who would set specific requirements for the machines. The work was financed through the GKNT and the Academy of Sciences with little, if any, industrial co-sponsorship. The basic requirements for the MARS-M were that it run, and have a peak performance rate of 20 Mflops. Within these sparse guidelines, operational characteristics were considerably less important; developers were free to define system characteristics themselves. Although (potential) user requirements came to play a growing role for the MARS-T (or, more precisely, the Kronos processor), this machine also was predominantly a research vehicle. In short, there was little user influence to restrain experimentation with ‘‘interesting,’’ but somewhat impractical architectural ideas. This is not surprising, however, given the emphasis within the Academy of Sciences on producing original research results.

A fundamentally important factor was financial and material support, an important component of organizational slack. The financing provided by the GKNT and the Academy of Sciences was a necessary, although not sufficient, condition for successful execution of the MARS projects. The MARS-M project ended in 1990 because insufficient funds could be found to continue its development. The GKNT and Academy of Sciences were not willing to fund the project past the prototype development stage, and there was no commercial market for the machine. The machine lacked such basic features as compilers for traditional languages (FORTRAN, C) and was, in 1990, not suitable for industrial use. Later developments such as the the willingness of the Penza VEM factory to manufacture the machines could not overcome the fundamental problems of a lack of funding and the lack of a market.

The possibility of developing a commercially viable, profit generating product spurred the development of the Kronos processor which, of all the MARS hardware projects, had the greatest emphasis on industrial use. Originally just the component processor for the MARS-T, the Kronos was spun off as a major development project because Kotov, A. G. Marchuk and others realized the many potential uses of the processor, both for research and for generating funds to support other research. Thanks to generous GKNT financing and large bank loans, Kronos development through 1990 had adequate funding. Its principal barriers to development and manufacture were technological. The inability of Kristall to manufacture a chip in a timely manner meant that by the time some Kronos workstations were manufactured, they were not competitive with comparably priced imported personal computers. Consequently, the termination of state funding for Kronos hardware development, the inability to earn money through sales of Kronos systems, and a lack of prospects for doing so made it all but impossible to continue Kronos hardware development. The commercial efforts coexisted with the more research-oriented work, but competed for resources and conflicted at times with individual researchers' desires to do ''real science.'' This was especially true as the fortunes of the Kronos systems declined.

Technological availability had a profound impact on both the MARS-M and the MARS-T. Three important components of technological opportunity are ideas, know-how, and the technology necessary to construct the prototypes. The MARS ideology was formulated on the basis of a review of existing computer architectures and a few key implementation ideas–horizontal architectures in particular–were absorbed from Western work. In spite of parallels with recent Western work, most of the MARS-M implementation ideas originated in Novosibirsk. The role of Western ideas was considerably greater

in the MARS-T and Kronos projects which drew strongly and directly from work on the Lilith processor and transputers.

The MARS-M and MARS-T were the first high-performance hardware projects undertaken by their developers. Significant time was spent learning how to build computers and this lack of experience and know-how caused delays in the development cycle.

The availability of tools, manufacturing technology, components, and other supplies significantly impacted the construction and design of the prototypes, the length of the development cycle, and, ultimately, the lack of success in the marketplace. Had not the MARS-M developers been able to use technologies and facilities provided through ITMVT, the MARS-M would not have been built. However, the need to accomodate ITMVT practices forced a scaling back of design plans: limiting the number of processors, the word length, the number of functional units, etc. The linking of the MARS-M to an El'brus complex obviated the need to develop I/O capabilities independently. This feature would have seriously limited the market for the MARS-M even if the unit itself had been more amenable to production and use. By the end of the 1980s probably not more than fifty El'brus installations existed.

There were problems with the use of industry production facilities and series production components, however. The difficulty in acquiring the necessary components and the reluctance of the Penza factory to make the system a high priority stretched out development times. The real production time required for the machine would have been only a year if it had been made a top priority. Even if the Penza factory had made MARS-M construction a priority, it is debatable whether it could have been completed in fewer years, given the time amount of time spent hunting for components. Researchers from Novosibirsk visited numerous factories throughout the Soviet Union, spoke with deputy

ministers, and tried to work through military organizations. For want of chips, work on the MARS-M was halted for months at a time.

The nature of the MARS-T implementation was significantly impacted by the Kronos processor which, while not built specifically as part of a parallel system was locally available to developers. Had the MARS-M control processor been used as the basic computational element as A. G. Marchuk originally proposed, the basic software environment, physical size, and interconnect mechanisms would have been very different.

It is not possible to speak of a technological trajectory for the MARS computers alone, since only one generation of MARS-M and MARS-T were built. It is therefore difficult to determine fully which design principles and features were part of some overarching technological paradigm, and which were simply incidental to these machines. It is clear that certain guiding principles were at work. These have been discussed at length above.

Had hardware development continued, which elements of the guiding principles would have remained? We may never know, but there is reason to believe that significant elements would have been rejected, or at least altered. The MARS-M is viewed by some of its engineers as overly complex, incorporating too many widely different architectural ideas. Furthermore, a basic element of the MARS Conception, that the future evolution of computers will be towards systems which can be statically reconfigured to adapt to individual applications, does not appear to be endorsed by world practice. Rather, an emerging trend in HPC appears to be towards multiple, heterogeneous computers linked together via networks in an integrated environment where individual applications run on the most appropriate computer(s). No reconfiguration of individual machines is required.

6.7.2 The Organization

The organizations involved in the research and development of MARS machines underwent regular transformation and reorganization during the period covered in this study. These changes reflected on-going efforts to find organizational forms which were most viable from a financial and technological perspective, the opportunities presented by the legal environment, and the nature of the technology itself. Some of these factors are shown in table 6-3.

One of the most consistent factors running through the transformations is the desire to achieve greater levels of financial independence and viability, to maximize the resources available to the organization. For years prior to the creation of START, Kotov and others had been searching for ways to separate themselves financially from the Computer Center and gain greater control over their own finances. The START program, besides bringing in welcome research funds, provided an opportunity to achieve a measure of financial independence. The formation of ISI in 1990 was largely driven by similar concerns.

The experimentation with joint ventures and commercial ventures reflected an effort to draw in resources from sources which were non-traditional, at least for Academy institutes.

The internal structure of organizations was shaped in part by another aspect of organizational slack: the financial pressures of shrinking budgets and rising inflation of the later reform years. These pressures supported trends towards decentralized control and responsibility of finances which gave both greater control and greater responsibility for finding sources of income to the individual laboratories or research teams. At the inter-organizational level the shrinking research allocations and trend away from large-scale projects also forced a fragmentation of effort. It became easier to find multiple smaller contracts than a few large-scale ones.

Environment Legislation establishing traditional organizational structure of Academy research institutes Administrative process for approving creation of institutes Legislation establishing VNTK **Legislation on cooperatives, joint ventures, other commercially-oriented structures** **Legislation on khozraschet in research institutes**	Technology Number of tasks needed to be carried out under START Nature of the individual tasks Organizational structure Existing laboratory structure Organizational structures under START
Technological availability **Examples of use of cooperatives, joint ventures, etc.**	Beliefs Organization should be structured to maximize organizational slack, efficiency Research is core activity, and organization should be structured to facilitate it
Organizational slack Generous, unified funding under START, Continued funding for MARS-M, -T through 1990 **Rapid decrease in state funding, especially for hardware development** **Inadequate income from commercial ventures, contract work**	Strategy Experiment with new organizational forms **Give laboratories greater responsibility for own viability**

Table 6-3 Factors Influencing Organizational Structure Around MARS Projects

The internal organizational structure was also shaped by the nature of the technological developments, however. One of the chief motivations for adopting the working brigade organization in START was a mismatch between the number of existing laboratories, the number of projects, and the positions of the individuals who were to work on the projects. Although the formal ISI structure remained based on the traditional laboratories, the actual structure depended on the structure of the flexible teams.

Each organizational change (with the exception of the formation of ISI which was along more traditional lines) was enabled by changes in legislation which created opportunities for alternative organizational forms. In particular, the 1983 decree on temporary scientific-technical collectives, and the 1987 decrees on joint ventures and cooperatives,

opened the door to organizational experimentation. The 1987 decree on the transition of scientific research institutes to *khozraschet* made decentralization of financial responsibilities possible.

It is difficult to identify a premeditated organizational transformation strategy. The most consistent strategy to emerge over the years was one of paying close attention to the organizational opportunities available, and applying them to the greatest advantage.

6.7.3 Prospects

The short-term prospects for applied research in hardware develop are very bleak. Both the MARS-M and the MARS-T/Kronos have ceased to exist as hardware projects. Without the appropriate tools, component base, devices, talent, and funding, the institute cannot carry out world-class applied research. There are two alternatives: do commercial work with low research content and scientific interest, or concentrate on theoretical developments which do not rely as heavily on the availability of tools, components, and other material resources. Kotov's heart lies in research, so his desire is to scale back ISI operations to a core of talented individuals who can focus on theoretical development.

Doing research remains a strong desire of those at ISI, but survival remains a dominant goal. Small-scale contractual work, with or without significant scientific content, will remain a major activity until levels of funding for basic research rise again.

What are the implications of recent developments at ISI on its ability to conduct applied HPC research in the future? The prospects for improved idea generation are mixed. On the one hand, researchers have, in principle, greater access to ideas from the West. Thanks to electronic mail connections, researchers can interact with Western colleagues

in a fast, reliable manner.[5] Researchers have been able to travel and work in the West where they have gained considerable exposure to Western developments and practices. Thanks to the emphasis on scientific institutes' ''paying their own way,'' researchers have increased interaction with industrial customers. The degree to which this contact advances science is variable, however.

On the other hand, several developments serve to reduce idea generation. The transition to *khozraschet* at the laboratory level has made laboratories more protective of their ideas, especially those that could have some commercial value. The present financial crisis and the accompanying lack of hard currency make it difficult to subscribe to Western publications and to travel to Western conferences where much exchange of ideas occurs. In the balance, the short-term prospects for the idea generation are quite poor, although the long-term prospects, assuming the basic state of the Russian economy improves, are considerably brighter.

Recent developments have eroded the political base of support for research at ISI. Marchuk gave up his post of GKNT chairman in 1986 to become the President of the Soviet Academy of Sciences. In the Academy elections in December, 1991, held in the aftermath of the attempted coup and breakup of the Soviet Union, he stepped down and was replaced by Academician Yuriy S. Osipov. The Academy of Sciences as a whole has lost much influence in Moscow because of its non-support of Yel'tsin during the August, 1991 coup attempt.

Recent developments have led to a fragmented pattern of resource allocation. Basic funding for fundamental research continues to be supplied by the Academy of Sciences, although at a level which is hardly sufficient to support ISI scientists. Additional income

[5] For example, [Doro92] was written nearly exclusively via e-mail between Novosibirsk and Tucson, Arizona.

is earned through contract work, and this has been augmented with some income through commercial dealings coordinated by the commercial Start. In principle, these mechanisms–particularly the centralized state funding–could supply the financing necessary for high-performance computing projects. In practice, the short-term prospects are very poor, given the catastrophic state of government finances and the decline in allocations for basic and applied research.

Recent developments have had both positive and negative consequences for the ability to prototype systems. On the positive side, the ability to form flexible research groups which are oriented around a specific project and which can draw in the scientists best suited to the task help shorten development cycles and focus research energies in key areas. Also, it has become possible to acquire Western technology which previously was accessible only through a long and uncertain acquisition process through the Academy of Science, foreign trade organizations, KGB, etc. CoCom restrictions have been raised considerably and much technology is freely available in Russia, for a price. On the negative side, financially strapped institutes despair of being able to acquire the basic technological inputs for research: CAD systems, high-quality components, subsystems, and instrumentation, etc. Good research only partially compensates for poor equipment and inputs. Under very difficult economic conditions, scientists are forced to spent extraordinary amounts of time earning money on the side doing tasks with little scientific content, searching for basic material necessities, caring for their families, etc. Development life cycles are drawn out considerably as a result.

There have also been positive and negative changes in the ability of the Academy institutes to move innovations into production. As demonstrated by the changing relationships between ISI and the factories involved with MARS development, the factories are increasingly willing to consider taking on the production which will use their idle capac-

ity. On the other hand, they are unwilling to tool for the manufacture of products which enjoy only a small market. An additional negative factor, observed in the case of the MARS-M, has been the unwillingness of the factory to invest the resources necessary to bring a prototype machine into series production.

CHAPTER 7. OTHER SOVIET HIGH-PERFORMANCE COMPUTING PROJECTS

7.1 Introduction

In this chapter we discuss at greater length HPC projects introduced in chapter 3 which were not part of the case studies of the preceeding chapters. We present additional detail of the technical characteristics of the machines, identifying major design influences and principles, give basic chronologies, and discuss factors which influenced their development and ultimate fate. The next chapter, chapter 8, will consist of a cross-cutting discussion of these systems and those of the preceeding chapters, and the organizations in which they were developed.

7.2 Modular Pipeline Processor (MKP)

The Modular Pipeline Processor (MKP) is, besides the El'brus series, the second major high-performance computing project in progress at the Institute of Precision Mechanics and Computer Technology (ITMVT). The project is headed by A. A. Sokolov, who headed the AS-6 project and was directly involved in BESM-6 development [Ryab91b]. Like other Soviet projects funded in the mid-late 1980s, the MKP sought to break the one Gflops threshold. In contrast to the El'brus-3, the MKP was designed for a broad set users who would not necessarily be able or willing to pay very large amounts of money for the fastest system available [Supe91, 16]. The design goals also included developing a general-purpose system with a full complement of systems software in a manner which would reduce the cost of hardware and maintenance. The system is called ''modular'' because it was designed to be attached to a number of different machines, in a number of different configurations, depending on the needs of the user. G. G. Ryabov felt very strongly that the system should be designed for use in universities and scientific centers

with a wide array of applications; here the chances for building a broad user base were the greatest. These considerations resulted in the following design goals and principles [Ryab91b]:

- achieve qualitatively new level of performance in domestic supercomputers (billions of floating-point operations per second);
- design hardware and software which is reliable, easy to use, and easy to maintain;
- follow international trends in the development of information technologies (UNIX, networks standards);
- involve scientific centers and universities in the initial stage of development in the creation of systems software and an algorithmic base.

The following brief description of the MKP is taken from [Byak90; Byak91].

7.2.1 MKP Architecture

Logically, the MKP consists of a number of functional blocks shown in figure 7-1. The two scalar blocks do preliminary processing on a stream of program instructions. This includes instruction decoding, address calculation, control command formulation, and interrupt processing. Instructions are then sent to other MKP blocks for final execution. The scalar blocks have independent scalar processing resources, but treat the vector and vector sorting blocks as a shared resource. Synchronization of the two streams is done explictly using semaphores.

The vector processing block manages the execution of vector arithmetic-logic operations. It does not perform the computation itself, but sets up the computation by managing the distribution of vector resources, reading and writing vectors to memory, and link-

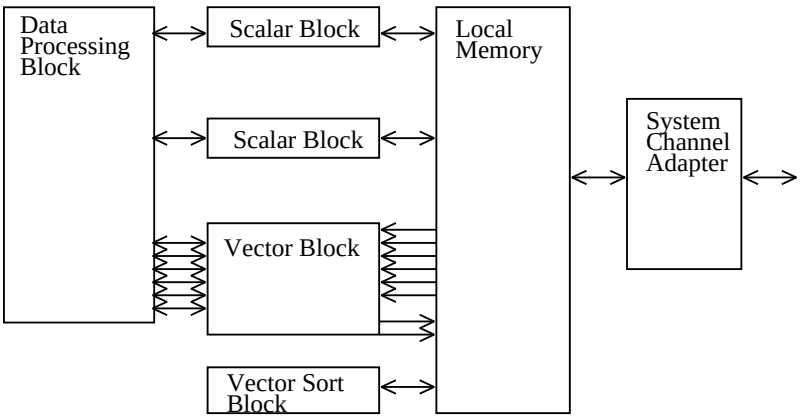


Figure 7-1 MKP Structure
Source: [Byak90; Byak91]

ing individual vector instructions. The vector sorting block provides a variety of sort options on vectors.

The scalar and vector blocks rely on the data processing block for the execution of real and whole-number arithmetic, logic operations, shifts, and a number of other operations on scalar and vector elements. Unlike traditional vector-pipelined (Cray-like) computers which have separate scalar and vector units with independent functional units, the MKP data processing block contains a number of pipelined functional units which can be used by both the vector and scalar blocks. This approach requires a more complex functional unit design, but considerably less hardware. In the standard MKP construction, there are eight pipelined functional units: two each of addition, multiplication, division, and logic. The precise mix was determined through an analysis of the applications of the primary customers (such as the Scientific Research Institute of Experimental Physics in Arzamas-16, a principal nuclear weapons design facility [Gigl930122, 2]), but the issue

was not a pressing one. In principle, the MKP can be built with a different mix of functional units to customize it to the needs of specific customers.

The two scalar blocks make it possible to execute two streams of instructions simultaneously. As they and the vector block set up the execution of individual instructions for the data processing block, the latter might simultaneously be executing vector and scalar operations from the same or different instruction streams.

In vector processing, the MKP allows for up to five vector operations to be chained together so that the results of some operations are used directly as arguments for others without an intermediate write to memory.

Scalar and vector operations are combined by buffering vector operators when they are passed to the data processing block. Scalar operations can be inserted arbitrarily between vector operators without upsetting the pipeline.

Table 7-1 lists the constituent parts of the MKP. Table 7-2 lists the operational parameters.

The earliest design work on the MKP began during the mid 1970s shortly after A. A. Sokolov's team finished work on the AS-6. The design of the machine was strongly influenced by the available component and construction technologies, the traditions of A. A. Sokolov's group, and the vector-pipeline concepts popularized by Cray Research, Inc.In keeping with ITMVT tradition, engineers sought to implement some of the basic pipeline ideas in a manner which was more appropriate to local conditions and design goals rather than try to duplicate the Cray architecture. According to one engineer [Li91]:

> We always wanted to adapt the new ideas which we encounter to our capabilities. On the one hand, we set before ourselves the task of [building a machine with a peak performance of] billion operations per second. On the other hand, we ... don't want to place too high requirements [on the construction]. Therefore, we are obligated, when taking

Data Processing Block 8 pipelined functional units operating on floating-point operands, whole numbers, bit sets, logical values
Scalar Block 4 64-bit adders 128 64-bit general-purpose registers 16 37-bit address-index registers 32 64-bit display registers 1 address functional unit operating on address pointers, indexes, descriptors, and semaphores
Vector Block 1 operator with 5 operations 2 operators with 6 operations 2 packed read vectors from the scalar block 2 vectors for reads to local memory
Vector Sort Block Simple reference packing/unpacking of vectors under direction of logic conditions scale and displacement vectors
Local Memory 2M or 8M 64-bit words 32 banks 12 ports
System Channel Adapter 32 parallel read/write operations

Table 7-1 MKP Functional Blocks
Source:[Byak90, 9]

into account Western ideas, to find a means of implementing them with lower technological levels. We feel that such possibilities open up when we, on the one hand, incorporate the spirit of the new ideas, and on the other hand, try to draw from the most important, and not simply familiarize ourselves with the [Western] implementation and try to copy it. We try to find the realization of those ideas which we like in such a way that it's agreeable to us.... It appears to us that simply copying without having continual access

Word length	64 bits
Clock period	10 nsec
Local memory	
Size	16-64 Mbytes
Total throughput	8 words/cycle (6.4 Gbytes/s)
System adapter	
Total throughput	800 Mbytes/s (400 Mbytes/s in each direction)
Theoretical peak performance on fl. pt. operands	600 Mflops (one result per clock period in each of 6 fl. pt. pipelines

Table 7-2 MKP Performance Characteristics
Source: [Byak90; Byak91]

> to the literature and documentation, not having the ability to use the same components, that this is a dead-end approach. It absolutely leads to a situation in which we continually lag behind....

Designers sought to implement a pipelining scheme which could be supported by the technology available, provide higher performance than a Cray with a comparable clock period,[1] and at the same time use the hardware as economically as possible to reduce the cost of implementation. The clock period and the number of functional units were mostly determined by the number necessary to achieve the goal of one Gflops, given the component technology projected to be available when the MKP design was completed [Ryab90d, 1].[2] The chip and printed-circuit board technologies also placed an upper bound on the number of functional units and amount of local memory which the MKP

[1] With a clock period of 9.5 nsec, the Cray X-MP has a peak performance of 210 Mflops per processor.
[2] The 1 Gflops threshold is actually achieved only in a configuration with two or more MKP.

could contain [Ryab90d, 3]. The final design discussed above reflects the tradeoff between these design goals.

Many of the architectural ideas for the interrupt system, monitoring of the system state, etc. reportedly grew out of the AS-6 project, although so little has been written about this machine that it is difficult to make a comparison.

7.2.2 El'brus-3-1

The MKP consititutes only the computational component of a full system. Through the system adapter, the MKP is attached to a channel system which provides connections to:

- shared main memory modules
- other MKP
- a variety of front-end systems, such as ES mainframes, El'brus-2, BESTA workstations, El'brus-B, etc.
- Peripheral devices

Such a complex, shown in figure 7-2, is called an El'brus-3-1. By allowing a variety of systems to function as an MKP front-end, ITMVT decision-makers hoped to provide a migration path to multiple classes of users: the ES mainframe, BESM-6, and El'brus-2 camps. The MKP is software compatible with none of these machines for a number of reasons. First, no reasonable system could be compatible with all three classes of machines. Second, the user community for the previous machine developed by Sokolov, the AS-6, was small. Third, and most importantly, the MKP was developed under the ideology that machines should be built to achieve the highest performance possible, even if that meant that they would not be compatible with other systems. The one Gflops threshold could not be achieved with an architecture compatible with the machines most widely

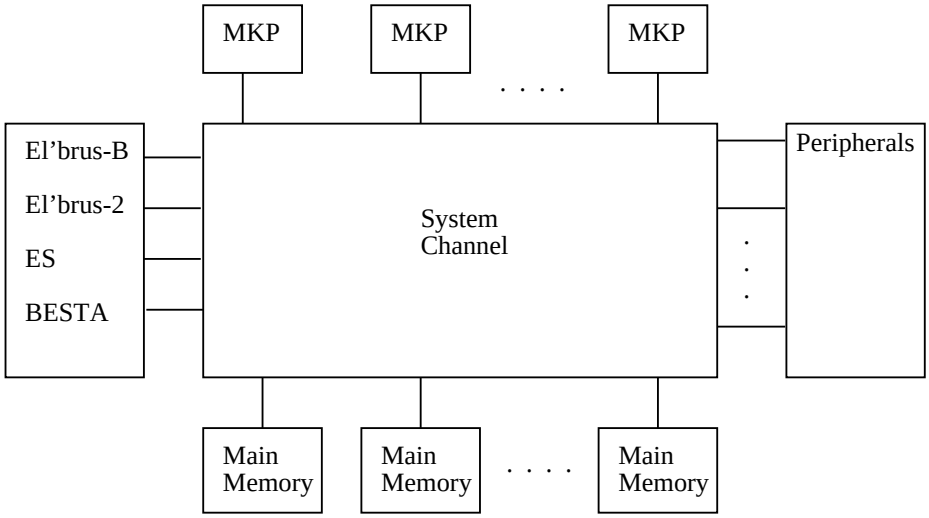


Figure 7-2 El'brus-3-1 Structure
Source: [Byak90; Byak91]

used in the early 1980s, the ES and BESM-6. Theoretically, compilers for El'-76 could be built to port El'brus applications to the new machine.

The El'brus-3-1 reflects the multi-machine philosophy that had developed during the construction of the AS-6. That machine can be thought of as a cluster of interacting systems connected by a network. The communication channel network of the El'brus-3-1 makes it possible to join a variety of different systems into a single complex.

7.2.3 MKP Development

The earliest design work on the MKP began not long after Sokolov's team finished working on the AS-6, in 1976. Like the BESM-6 and AS-6, the MKP was developed together with the Moscow SAM Plant. Prototype construction did not begin until the mid-1980s. Before then, work on ECL gate arrays with 1500 gates per chip had not progressed far enough. Also, this project was given significantly greater support from the di-

rectorate of ITMVT after G. G. Ryabov succeeded V. S. Burtsev as director in 1984 [Pent93c]. The first mock-up was built by 1988 and by December, 1990, the prototype was ready for the first test software routines [Alek91]. By the end of 1991, a total of four MKP had been built. The system passed state testing around the end of 1992.

The MKP uses much of the same technology–ICs, PCBs, CAD, cooling, etc.–that are used in the El'brus-3. Naturally, ITMVT could save considerable time, effort, and money by using the same technology in multiple projects [Supe91, 17].

Designing the functional units for both scalar and vector operations did add to the complexity of the hardware, but designers felt that the reduction in hardware and cost which this permitted made it a good tradeoff. Even so, the logic of the MKP is less complex than that in the El'brus machines. For example, the MKP does not use hardware tags, relying instead on descriptors, reducing the complexity of hardware control considerably. The use of less complex logic reportedly improved reliability [Supe91, 17].

In spite of efforts to design the MKP for use by a broad user base, and very active promotion by ITMVT, the market for the machine remained virtually non-existent through the beginning of 1993. At the October, 1991 Conference ''Problems of the Development of High-Performance Computing Systems,'' Ryabov and others made strong efforts to find new customers for the machine. Although less expensive than an El'brus, the MKP prototypes were not cheap; they cost on the order of 10-15 million rubles in 1991 [Supe91, 18].[3] During 1992, no MKP units were manufactured; the total remained at four.

[3] At this time, the ruble was valued at 20./$.

7.3 Elektronika SSBIS (‘‘Red Cray’’)

In 1978 V. A. Mel’nikov left ITMVT to work at the Delta Scientific Production Association within the Ministry of the Electronics Industry (Minelektronprom) and took with him a number of members of the team that had developed the BESM-6 and the AS-6. In 1980 he began a project to build a vector-pipeline supercomputer based on the principles popularized by the Cray computers only a few years earlier. Mel’nikov continued many of Lebedev’s traditions, particularly that of trying to build the fastest machine possible on the ‘‘available’’ component base. The Elektronika SSBIS is noteworthy for a number of reasons. First, it is perhaps the only HPC system developed within Minelektronprom, a ministry with considerable experience in the development of microelectronics, but whose systems development was largely limited to mini- and personal computers. Second, to a greater degree than other Soviet HPC systems, it was originally patterned after Western computers, the Cray family. In this regard, Mel’nikov departed from the traditions of Lebedev. Engineers did implement a number of non-Cray features, particularly with regard to memory systems, however. Like other Soviet projects which stretched the limits of indigenous technologies, it had a development cycle of over ten years.

Ironically, the decision to develop a system with a Cray-like architecture rather than an indigenous one grew out of a desire to build a machine quickly. Leaders of the military-industrial complex and prominent politicians had become quite worried by the introduction of the Cray-1 into series production in 1976 [Sher92b]. The vector-pipeline architecture had been proven sound and effective by Cray Research Inc. and Control Data Corporation, and policy makers felt that by minimizing architectural innovation, the development cycle could be shortened considerably [Gigl921217, 1]. The El’brus computers would not be ready for use for a number of years, and in the early 1980s Western

export controls made it virtually impossible to acquire systems like the Cray, even if military users had wanted to run their code on a foreign system.

The Elektronika SSBIS was a high profile, high priority project. In 1984 the Delta Scientific Production Association became affiliated with the Institute of Cybernetics Problems (IPK), a new organization created as part of the Academy of Sciences Department of Informatics, Computer Technology, and Automation (OIVTA). This Department, headed by Academy Vice-President Ye. P. Velikhov, was designed to provide a forum within the Academy for coordinated computer policy-making. The development of a supercomputer was one of its main goals [Mikh84; Veli85]. V. A. Mel'nikov was named director of the IPK. Although it was subordinate to the Academy of Sciences, the IPK retained close contacts with Minelektronprom.

The system consists of 1-2 subsystems sharing a solid-state storage subsystem and peripheral storage. The basic system parameters are shown in table 7-3. The performance of 250 Mflops is achieved by generating one result in each vector functional unit per ‘‘synchronization period’’ (two clock periods) [Meln91].

The Elektronika SSBIS uses workstations or minicomputers as a front end and supports standard international network protocols (TCP/IP, Ethernet) and programming languages (FORTRAN 77, Pascal, C) [Elek91, 4-5]. The system is constructed using ECL gate arrays with 256 gates per chip. The 16-layer printed circuit boards, by some accounts of rather good quality, were designed and manufactured at IPK.

The Elektronika SSBIS is nearly compatible with the Cray at the level of assembler, but is not binary compatible. According to one individual intimately involved in development, developers had little information besides published descriptions of the Cray [Sher92b]. The memory system differs considerably from Cray's. Drawing on some ideas from the AS-6, developers designed an ‘‘intelligent memory’’ system. The solid-state

Number of CPUs	1-2	**CPU speed**	
		CPU clock period	13.5 nsec
		vector performance	250 Mflops
		scalar performance	75 MIPS
Registers in each CPU		**CPU Technology**	ECL
address (24 bit)	8		
intermediate address (25 bit)	64		
scalar (64-bit)	8		
intermediate scalar (24-bit)	64		
vector registers (64x64)	8		
Instruction cache		**CPU Main memory**	
capacity	2 Kbytes	word size (data + error	
number of instruction buffers	16	correction bits)	90 bits
		capacity	8-32 Mbytes
		number of banks	16
		bank cycle time	54 nsec
		max. transfer rate	300 Mwords/sec
Number of CPU pipelines		**Shared solid state storage**	
address	2	word size (data + error	
scalar	7	correction bites)	90 bits
vector	7	capacity	256 Mbytes
		number of banks	16
		bank cycle time	640 nsec
		max. transfer rate	150 Mbytes/s
		technology	CMOS/TTL
Word size		**Disk Storage Subsystem**	
scalar and vector operands	64 bits	disks per subsystem	1-8
address operands	24 bits	number of subsystems	2-8
		max. capacity	
		(317.5 Mbyte disks)	20 Gbytes

Table 7-3 Elektronika SSBIS Parameters
Source: [Elek91; Meln91]

memory subsystem has its own processor and instruction set which makes it more than a buffer between main and peripheral storage. It can perform complex memory retrieve functions (such as selecting the diagonal of a matrix) and send only the salient data to main memory. This increases the effective speed of the channels [Supe91c, 7].

The software is indigenous. Porting Cray code to the Elektronika SSBIS was never an issue: the hardware was not binary compatible, and system software developers had little desire to copy others' work. By developing their own systems software, they could improve the quality and maintainability of their code in future years.

Research on the Elektronika SSBIS began in 1980. Gate arrays designed for this machine with 256 gates/chip became available to developers in 1982-1983. Early specifications were ready by 1982 and by 1985 documentation for the prototype was completed and turned over to the factory to begin prototype construction [Gigl921217, 1]. The first prototypes neared completion in 1989 and underwent state testing in that year. By the end of 1991, approximately five prototypes had been constructed at the Kvarts Plant in Kaliningrad (Köenigsberg) on the Baltic Sea [Sher92, 9; Sher92b].

Although the machine incorporated a proven architecture, many of the same systemic factors which delayed development of the El'brus computers affected the Elektronika SSBIS. Having close contact with Minelektronprom helped developers get the necessary ECL gate arrays. Like the El'brus computers, the Elektronika SSBIS helped drive the development of many technologies. The monopolistic nature of the ministry and taut planning created a reluctance by factories to assimilate production of new technologies.

Unlike the Ministry of the Radio Industry (Minradioprom), Minelektronprom did not have a well developed infrastructure for developing large-scale systems. Not only did new design, construction, and manufacturing technology have to be developed, but the inter-organizational contacts between IPK and participating factories and institutes had to be established. During the 1980s these ties were not horizontal, but ran through the ministerial hierarchy. Establishing this infrastructure took an enormous amount of time and was a major cause of delay.

By the end of 1992, two Elektronika SSBIS had been installed at user locations. The project had been sufficiently near completion that financing for development continued at least through the end of 1992 [Gigl921217, 1]. Volume production of the systems is unlikely. There are few users who can afford to pay millions of rubles for a computer which is still not mature. Four more systems, already constructed, stand idle, without customers to purchase them. Like the El'brus systems, the Elektronika SSBIS depends on products from plants throughout the former Soviet Union. Disruptions in supplies can hinder production. For example, connectors were manufactured in Armenia, but production in that state has come to a virtual standstill. The Kaliningrad plant was able to manufacture Elektronika SSBIS prototypes only because it had stockpiled connectors in advance [Gigl921217, 3].

Engineers at NPO Delta have designed a multiprocessor configuration with up to 16 processors, but prospects for funding and the development of the necessary component base are dim. A small-scale entry-level system is also being designed [Gigl921217, 4].

7.4 ES-1191

Until the mid-1980s, the Scientific Research Center for Electronic Computer Technology (NITsEVT) was involved in HPC only as a facilitator, providing general-purpose mainframe hosts for attached array processors (AAP) and other specialized processors. Around 1985, NITsEVT engineers began work on a general-purpose supercomputer containing vector-pipeline processors which were integrated into the system architecture from the outset, rather than added on, as AAPs. The design ideas of what was later called the ES-1191 were presented at conferences in 1986 [Lomo86; Valk87]. The basic approach of designing general-purpose machines with vector processors was adopted by IBM in the 3090 VF (Vector Facility) series, announced in 1985, but there are significant differences in the architectures of the 3090 VF and ES-1191 [Dpro86].

The ES-1191 grew out of the desire of many engineers and policy makers during the early-mid 1980s to reach the one Gflops threshold on 64-bit data. Three additional development goals were: 1) to design a system for which the nominal processing rate would be close to the peak performance; 2) to incorporate programming methods already in use on the general-purpose mainframes, preserving software compatibility and the investment in systems software; 3) to separate as clearly as possible the processing functions from the control functions in order to free up the high-powered computational units from routine tasks which don't require high-speed computing [Lomo86, 60-61; Lomo91].

Recognizing that the key to high nominal performance in a vector processor is high scalar performance [Loma86, 61], NITsEVT engineers designed a system with a conventional ES mainframe used as a control system, plus a computational subsystem consisting of four scalar processors, a vector processor, a memory management device, a monitor processor, and expanded main memory [Loma86; Lomo91]. The scalar processors treat the vector processor as a shared resource. The basic scalar portion, with a performance of 15-60 MIPS depending on the number of scalar processors, is called an ES-1181. A configuration consisting of the ES-1181 and a vector processor is called an ES-1191. The control system is responsible for I/O, program compilation, data exchange between the computational subsystem and the control processor, etc. [Loma86, 64].

The vector processor contains sixteen pipelines; there are four each of addition, multiplication, division, and fixed-point logic [Loma86, 67]. An additional pipeline handles reads/writes from/to memory. The theoretical peak performance, 1.07 Gflops, is achieved when the sixteen pipelines each generate one result per 15 nsec clock period. The vector processor contains a reconfigurable register file with a total of 16K 64-bit elements. These can be organized as sixteen files of 1024 elements, 256 register files with 64 elements, etc. [Loma86, 68].

The ES-1191 is one of three Soviet systems (the others are the PS-2100 and Elektronika SSBIS) which incorporate solid-state external storage as an intermediate stage between main and peripheral storage. The 256 Mbytes of main memory are augmented by up to two Gbytes of so-called ‘‘expanded RAM’’ with a transmission rate close to one Gbyte/s [Lomo91].

Much of the systems software development was carried out at the Institute of Applied Mathematics in Moscow. This includes the development of parallel programming languages and an assembler-level language oriented towards the parallel system [Yush91].

The ES-1191 was originally scheduled for completion by 1989 [Valk87]. As the *perestroika* reforms progressed, the development time was drawn out. Although financing continued, it remained problematic. It was sufficient to support the development teams, but insufficient to get the customized integrated circuits built by the electronics industry. As users and factories began operating in self-financing modes, there were few who could afford to allocate the huge sums required for the development, manufacture, and purchase of such a machine. The design reportedly was scaled back to include four pipelines rather than sixteen. Others have reported that by the end of 1992, only a scalar version of the machine was being developed. But even individuals within NITsEVT remain skeptical about the prospects. ‘‘Who would have use for such a machine? No one,’’ commented one senior NITsEVT manager.

7.5 Other HPC at the Scientific Research Institute of Control Computers

7.5.1 PS-3000

Like the PS-2000, the PS-3000 grew out of a collaboration between the Institute of Control Problems in Moscow (IPU) and NIIUVM. During the mid-1970s, researchers at

IPU under V. V. Ignatushchenko worked on parallel processing ideas and developed ideas about pipelined processing and dynamic allocation of computing resources.

To overcome the limitations of a static architecture typical of SIMD machines in which one control unit has exclusive access to a fixed set of processing elements, Ignatushchenko designed a two-phase execution process in which control units and processor fields were decoupled from one another. A set of control units would operate independently and in parallel, feeding instructions into a single buffer. A set of so-called processing element blocks, each consisting of multiple processing elements, would independently and in parallel execute instructions. Any instruction could be initiated by any control unit and executed on any processor block. Simulations showed that such an approach could result in 22-36% greater loading coefficient than a rigid configuration [Igna84, 128-137].

The other key conceptual idea was pipelining, now a standard feature of supercomputers. The idea itself cannot now be considered a novelty, but Ignatushchenko claims that he developed it independently and rather fully before architectural details of the Cray-1 computer were disseminated in the Soviet Union. He claims that evidence that Cray Research incorporated pipelining ideas proved convincing justification for his ideas and played a large role in convincing authorities to support his project. The Cray-1 was not the first computer to implement pipeline processing, however. The CDC 7600, introduced in 1969, incorporated eight pipelined functional units [Hock88, 16] and the basic ideas of overlapped execution of different computer functions had been around considerably longer than that [Rama77]. We do not know to what degree Ignatushchenko was familiar with the CDC 7600, although although such information could have been available, given CDC activities in the Soviet Union during the 1970s.

In its implementation, the PS-3000 incorporated only the dynamic linkage between control units and processing elements. True pipelining of functional units was deemed too complex to implement. As an alternative, a quasi-SIMD vector processor consisting of multiple processing elements was created which in many respects mimicked the operation of a true vector-pipelined processor. Although its performance on various vector operations was more uneven than that of Western pipeline processors, it did perform reasonably well on many operations, given the technology with which it was constructed.

NIIUVM-IPU collaboration on the PS-3000 began in 1975/76, and construction of the prototype began around 1978. Although a prototype was completed sometime between 1982 and 1984, the machine never entered large-scale production. On the order of ten machines were manufactured.

The 32-bit PS-3000 multiprocessor was designed for use as the top level of complex hierarchical data processing and control systems, such as real-time systems in atomic energy plants, simulation systems, etc. Key requirements were high performance, high reliability, and compatibility with prior processor control systems developed at NIIUVM.

The PS-3000 consisted of two or four scalar processors where each pair of scalar processors was associated with a single vector processor. Scalar processors executed different tasks, or different branches of the same task, but used the shared vector processors to execute vector instructions [Impu85, 5; Iosh87, 114]. Of the approximately ten machines which were manufactured, no full configuration machines were built. Reportedly, all systems contained two scalar processors and one vector processor.

While Ignatushchenko designed the vector processors to use pipeline processing and the Cray experience demonstrated the viability of this approach, the PS-3000 vector processors did not use true pipelining. As explained by V. M. Borisenko, one of the chief NIIUVM engineers on the project, the representatives from the prototype development

factory objected to the complexity of a true pipeline implementation. They claimed that it was too difficult to manufacture a pipeline given the SSI and MSI chips available at the time. As a compromise, they implemented the vector processor as a set of eight processing elements operating in a quasi-SIMD mode [Iten85]. The vector processor consisted of a control unit and a processor field. The control unit received operations from the scalar processor, divided them into instructions and data and, when operands were available for a particular operation, placed the instruction and operands in a buffer of instructions to be executed. The processing elements operated independently of each other, fetching the next operation in a buffer for execution. In practice, vector operations involved the execution of the same instruction on multiple data items. Thus the operands were fed to the processing elements in a round-robin fashion. In contrast to a true SIMD architecture in which all processing elements operate in lock-step, the PS-3000 processors operated independently such that PEs could begin executing the next vector operation even if not all PEs had finished executing the current one [Iten85].

The scalar processor incorporated some pipelining at the system level, in the instruction processing functions. This concept, in which the fetching, decoding, operand fetch, and instruction execution of one instruction can be overlapped with that of another was first incorporated commercially by IBM in 1977 in its 3033 mainframe [Lusa78; Pras89]. It was not common practice when the PS-3000 was first designed.

All processors shared a field of four or eight Mbytes of memory, organized in units of two Mbytes each. The units had independent power supplies and independent links with CPUs and I/O processors. The shared memory and independent operation of modules not only facilitated inter-process communication and enabled a single, re-enterable copy of the operating system to serve all processors, but also made it possible to run the system in

a fully redundant mode to increase reliability [Impu85, 5; Iosh87, 114]. The system had an addressable virtual memory space of 256 Mbytes [Reza83; Impu85, 3].

To enable the central processing units to devote more time to computational tasks, I/O tasks were off-loaded to peripheral I/O processors which handled I/O dispatching, I/O program execution, and queue control. The I/O processors were attached to the scalar processors and provided exchange of data not only between peripherals and main memory or the scalar processors, but also between configuration subsystems [Impu85, 5-6; Iosh87, 114]. The latter included a data processing subsystem based on a PS-2000 [Impu85, 5]. Each configuration subsystem was attached to two I/O processors, primarily to provide redundancy and improve system reliability [Impu85, 6].

The PS-3000 served as the basic computation engine for so-called regional geophysics computing systems (RGVK). Three RGVK versions were planned, the K143-12, K143-13, and K143-14, shown in table 7-4. The three differed primarily in the number of processors and mix of peripheral devices.

Why did the PS-3000 never enter series production? Individuals give different answers, but a common thread between the explanations is a lack of effort on the part of developers to see their project through. Some claim that by the time the machine was ready for series production, the component base had become completely obsolete; developers felt that their energies were better spent designing the next generation than investing the time, energy, and resources necessary to both see the machine through to production and establish the support network necessary to maintain it in the field. Others claim that the project was terminated for economic reasons. The Severodonetsk Instrument-Building Factory was reluctant to manufacture the PS-3000. In addition, there was little demand for the machine. As the Soviet Union opened to a flood of imported high-end personal computers in the late 1980s, many potential customers preferred to purchase

	K143-12	K143-13	K143-14
Number of scalar processors	2	4	2
Max. performance on fixed-point add (MIPS)	6	12	6
Number of vector processors	1	2	1
Max. performance on fixed-point vector adds (MIPS)	10	20	10

Table 7-4 PS-3000 Configurations
Source: [Impu85]

personal computers which, although not parallel machines, compared not unfavorably with the PS-3000 in terms of performance, memory, and reliability. Still others lay the responsibility solely at the feet of the developers, claiming that they simply did not make the effort to push the project through to completion. Each of these perspectives undoubtedly has merit and the final answer probably lies in a combination of these reasons.

7.5.2 PS-3100

The successor to the PS-3000, the PS-3100, was at the scientific-research stage *(nauchno-issledovatel'skaya razrabotka)* in 1990. It differs from the PS-3000 in a number of ways. First, the processing elements within the vector processor will contain a small cache, accessible to the programmer. The number of vector registers will be increased, and the physical dimensions will be decreased. In addition, a number of changes to the basic approach are being made to make the system more marketable. These include implementing Unix as the basic operating system, writing a C compiler to increase the amount of software available, and designing the vector processor so that it can interface with a number of processors besides the system's own scalar processor. Given a standard

set of interfaces, the PS-3100 vector processor could become a high-performance attachment to existing machines. Given the lack of funding for the project, however, it is not likely that it will be completed.

7.6 HPC at the NII of Computing Systems (NIIVK)

7.6.1 M-10

During the 1970s a multiprocessor called the M-10 was developed at the Scientific Research Institute of Computing Systems (NIIVK) under the direction of M. A. Kartsev. Kartsev had worked on some of the earliest Soviet systems, including the M-2 which was prototyped in 1952 [Tadz77, 6-7; Yers80b]. The M-2, a medium-sized, general-purpose machine running at 2000 operations per second and emphasizing economy and efficiency over size, was not considered a ‘‘high-performance’’ system [Kuzn61; Rudi70; Yers80].

We do not know what Kartsev, who had close ties to the military, worked on during the 1960s, but during the 1970s he focused on signal and image processing systems [Supe91, 27]. The M-10 was a vector-oriented multiprocessor for such applications. The machine had multiple types of processors for various functions which shared main memory. The main processor section consisted of two ‘‘lines’’ (sets) of eight 16-bit processors, each set operating in SIMD mode. An additional specialized processor performed logic operations on boolean variables in parallel with the main processor lines. These could provide conditional control functions and masks for regulating the main processors. A further specialized processor performed index operations. It had an average performance of five MIPS, and five Mbytes of main memory [Kart79; Kart81; Sots790913]. The M-10 operating system could support multi-tasking with up to 48 batch and interacting jobs [Bely80].

The distinguishing feature of the M-10 was its ability to process conveniently in parallel data of different formats, dynamically changing the clustering of processor to match the format of the data at hand. Data could be fetched from memory in units of 2-64 bytes and the eight 16-bit processors could execute identical instructions on two 64-bit, four 32-bit, or eight 16-bit numbers [Kart79].

Of the five Mbytes of ferrite-core main memory, one Mbyte was reserved for the resident portions of the operating system and four Mbytes were available to users [Kart80; Gakh82]. Each process could access up to four Mbytes of virtual memory. The system used peripherals designed for the ES mainframes [Kart79; Sots790913; Grin82].

It was possible to execute different instructions on different pieces of data (MIMD), but higher processing rates were obtained in SIMD mode. The processors each executed instructions at an effective rate of one per 1.8 microsecond machine cycle [Kart79; Kart81b].

The M-10 was used as a platform for the development of early parallelizing compilers. The earliest efforts, such as a compiler for Algol-60 completed in 1974, focused on parallelizing sequential programs. Later efforts on FORTRAN compilers for the machine placed the burden of indicating parallelism on the programmer [Berk85; Prog86; Varc86].

The M-10 was operational by 1979, but possibly several years earlier [Sots790913; Varc86]. We do not know how many M-10s were manufactured, but published reports seemed to indicate that this was a serious industrial machine. A number discuss actual use in modeling seismic activity and plasma behavior [Bere80; Same85]. The M-10 continued to be used well into the 1980s [Golo85; Golo87].

7.6.2 M-13

Kartsev died in 1983 and was succeeded as director of NIIVK by A. A. Novikov [Kart83]. Under Novikov's direction an M-10 successor, the M-13, was developed and passed state testing in 1990 [Supe91, 27].

The M-13 was designed primarily for real-time applications. Building on Kartsev's work, it incorporates multiple heterogeneous processors operating on shared memory. The three basic processor types are the central control processor, the central processor for vector operations, and specialized processor for executing specialized functions [Grin90, 4]. The 32-bit vector processor reportedly runs at 50 MIPS [Supe91b, 9]. It contains six vector registers each holding up to 256 vector elements [Grin90, 13]. The vector processor has eight multi-format arithmetic units. Each can function as one 64-bit unit, two 32-bit units, four 16-bit units, or eight 8-bit units [Supe91b, 8].

The most significant features of the architecture, according to [Grin90], are the organization of memory using a memory controller and the address protection mechanism. The M-13 implements capability-based addressing in which each process has access, potentially, to the entire virtual memory space. This concept, developed in a number of Western system such as the Cambridge CAP during the late 1970s, does not limit each process to sharply delineated portions of virtual memory, as is conventional in most commercial systems and in the M-10 as well [Grin90, 15-18]. Although capability-based addressing allows more flexible sharing of data between processes, it places greater demands on the system which must guard against unsanctioned memory access. The M-13 incorporates a number of memory protection features in hardware.

The M-13 has a 300 nsec cycle time and is built using quite conservative Logika-2 TTL technology, the same used in the El'brus-1 during the late 1970s [Supe91b, 9].

7.6.3 El’brus-M14E

Another high-performance project carried out at NIIVK during the late 1980s was called the El’brus-M14E. This machine was an effort to implement a system having the same basic architecture as the El’brus-3 at ITMVT, but on a smaller scale [Baba89]. Naturally, there was close cooperation between ITMVT and NIIVK on this project. This 64-bit system was designed to have 2-8 processors in the maximum configuration. Each processor, built using ECL technology was to have a clock period of 25 nsec and a peak performance of 160 Mflops. The full configuration would have a theoretical peak performance of 1.28 Gflops [Baba89, 879]. This project had effectively ended by 1991 because of a lack of financing.

7.7 Dynamic Architecture Machines

Valeriy A. Torgashev and others have conducted research on dynamic architecture machines (MDA) since the late 1960s, first at the Leningrad Institute of Aviation Instrument Building (LIAP) and later at the Leningrad Institute of Informatics and Automation of the USSR Academy of Sciences (LIIA). Earlier references to these machines use the term ‘‘recursive architecture machines.’’ The earliest presentation of some embryonic ideas is found in [Glus74]. Arguing that the von Neumann architecture–characterized by low-level machine language representations, sequential execution, and linear storage in memory–required a burdensome amount of complex software to bridge the distance between the hardware and the applications to be executed, the authors proposed a so-called recursive architecture solution. Such a machine would be characterized by a recursive internal language, a recursive parallel control method, a recursive memory organization, and recursive internal and external architectures. The internal language would allow a programmer to define a set of low-level primitives and then construct additional levels of language elements that themselves would be viewed as entities, but which would be de-

composed during execution. Control would be based on dataflow principles in which operators (at any level) would execute as soon as all the necessary operands were available. Memory would be stored and referenced as objects, rather than memory locations; the logical-to-physical mapping would be done at run-time. The internal architecture, the set of connections between processors, would dynamically change in response to the structure of the task being executed. Finally, the machine would consist of a hierarchy of physical switches that would allow one to cluster processors physically together. These ideas, expressed in very general terms in [Glus74] have, to varying degrees, been incorporated into Torgashev's later work on dynamic architecture machines.

Torgashev's work falls into the general category of non-von Neumann architectures that are classified as data-driven and demand-driven [Trel82]. In dataflow machines, the availability of operands triggers the execution of the operation to be performed. In demand-driven (reduction) computers, the requirement for a result triggers the operation that will generate it. Two primary motivations behind these developments are the desire to improve performance by uncovering and utilizing the greatest amount of parallelism possible in a program, and to support declarative languages, particularly functional languages which have desirable properties for programmability and execution [Huda89; Vegd84].

The open literature does not provide a great amount of detail about Torgashev's machines and it is difficult to see through Torgashev's terminology to make a clear comparison with Western work. Comparisons made here are, therefore, somewhat tentative. Torgashev's machines are described in [Pono83; Torg84; Pono87; Torg88; Torg89].

The underlying model of computation is the dynamic automata network (DAN). In such a network, the division of a problem into algorithms and data (typical in traditional programming) is minimized. Each automaton represents an object of a given problem,

which can be data, operations, relations, references (pointer-like entities), or physical machine resources. Automata can be complex, consisting of other, simpler automata. The execution of a problem, represented by a DAN, consists of applying transformations to the DAN until no further transformations are possible. The final structure represents the solution. An important point is that not only data can be transformed, but also other types of automata, such as relations and operations.

Treleaven et al. describe the program organization as the way machine code programs are represented and executed in a computer architecture. Two basic categories of mechanisms are the data mechanism that defines the way a particular argument is used by a number of instructions, and the control mechanism, which defines how one instruction causes the execution of another [Trel82]. It appears that the MDA use at least two of the three data mechanisms described by Treleaven–by value and by reference. It also appears that the MDA make provisions for using each of the three control mechanisms described in that article: sequential, where a single thread of control signals passes from one instruction to another; parallel, where control signals the availability of arguments and an instruction is executed when all its arguments are available (as seen in dataflow); and recursive–where control signals the need for arguments and an instruction is executed when one of the output arguments it generates is required by invoking the instruction.

The lowest level of automata are represented by so-called program elements that consist of code fragments to execute individual instructions (operator automata), to provide simple or complex memory access (data automata), to verify a relationship (relation automata), and to manage the assignment of program elements to physical resources (resource automata).

The ES-2704 and ES-2727 are two implementations of Torgashev's ideas. The structure of the ES-2704 is shown in figure 7-3. The machines consist of two basic compo-

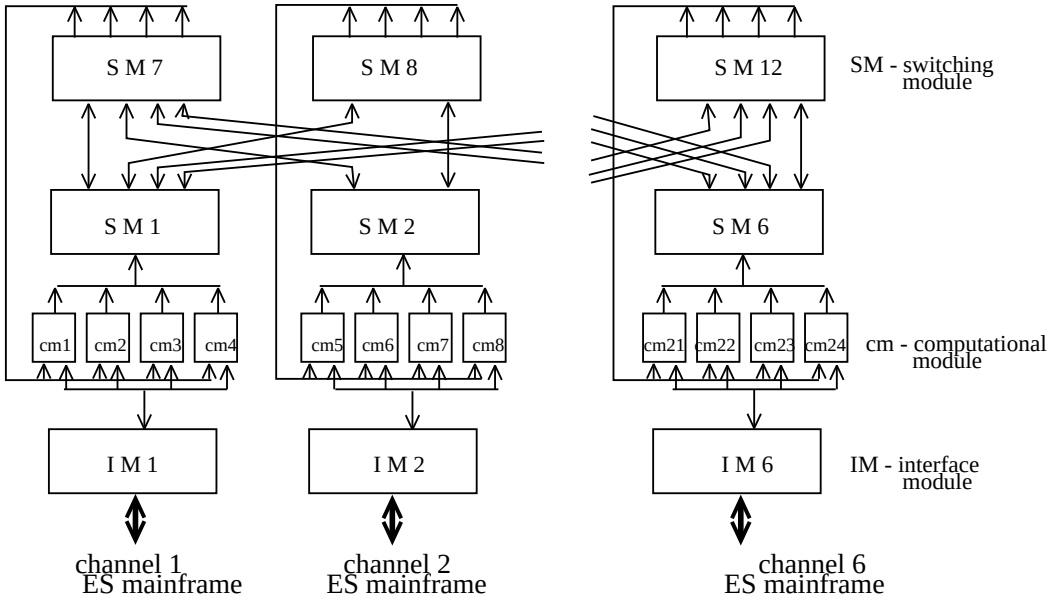


Figure 7-3 ES-2704 Structure
Source: [Pono87; Torg88]

nents: a set of computational modules, and a set of switching modules. The computational modules consis of a set of processors–administrative, execution, control, and three switching units, in the case of the ES-2704–together with shared memory. Automata exist in memory and are queued to the execution processor. If the execution processor becomes too heavily loaded, automata are queued to the switching processors that send them to other computational modules to be executed. The ES-2704 consisted of up to 24 computational modules, 12 switching modules, and six interface modules that provided access to external host computers, which provided access to external data stores. Several prototype models were built using the ECL IS-500 series of chips, Soviet analogs to Motorola's MECL 10K series [Torg88, 182]. With an 80 nsec clock period, the peak performance was 100 MIPS on 16-bit, fixed-point data [Pono87; Lomo91, 20]. The ES-2727 consists of 72 computational modules, 36 switching interface modules arranged in a

two-tiered hierarchy. The theoretical peak performance of this machine is said to be 750-1000 MIPS [Torg89].

In comparison to Western work on data-driven and demand-driven machines, the MDA have a greater ‘‘real-world’’ flavor. First, although the mathematical foundation for the MDA is not clear in the literature, it is not clear that it is as theoretically pure as that underlying many Western machines of this class. For example, it is not clear that the MDA are free of side-effects, a property which makes functional programming attractive. Second, unlike most Western work which uses either control flow, dataflow, or reduction as the computational model, the MDA appear to provide constructs that can be used to specify any of these modes. Third, the MDA apparently place significant emphasis on performance, perhaps compromising theoretical purity. Resources must be explicitly managed by a programmer. Automata can be assigned to run on specific physical processors or be stored in specified portions of memory. The systems also integrate uniprocessor and multiprocessor ideas. Each computational module can manage the execution of a number of automata concurrently, so it is possible to balance the cost of communications overhead with the benefit of distributed processing. Fourth, the MDA implementation emphasizes reliability. The switching network consists of switching processors that maintain ‘‘indicator sets’’ that record the status of the physical resources in the system. When processors or memory fail, automata can automatically be re-initiated on operating resources without the intervention of the user.

The MDA have had a long and difficult history. Torgashev persistently promoted his machine, trying to get resources to support it, for two decades. Although Glushkov was the lead author in the 1974 paper presented at IFIP ’74 in Stockholm, he had little to do with the project. To build a broader base of support, Torgashev had invited him, a prominent Academician very influential in computing, to be a co-author together with V. A.

Myasnikov who for many years was the head of the Main Administration of Computer Technology and Management Systems at the State Committee on Science and Technology (GKNT) [Prav740508].

The project attracted the attention of Control Data Corporation, but intergovernmental difficulties prevented any joint projects.

The first mock-ups were completed by 1979, consisting of only four modules. It took several more years to overcome problems with the machine's control system and develop a complete version, however [Samo89]. An ES-2704 neared completion in 1987 [Marc87] and a number of operational units had been installed by 1989 [Ekon89; Przh89].

The ES-2704 prototype was constructed at the NITsEVT prototyping facility in Moscow using the same technology used to build the ES mainframes. Financing for this stage was orchestrated by NITsEVT and lasted through approximately 1986 or 1987. Since he reportedly followed NITsEVT construction guidelines, building it with only a half dozen different types of boards, the machine could be relatively easily accomodated by ES mainframe factories. In order to get the system into series production, however, Torgashev needed to find a critical mass of customers willing to pay for the system. Although some organizations expressed an interest, most users were dissatisfied with the machine. It used its own programming language, Ryad, which was unlike any used by existing HPC users. Furthermore, around 1989, the language had not even been fully implemented, and coding had to be done largely in an assembler-level language. In short, programming the machine was very difficult and discouraged potential users. Also, because of its reliance on the mainframe I/O channels and disks, data could not be moved in and out of the processor fast enough to support high processing rates. Complicating mat-

ters, as *perestroika* progressed, potential customers became less and less willing and/or able to spend money on unproven technology.

A colleague of Torgashev's at LIAP, M. B. Ignat'yev, continued work on a related project called System-3M after Torgashev moved to LIIA [Myas84; Igna86; Valk87]. He later proposed the creation of a highly parallel architecture for a personal supercomputer based on the recursive concepts [Igna89].

7.8 Macro-pipeline Processors

Around 1976, inspired by multiprocessor projects within the Soviet Union and abroad, a number of researchers at the Institute of Cybernetics of the Ukrainian Academy of Sciences in Kiev (IK AN UkSSR) including V. M. Glushkov, A. A. Letichevskiy, Yu. V. Kapitonova, and I. N. Molchanov worked on the mathematical, software, and hardware foundations of a multiprocessor ‘‘macro-pipelined’’ computer [Glus78]. In this approach, which in its rough form resembles coarse-grain systolic computation, the data for a given algorithm propagate across a network of computational components [Mikh86].

An important element of this approach is that the computational units executed by individual processors are large relative to the amount of data transmission between processors so that communications overhead is minimized [Leti85]. Engineers were particularly interested in improving the distribution of tasks through the machine. The wanted to be able to distribute not only arithmetic and logical computation, but also control.

Work on the construction of a machine was initiated in 1979. NITsEVT helped arranged for GKNT funding from 1980 through 1986 or 1987 [Prav850827]. The macro-

pipeline processor concepts were implemented during the early 1980s in a system called the ES-2701. The design allowed for variable-sized configurations shown in table 7-5. [4]

The ES-2701 consisted of computation processors, control processors, a switching network, and a systems monitor based on a minicomputer. The computation processors executed a subset of the ES mainframe instruction set on 32-, 64-, and 128-bit operands [Mikh88, 157]. The switching system was a two-level network. Each switching processor in the first level was linked to 16 other system entities. These could be computational or control processors, system monitors, or channels to the mainframe host. Switching processors of the second level were linked to all first-level switching processors.

In addition to the standard system startup and monitoring functions, the systems monitor had the ability to restructure the ES-2701 dynamically in the event of a hardware or software failure without disrupting the execution of the task [Mikh88, 157].

The ES-2701 processor design was influenced by previous work at the IK AN Uk-SSR. In particular, the control processors used a high-level interpreted programming language [Pogr87]. This feature had its roots in the MIR computer, developed during the 1960s. The MIR, oriented towards scientific computation, included a high-level interpreted language designed for easy coding of scientific equations [Rudi70; Moro75; Litv81]. Letichevskiy and Molchanov both worked on the software for this machine [Ikan90]. In the ES-2701, the use of a high level language reduced the amount of control information that had to be transmitted through the interconnect network and simplified dynamic allocation of parallel branches of the task [Pogr87; Mikh88, 153].

[4] The figures for tables such as these vary considerably. For example, [Vnesh89] lists processing rates for the three configurations as 25, 50, and 100 MIPS, while [Vnes89b] gives rates of 133, 266, and 533 MIPS. The amount of main memory distributed throughout the configurations is 2.4M, 4.8M, and 9.6M words according to [Molc85]; [Vnes89] reports 25, 25, 100 Mbytes; and [Vnes89b] states 96, 192, and 384 Mbytes. These figures reflect differences between the first prototype (1984) and the second (1986). The latter used the same component base, but vector operations were added to improve performance on vector operations.

	ES-2701.01	ES-2701	ES-2701.02
Number of processors			
computational	48	96	192
control	8	16	32
switching	8	16	32
System monitors	2	2	2
Main memory per processor	.5-1	.5-1	.5-2
Performance (MIPS)	133	266	533

Table 7-5 ES-2701 Parameters
Sources:[Molc85, 106; Vnes89; Vnes89b]

To enable programming of the system at various levels, developers created the programming language MAYaK (a Russian acronym for 'Macro-pipeline Language' which translates as 'lighthouse' or 'beacon'). MAYaK is actually a family of languages with three parts: MPP (multi-modular programming) and two subsets, Prostor and YaDRO [Mikh86]. Other subsets, mentioned in [Goro84; Goro84b] apparently were not fully implemented. YaDRO is a low-level language for applications programming similar to standard programming languages. The primitives include process forking and joining and communication through shared memory. Prostor is used for assembling parallel modules into a parallel computation, implementing a given macro-pipelining scheme. MMP includes the other two languages and is often nearly synonymously with MAYaK.

This language has been categorized by one Western observer as a 'coarse-grain composition language' [Dong92b]. Not widely researched in the West, such languages are designed to provide mechanisms for allowing coarse-grain processes programmed in other languages into execute concurrently as one program. The benefit of MAYaK for tradi-

tional high-performance computing users is that existing FORTRAN code, for example, can be packaged with MAYaK headers and run as a parallel process on the macro-pipeline processors. None of the existing code needs to be changed. For example, in one case a 15,000-line FORTRAN written for an ES-1066 computer was converted to run on the ES-2701. The only changes to the code were the addition of MAYaK control statements. The program reportedly ran 33 times faster on a 41-processor system than on a single-processor ES-1066.

The first, 16-processor, ES-2701 prototype was completed 1984, in close cooperation with NITsEVT [Prav850827; Ikan90]. It was constructed using standard (but old) TTL ES mainframe components to speed the development time. The ES-2701 was designed as a special-processor attached to a mainframe host which would provide all interaction with peripheral storage. The host was an ES-1060 for the first prototype. A second prototype, completed in 1986, had an ES-1066 host and the complex was named the ES-1766. Such a configuration passed state testing in 1987 [Elec87; Kale87; Lari87; Przh89, 36]. This 48-processor unit reportedly had a performance of 133 MIPS [Przh89, 36; Vnes89b].

The prototypes were constructed at the VEM Plant in Penza [Ikan90]. Only three ES-2701 prototypes were ever constructed, and none in the full configuration.

Relationships with the factory were problematic. In spite of the support of NITsEVT and efforts to build the machine using standard ES equipment and components, the factory in Penza had little desire to build a machine designed in the Academy of Sciences, with unproven performance and a very small market. Although the machine was reasonably programmable because large portions of existing code could be used, the system was extremely constrained by the I/O bottleneck of the mainframe hosts which, in these configurations, limited to 2-3 Mbytes/sec. Whatever high performance could be achieved on

the multiprocessors was compromised by the mainframe and the peripherals. The time required to load a task was often a large percentage of the time needed to solve it.

Matters were further complicated when Minradioprom decided to move production of several machines, including the ES-2701 from Penza to Minsk, even though the production documentation for the latter had already been based on the equipment in Penza. The reasons for the move are not clear to us, although the most likely reason is that the ministry wanted to make the production facilities at Penza available for some other type of production. The end result was that the ES-2701 production documentation would have to be reworked, delaying matters considerably.

As the possibilities for direct negotiations with factories grew, engineers negotiated with others to try to get the system into production. Negotiations have not been successful, and no other systems along these lines were built.

Nevertheless, the machine continues to appear in articles. Design work on a successor consisting of up to 340 arithmetic processors and a peak performance of one GIPS, the ES-1710, is reported in [Przh89; Lomo91], but the prospects for such a machine are currently very poor.

7.9 Multiprocessor Computing Systems with Programmable Architecture (MCS PA)

The Scientific Research Institute of Multiprocessor Computer Systems (NIIMVS) is home to a large body of research on multiprocessor systems with programmable architecture. The Institute was founded in 1973 in the city of Taganrog on the coast of the Sea of Azov through the efforts of A. V. Kalyayev.

Kalyayev had been involved during the 1960s in the development of digital differential analyzers and around 1964 had built such a machine consisting of 100 parallel digital integrators. During the 1970s his work underwent a transition from integrators which

were designed to model analog machines digitally to computing systems with a more general-purpose orientation.

Kalyayev wanted to draw together into a single organizational framework a number of different scientific organizations which would support the research cycle from theory to production, and provide a training ground for students whose talent could be drawn on by the research institutes. NIIMVS is the research institute in a complex including an institution of higher education, a design bureau, a pilot production plant, and several associated laboratories in the industrial ministries [Mvs89]. This research complex was formed within the Ministry of Higher Education (MinVUZ), and is the leading example of high-performance computing in that ministry.

Strongly influenced by the work of Yevreinov and others described in section 3.3.2.1, work conducted at NIIMVS during the 1970s was labeled ‘‘homogeneous computational structures.’’ This was later changed to ‘‘multiprocessor computing systems with programmable architecture’’ to minimize confusion with other research on homogeneous computing systems carried out elsewhere in the country. An early implementation of the ideas described below was called the OVS-80, consisting of 16 general-purpose (non-NIIMVS) processors. It was begun in 1980 and the first units were completed in 1982.

7.9.1 General Characteristics

A computer’s performance depends on how well its architecture is suited to a particular algorithm (or how well the algorithm can be adapted to a given architecture). This is particularly true in the field of parallel systems [Hock88]. Ideally, a system with a programmable architecture can be adapted to fit the structure of data flows of a particular algorithm. The programmable architecture concepts permeate nearly everything done at NIIMVS, which has divisions devoted to architecture of multiprocessor systems with pro-

grammable architecture, signal processing, robotics, neuro-like computing structures, microelectronics, and others.

A second design philosophy in effect at NIIMVS is that the machines should reflect the ‘‘natural’’ language of the users, primarily engineers. The systems are characterized by high-level instructions (called ‘‘macro-operations’’) which execute such functions as solving systems of linear equations, differential equations, fast Fourier transforms, calculation of eigenvalues of matrices, and many more. These are executed on ‘‘macro-processors.’’ Effort has been made to define a set of macro-operations which is on the one hand general enough to solve many numerical problems and on the other hand minimalist to keep the structure and tuning of macro-processors simple.

The ideological consistency of the many projects is noteworthy. The adherence to crossbar style interconnects, processors which execute complex operations, and the ability to tune the interconnects has remained nearly absolute for many years. This uniformity reflects A. V. Kalyayev’s strong control over the research agenda.

NIIMVS has developed a number of special- (such as for signal processing) and general-purpose systems which share at least the following features:

- A number of macro-processors which can operate asynchronously. The macro-processors consist of multiple ‘‘microprocessor sections,’’ which in turn consist of an elementary processor performing basic arithmetic and logical functions, a microswitch controlling links with other microprocessor sections, local memory, a switch memory unit storing switching patterns, and an elementary operations memory unit storing the computational instructions for individual macro-operations.
- A programmable cross-bar switching system. This switch provides point-to-point links between any two macroprocessors.

- A distributed, programmable memory system, enabling access of complex memory objects.
- An internal user-oriented high-level machine language based on a set of macro-operations and macro-switches.
- An internal machine language oriented towards the distribution of macro-operations and macro-switches in a parallel multiprocessor configuration.
- A macro-dataflow model of computation. Individual macro-operations execute when all operands are available, but the high-level organization of computation is more centralized than in true dataflow machines. A centralized control manages the allocation of macro-operations, but individual macro-processors have exclusive control over their execution.

To support systems development, one NIIMVS division works on components, building specialized processor, memory, and switch components for the programmable architecture machines. Products include the K1815 family of chips for general-purpose and digital signal processing, the SK 1509 KP1 16x16 single-chip switch, and many others [Niim90; Kaly88b; Kalo86; Bobk86; Niim89b]. Thanks to Kalyayev's personal contacts with individuals in the main administration for science in Minelektronprom, he was able to ''push through'' three chips–a macro-processor, a switch, and a memory unit–into series production during the early 1980s [Vadi84].

7.9.2 ES-2703

The ES-2703 is one general-purpose, high-performance MCS PA. It was one of the machines funded by the State Committee on Science and Technology in conjunction with NITsEVT. The 32-bit prototype system, shown in figure 7-4, was linked to an ES-1061 mainframe host. It incorporated sixteen macroprocessors, a switching system, a control

unit, an exchange processor. The operating system is resident on the control unit and manages the control unit and field of node processors (macroprocessors). Each task consists of a control module running on the control unit and a processing module running on a set of one or more node processors. All node processors participating in a given processing module execute the same program, but on different data. Multiple processing modules can execute simultaneously on the field of node processors. Consequently, node processors operate in uniprogram mode, while the control unit supports multitasking [Babe91].

Although direct connections with peripheral storage were designed for later machines, the ES-2703 prototype accessed all peripheral storage through the host (contrary to that shown in figure 7-4). The machine was designed primarily for gas-dynamic applications, which influenced the selection of the word-length, the macro-operations, and computational methods, and prompted designers to build a machine in which the processing elements could be grouped into arbitrary configurations [Babe89; Baba90]. Design began in 1981 and the technical statement of work was worked out in 1982. Between 1982 and 1985 the prototype was constructed. The machine passed state testing in 1986 [Mvs89]. Only two of the machines reportedly were built.

The ES-2703 macroprocessors were constructed using Soviet analogs to the AMD 2900 bit-sliced chips, called the 1804 series. These chips were used because during system design NIIMVS' own chips had not entered production and were not ready for use. Since the focus of this project was on systems architecture, it was felt that the development process could be simplified and speeded up by using traditional components, at least for the first version of the machine. Each macroprocessor had 64 or 256 Kbytes depending on the capacity of the memory chips [Babe89].

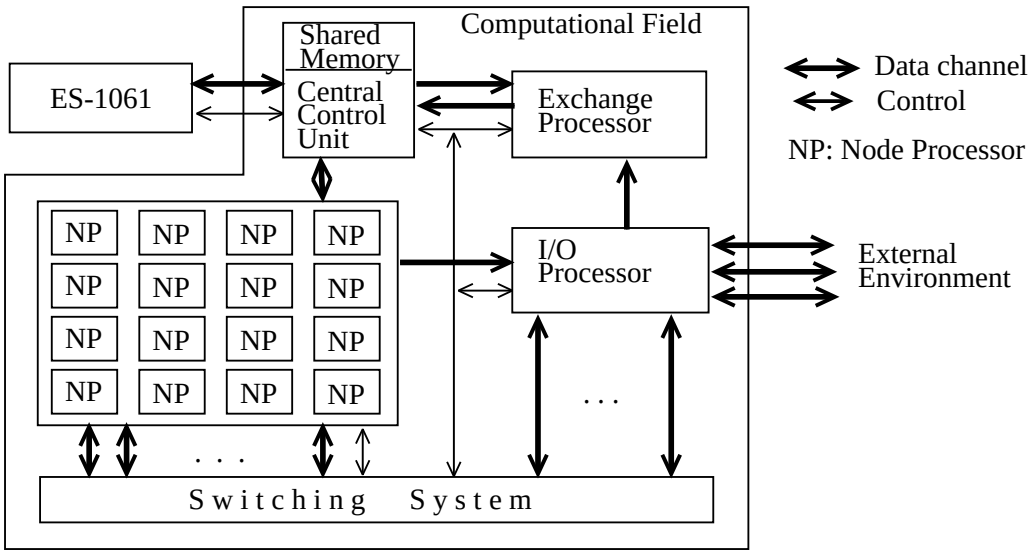


Figure 7-4 ES-2703 Structure
Source: [Kaly88, 162]

The published literature gives inconsistent information about system performance. Figures of 30-50 MIPS and 125+ MIPS are given in [Babe88; Kaly88; Lomo91]. Performance measurements reported in [Babe88] show a curious anomaly. Table 7-6 shows the speed-up in moving from one processor to four for a number of basic computations. These data seem to indicate that the system does not scale well even though the tasks are easily parallelizable.

7.9.3 ES-2703 Successors

Following completion of the ES-2703, design work underwent a bifurcation. The quest for high performance drove the design of a system with 128 processors and a theoretical peak performance of 1 Gflops on 64-bit data. The realities of the late 1980s–the difficulty of finding sponsors for large-scale work, the growing reluctance of industry to assimilate the production of unproven machines with weak markets–drove the develop-

Task	Speed-up (4 proc/1 proc)
Addition of two 12x12 matricies	1.26
Multiplication of two 12x12 matricies	2.80
Scalar product of two 132 element vectors	1.19

Table 7-6 Speed-up On ES-2703
Source: [Babe88, 8]

ment of small-scale macroprocessor systems which could be used as scientific co-processor attachments to a personal computer [Niim89; Niim89b].

7.9.4 Special-purpose Systems

Besides general-purpose systems, NIIMVS researchers have developed a series of special-purpose systems for real-time signal and image processing [Niim89c]. These include the PVK-460 which is built using the NIIMVS switching and macroprocessor chips manufactured by Soviet industry. The PVK-460 includes vector and scalar modules which can be combined in a variety of configurations, depending on user requirements. The machine has a word-length of 20 bits. Each vector module is equipped with 32 macro-operations (expandable to 512), and executes at 50-100 MIPS. Containing up to a total of 1024 processors and a theoretical peak performance of 460 MIPS, the PVK-460 was completed in 1988. A successor constructed during the early 1990s, the PVK-1600, has a theoretical peak performance of 1600 MIPS. A similar system has been built for medical image processing.

7.9.5 Research Trends

Two trends have characterized developments during the 1990s: an orientation towards smaller and/or specialized systems, and an orientation towards integrated systems. Both preserve the essential elements of the programmable architecture philosophy which has dominated NIIMVS research, and both reflect the changing nature NIIMVS' funding environment. The overall level of funding declined in real terms, and the portion of income from contract work increased.[5]

Although declining, funding for fundamental research at higher-education institutions from the state budget remained relatively good in comparison with other branches of Soviet science. NIIMVS was also able to win some proposal competitions held by the GKNT and the State Committee on Education. Such funding supported fundamental research in systems which integrate aspects of artificial intelligence, knowledge-bases, and neural-computing into a single, unified system [Niim90]. It also supported a continuation of research on architecture, integrated circuits, and neurocomputers.

The next generation of macro-processors have been called ‘‘super-transputers.’’ The shift in terminology reflects public-relations efforts more than a fundamental shift in the research program. The features of distributed memory and hardware control of inter-processor communications arguably put the macro-processors in the same general class as transputers, and the higher-level orientation of the macro-processors leads to the ‘‘super’’ designation. Transputers are also being used as an alternative hardware platform for systems developers [Gigl930122, 3].

The deteriorating financial state of NIIMVS meant that other funding had to be found, through contract work in particular. Overall funding through military organizations has

[5] Reportedly, to 70% of the budget in 1990.

declined, but through 1990, at any rate, contracts for image and signal processing machines remained stable and in some cases reportedly increased. Under *khozraschet*, customers became increasingly interested in getting a real product for their money and accountability increased. Reportedly this improved the quality of work performed.

During the late 1980s and early 1990s it became harder to establish contact with Minelektronprom plants. Kalyayev lost important contacts within this ministry. Efforts to enlist factories in manufacturing a new generation of NIIMVS-designed chips have been unsuccessful. Such arrangements now have to be done on an economic basis, rather than administratively. NIIMVS does not have sufficient funds to pay the factory to develop new chips, and the factories are reluctant to support such work with their own funds.

NIIMVS also made changes to its organizational structure to improve its financial position. The Scientific Research Center for Super- and Neurocomputers (NITsSN) was created at NIIMVS in 1991 under the All-World Laboratory, an international organization headed by vice-president of the USSR Academy of Sciences Velikhov [Nits91]. The latter was established in 1989. There were no restrictions on the wages fund in centers affiliated with the All-World Laboratory, so workers could be paid more than was otherwise permitted. As a result, the majority of the contract work done at NIIMVS was carried out within the context of NITsSN. NITsSN therefore has a floating employee pool. Groups are formed on a temporary basis to work on specific contracts; when a task is complete, the group breaks up and is replaced by other groups. The charter of the All-World Laboratory also gave daughter centers considerable tax breaks, and the waiving of customs duties.

NIIMVS proper also experienced some minor reorganization. As of 1988 the individual divisions operated on *khozraschet* principles and gained the right to reorganize themselves as they saw fit, subject to the approval of the director. At least one division, de-

voted to neurocomputing, had reorganized itself on the basis of temporary, rather than permanent laboratories. Reportedly such changes were not encouraged by the institute's leadership, however, since there was a feeling that they threatened the cohesion of the work.

7.10 ES-2702

Like other ES-270x systems, the ES-2702 is a specialized processor attached to an ES mainframe host which provides monitoring facilities and access to peripheral storage, etc. While not a high-performance system, we mention it briefly for the sake of completeness. The ES-2702 is a machine designed for symbolic processing. Its input language is REFAL, a symbolic processing language with a long history of development in the Soviet Union [Fauc68; Turc86]. Work on a REFAL-oriented hardware platform began at the Institute of Applied Mathematics during the late 1970s under A. N. Myamlin [Myam86, 301]. Starting with the basic processor of the widely manufactured ES-1035 mainframe, the researchers implemented their own microcode to support symbolic processing operations. The resulting ES-2702 was connected via standard I/O channels to a general-purpose ES mainframe. Like most of the other ES-270x projects, the ES-2702 work was supported by NITsEVT [Myam86, 316-317].

7.11 ES-2705

The ES-2705 was a analog-digital multiprocessor developed at Riga Polytechnical Institute in Latvia. It was designed for solving field theory boundary problems which are used extensively in seismic prospecting, aero-/hydrodynamics, weather prediction and other applications [Spal90]. The machine was prototyped as a special-processor for an ES mainframe during the early 1980s, but more recently a version has been built which can be attached to a personal computer [Komp91]. Although data transmission is done digi-

tally, computation is performed in analog by representing equations as transformations on voltage levels within each processing element. Input voltages are introduced into the system and the solution is represented by the voltage states in the individual processing elements when the system state has stabilized.

7.12 Attached-Array Processors

During the 1970s and 1980s attached array processors (AAP) were developed at ES R&D facilities in the USSR, Bulgaria, and East Germany. These are specialized processors attached to general-purpose mainframes, often through selector channels. The host CPU accesses them using standard I/O commands.

Like the associated mainframes, these systems were strongly modeled after Western systems. Although they did not have very high performance rates when measured against Western developments in high-performance computing, they were one of the few HPC alternatives available to many Soviet users. Thanks to the long development times of other systems such as the El'brus, the use of the series-produced attached array processors became rather widespread, particularly in the oil and gas industries [Mair81; Ivan88]. Manufactured with the same component base and technology as their mainframe counterparts, they were assimilated into production with relatively little difficulty.

7.12.1 ES-2335

The ES-2335 passed state testing in 1979 [Niko79]. Designed for use with the ES-1035 mainframe, this system came equipped with a library of specialized routines which could be invoked through program calls from the mainframe host. These included routines for matrix and vector processing, solution of differential equations, signal processing routines such as fast Fourier transform, etc. [Niko81]. The ES-2335 had a performance of 10 MIPS (5 Mflops) on 32-bit data [Niko81; Niko82; Niko82c; Kezl86]. It was

constructed using the series production technology of ES mainframes. The unit was manufactured in Bulgaria [Niko82].

7.12.2 ES-2345

The ES-2345 was the first completely Soviet attached array processor. Developed at the Scientific Research Institute of Mathematical Machines (YERNIIMM) in Yerevan, Armenia, this unit was introduced in 1978 for use with the ES-1045 and passed state testing 1979 [Meli79; Seme82]. The ES-1045 was also developed at YERNIIMM. Both system were manufactured at the Kazan' Computer Plant. The ES-2345 uses pipelined computation in addition and multiplication blocks to achieve a performance of approximately 30 MIPS (6.4 Mflops) on 32-bit data [Kuch81, 178; Seme84; Kuch85]. Hundreds of ES-2335 and ES-2345 AAP were used in the Soviet Union.

7.12.3 MAMO 1-M

Developed at approximately the same time as the ES-23x5 systems, the MAMO 1-M (Matrix Module) is an attached array processor designed by the East German Robotron computer manufacturer for use with the ES-1055 and ES-1057 mainframes [Merk80]. The ES-1055M was an upgraded version of the ES-1055, often equipped with a MAMO processor [Grue81; Muen81]. Unlike the Russian attached array processors, MAMO is connected to the host computer not via I/O channels, but with direct connections to memory. The host computer interacts with the AAP through a specialized set of instructions rather than through I/O instructions [Przh89, 36]. The MAMO was designed with a set of array processing instructions which could be invoked directly from within the user program on the host computer. In other words, when an array processing instruction is encountered, an interrupt to MAMO is invoked. On the Soviet machines, the AAP is invoked to process a library routine, i.e., a pre-programmed set of instructions. The MAMO

therefore functions well when a program involves the execution of isolated array processing instructions, but has high overhead when the equivalent of a library subroutine of densely packed array operations is used. The MAMO has a performance of 5-10 Mflops [Rajm88; Robo88]. It was also used in a number of Soviet installations for seismic processing and other applications [Robo88; Pois90, 2].

7.12.4 ES-2700

During the mid-1980s YERNIIMM introduced a successor to the ES-2345, the ES-2700 [Seme85; Musa86]. Unlike the ES-2345 and ES-2335, this system could be used with a variety of ES models, including the ES-1045 and ES-1068 [Musa86; Tass88]. It has a performance of 100 MIPS (30 Mflops) on 32-bit data [Seme88]. Although production was rather easily assimilated at the Kazan' Computer Plant, few units reportedly were built because the market for them was weak. Many users apparently preferred the ES-2706 (described below) because it was cheaper, more compact, and more reliable than the ES-2700.

7.12.5 ES-2706

Development of the ES-2706 began in 1980 at the Central Institute for Computers and Computer Technology in Sophia, Bulgaria. It entered series production in 1984 at the Computing Machinery Works in Sophia. By 1990, approximately 400 units had been manufactured of which reportedly 90% had been exported to the Soviet Union for use in seismic exploration [Prat90]. More than 50 installations reportedly contain more than one processor [Maly91; Tcha92]. It is a functional duplicate of Floating Point Systems' AP-190L and can run the latter's software [Coop88, 2; Prat90]. The ES-2706 is an improvement over the AP-190L. It removes the AP-190L's page restriction, increased memory size, and has re-designed connectors [Prat90]. Like the ES-2700 it can be used on a

variety of ES mainframes. Its performance was less than the ES-2700. The peak performance is 60 MIPS (12 Mflops) on 38-bit data [Mark86; Bere87]. Unlike the ES-2335 and ES-2345, it can be attached directly to external storage using a dedicated I/O processor which speeds up access considerably [Sagd87].

Unlike the ES-2335 and ES-2345, the ES-2706 was built using Schottky TTL technology, most of it imported, off-the-shelf technology from the West [Prat90]. For this reason, it was reportedly more reliable than its Soviet counterparts.

During the early 1990s, Bulgarians developed a number of successors to the ES-2706. The ES-2706M and ES-2709 both have a peak performance of 18 Mflops due to a second buffer in data memory and a second adder [Prat90]. We do not know of any shipments of these systems to the Soviet Union.

7.12.6 Loosely-coupled Array Processor Systems

The attached array processors provided an alternative path for Soviet users to high performance in many application domains. Several organizations worked on incorporating multiple attached array processors in a configuration with one or more mainframe hosts. Multiple user jobs could be distributed dynamically among the attached array processors, or different parts of the same job could run concurrently on them. In the West, E. Clementi, working for IBM, demonstrated that this approach could be used effectively on applications with coarse-grain parallelism [Bern84; Clem84]. The Soviet researchers did not duplicate his approach exactly, but were familiar with his work and even communicated with him [Anan87; Coop88].

A number of these systems, called the IZOT 1703E, were built by the Bulgarians, using the ES-2706 together with the ES-1037 host [Rabd88]. A system with four ES-2706s

was installed at the Institute of Space Research in Moscow in 1986. It was later upgraded to 10 ES-2706s [Sagd87].

A similar approach was used by Soviets in creating systems based on the ES-1066 or ES-1068 mainframes. The ES-1066.20 and ES-1066.60 are configurations manufactured at the Minsk Production Association of Computer Technology with four and eight ES-2706 processors, respectively [Mpov91, 33]. The ES-1068.17 consists of an ES-1068 mainframe with two ES-2700 and eight ES-2706 attached array processors [Mche88b; Tass88]. The first ES-1068.17 configuration was tested in 1988 [Przh89, 37]. Such a system was installed at the Computer Center of the Siberian Department of the USSR Academy of Sciences in Novosibirsk (VTsSOANSSSR) where it reportedly ran on some geophysics applications at 96 Mflops [Przh89, 37].

Researchers at the VTsSOANSSSR under N. N. Mirenkov developed significant portions of the systems software for both the Bulgarian and ES-1068.17/ES-1066.x0 which enables such loosely-coupled configurations to function. They have also built a large hierarchical, heterogeneous multiprocessor called the Sibir' to support the execution of so-called assembly-line parallel programming systems in which parallel programs are developed that can be dynamically and efficiently adjusted to the available resources of the multiprocessor. Typical applications include image processing, seismic data analysis, solution of systems of differential equations using difference methods, etc. The Sibir', completed in 1988, consists of three ES-1066 mainframes, eight ES-2706s, a STARAN-like associative processor, and a PS-2000. A programming system called Inya has been developed to integrate the software of all the components [Maly91].

7.13 Special-Purpose High-Performance Computers

Today's supercomputers, often costing millions of dollars, are cost effective because they are general-purpose, able to solve problems in a wide range of application domains.

It is possible, however, to design machines that cost significantly less, yet out-perform conventional supercomputers on a very narrow set of problems [Poll90]. Since 1989, Soviet researchers have published a handful of designs of descriptions of application-specific processors and multiprocessors. The most fully developed is a processor built at the Landau Institute of Theoretical Physics that is tailored to the Monte Carlo method for the Ising model with random links on a lattice containing 256x256 spins [Tala90]. This machine, a prototype of which was completed in December, 1989 [Tala90b], is capable of executing 4×10^6 elementary Monte Carlo steps per second. Upgraded versions of this machine, using a Western component base, are reportedly under development in the West.

Vyzhikovskiy and Kanevskiy of the Kiev Polytechnic Institute have published designs for a systolic specialized processor for solving systems of linear equations using the Gaussian method [Vyzh90]. Specialists at Moscow State University have designed a specialized processor for magneto-hydrodynamics problems capable of running at one Gflops [Bakh89]. Although the article makes it clear that such a machine has not yet been built, it presents an analysis of the requirements, showing that it could be built using currently available CMOS components. Donskov et al. at the Institute of High-Energy Physics in Protvino, describe a special-purpose processor for particle sampling by momentum in experiments studying central hadron collisions [Dons90].

CHAPTER 8. CONCLUSIONS

8.1 Introduction

In this chapter we review some of the overarching features of Soviet high-performance computing and cross-cutting themes of our study. Soviet high-performance computing has a rich and complex history, and the dynamic of technological and organizational development has been shaped by inter-related factors: elements of an organization's environment, levels of technological availability and organizational slack, belief systems and research strategies, and, not least, by the technologies and organizational structures themselves. Following a discussion of the contribution of the Soviet HPC sector to the scientific computing community, we examine the factors which have shaped Soviet HPC development in the past, and the impact the reform process has had on the HPC systems and the ability to development them. We will also discuss the impact of the reforms on organizational structures and the implications for HPC R&D capability.

In the next chapter, we will discuss the prospects for developers and users of Soviet high-performance computing, policy issues for Russian and Western policy-makers, and avenues for further research.

8.2 The Provision of HPC Capability To the Scientific Community

What has been the contribution of the Soviet high-performance computing sector to the Soviet scientific community? For all the years of research and resources invested in this sector, the amount of computing power provided has been disappointing. The most advanced systems have been characterized by extremely long (over 10 years) development cycles, plagued by reliability problems, and manufactured in only moderate numbers.

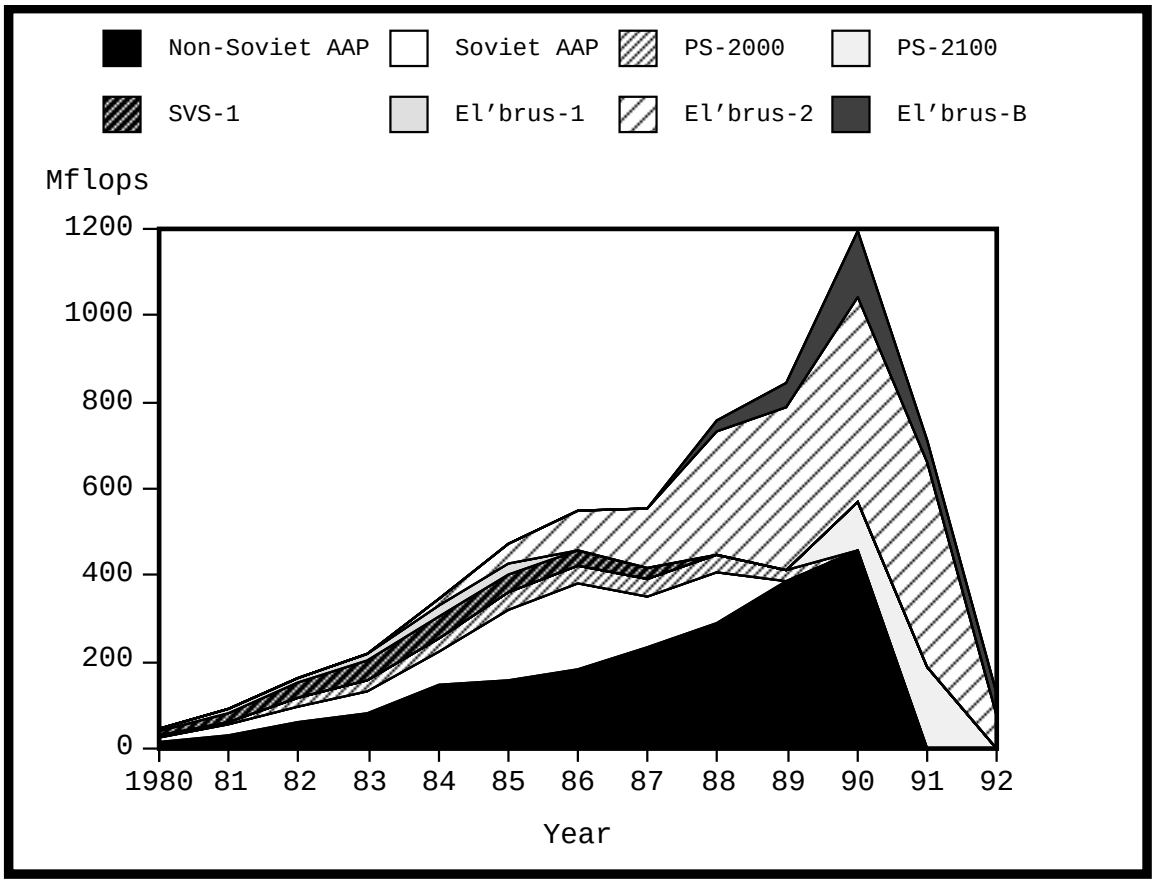


Figure 8-1 Annual Output of Series Produced Soviet HPC (in Mflops)

Figure 8-1 represents the annual contribution from 1980 through 1992 of the Soviet high-performance sector in terms of Mflops delivered by machines in series production. The specific annual figures are estimates, based on data of the total number of machines manufactured and the years of production. The graph is intended only to show in a rough way the amount of computing power delivered to the military and civilian scientific communities by the HPC sector. It does not reflect the utility of individual types of systems. In particular, the graph reflects performance on single precision or double precision (in the case of 24- and 32-bit machines) floating-point operations. The utility of machines such as the PS-2000 is under-represented, since while this machine had poor performance

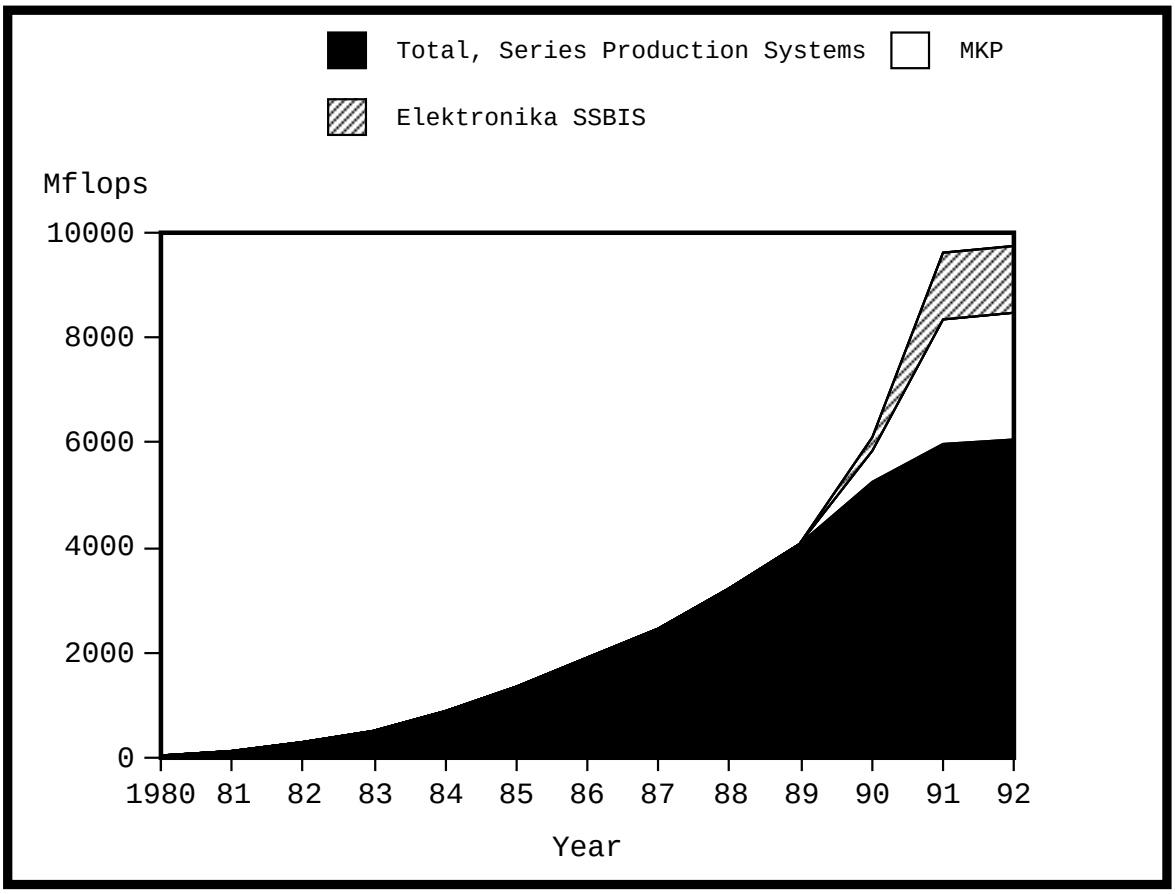


Figure 8-2 Cumulative Output of Soviet HPC (in Mflops)

on 48-bit floating-point operations, it had quite good performance on 24-bit fixed-point operations. Many applications required only the latter. The graph does not reflect the ease or difficulty of programming the system. The attached array processors cannot be considered general-purpose, since they executed only a limited set of library functions. In contrast, the El'brus computers had good performance on a much broader spectrum of applications.

Figure 8-2 shows the cumulative computing power, also measured in Mflops, delivered by the Soviet HPC sector. Two systems which could, in principle, dramatically increase the amount of computing power reached the prototype stage during these years.

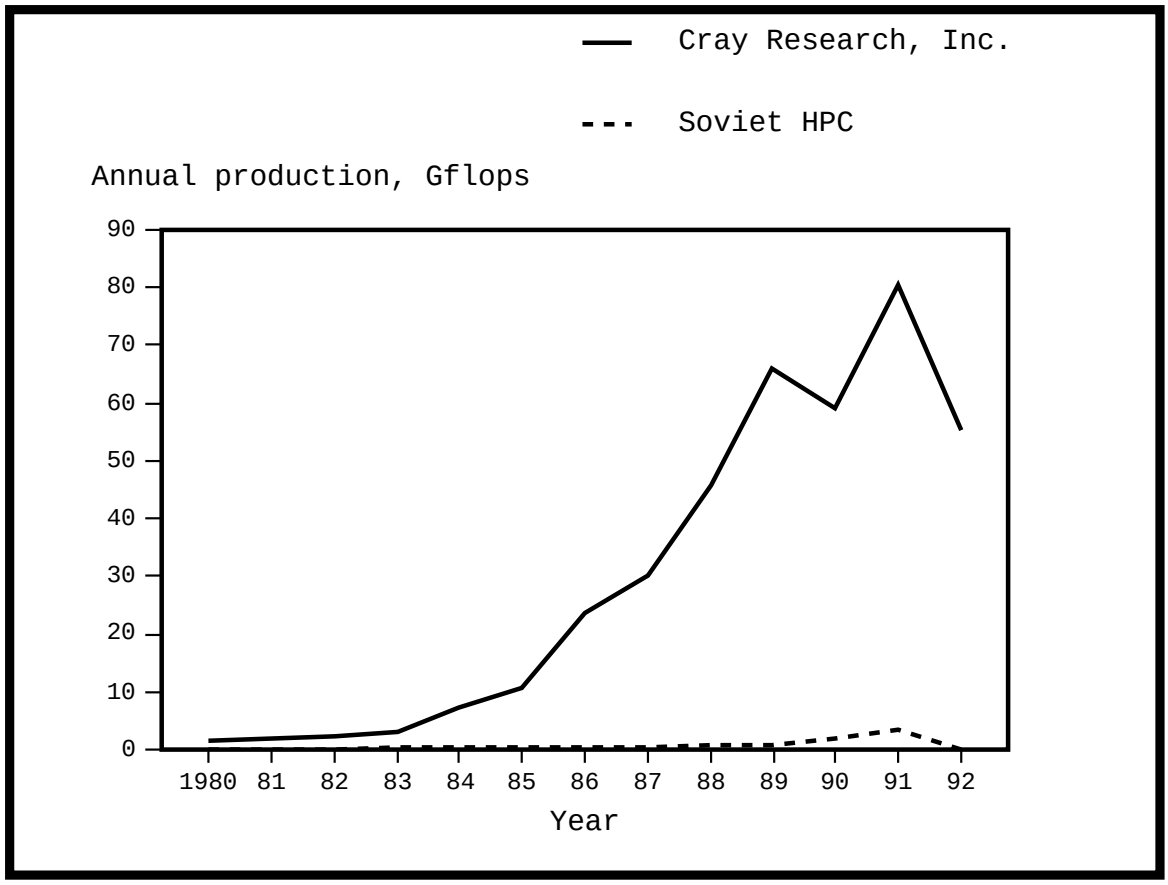


Figure 8-3 Comparison of Annual Output, in Gigaflops, of Cray Research, Inc. and Soviet HPC sector
Source:[Estimates, based on CRI annual reports]

They were not included in figure 8-1 because they have not yet entered series production. It is unlikely that they will. Figure 8-3 compares the aggregate computing power delivered by the Soviet HPC sector with that delivered by Cray Research, Inc. during the same period. Those systems which have been manufactured to date lack customers. In the absence of a market, more will not be manufactured.

Several items in the preceeding graphs are noteworthy. First, the ‘‘computer gap’’ between East and West has not been understated. The argument is frequently made that the Soviets had sufficient computing power for certain high priority applications such as nuclear weapons design and space mission control. It may well be the case that in selected

applications, and given a strong emphasis on the development of powerful algorithms and models, the computing power available was sufficient to meet certain objectives. However, in an age when computational methods have taken a place alongside experimental and analytical methods as a major tool of scientific research, scientific advance in a broad spectrum of research domains depends on the ability of large numbers of scientists to access advanced computing facilities. The Soviet scientific community has suffered considerably because of this lack. In chapter 9 we will discuss the options available to this community for gaining access to advanced computers.

Second, although the Soviet Union was the only Eastern Bloc country to have a serious high-performance computing sector, the attached array processors manufactured in Bulgaria and East Germany made a major contribution to scientific computing in the Soviet Union, in terms of raw Mflops delivered. Only during the latter half of the 1980s, as the El'brus-2 finally entered volume production, did the Soviet HPC sector begin to deliver large amounts of computing power.[1]

Third, during the 1990s the HPC sector has experienced a catastrophic decline in the amount of computing power delivered. In 1991, the Council for Mutual Economic Assistance (CMEA) was disbanded, and trade between its member countries began to be conducted on the basis of international prices [Wsj910107; Iht910629]. Transactions were no longer part of government-to-government agreements, but arranged by individual firms and industries. The volume of trade dropped precipitously. The sale of attached array processors by Bulgaria to the Soviet Union came to a standstill. As East Germany was re-united with West Germany, the sale of computer hardware also declined dramatically.

[1] We are deliberately limiting our discussion to high-performance computing. During these years significant numbers of mainframes and minicomputers also were being manufactured. For lack of other options, the scientific community often had to rely on these.

As the Soviet Union entered its final year, huge budget deficits forced dramatic reductions in state orders, which had been used to acquire most of the high-end systems like the El'brus. Individual customers could not afford, or chose not to purchase, indigenous high-performance systems. Orders for these systems declined to next to nothing by the end of 1992.

The decline in production of 1990-1992 is reflected in the flat curve during these years in figure 8-2. In reality, however, the amount of computing power from indigenous systems declined during these years as many El'brus systems were either turned off to conserve energy, or sold and scrapped for their precious metals content. We do not know how many systems have been affected.

8.3 High Performance Computing in the Soviet Context

As we look across the landscape of Soviet high-performance computing, we see that nearly every machine experienced great difficulties in reaching the prototype and series manufacture stages. Carried out within the Soviet economic and political system, the projects had many difficulties in common. Nevertheless, the specific reasons for these, and the relative impact of each varied from project to project.

Before the late 1980s, the nature of the Soviet economic structures and management created an environment which harmed the development of HPC systems more than it helped. High performance computing depended on an extensive infrastructure of supporting industries and enterprises under the management of an equally vast network, generally hierarchical, of ministerial, departmental, and Party administrations. In theory, this centralized arrangement should have streamlined development through greater coordination within the infrastructure, and promoted important projects through priority allocation of resources. In practice, it often gave none of these benefits, even within the military sector.

HPC illustrates the limits of prioritization under such a system. HPC systems required the priority development of an enormous number of technologies. The reality was that resources were not infinite, and HPC had to compete for them with many other advanced technologies, both civilian and military. Stern admonishments from the ministers and members of the Military-Industrial Commission could hasten development of one project for a time, but often at the cost of delays in others. Delays in any link of the development chain delayed completion of the end product.

The Soviet system of economic management provided a nearly impenetrable web which snared projects and reduced the speed and efficiency with which they could be carried out. Arrangements between organizations in the HPC infrastructure had to be made through the a bureaucracy characterized by administrative barriers difficult to circumvent and long chains of approval which needed to be negotiated. In order to take advantage of the centralized features of the economy, one first had to penetrate to the center. Projects with a high-level champion–an Academician, a minister, or an influencial member of an important government committee–often penetrated more easily than those that did not.

Negotiating the management labyrinths was a formidable task, but enlisting the participation of the the enterprises needed to provide the material inputs to a high-performance computer was also difficult. We have discussed the disinclination of factories to upset taut production schedules with the introduction of new technologies. When the volume of state orders exceeded a factory's capacity, the factory had an incentive to concentrate on continuing established production rather than suffer a drop in production to upgrade to a new technology.

The monopolistic nature of the Soviet economic system also hindered HPC development. Many critical components of a typical Soviet high-performance computer were manufactured at one, or at best a small number, of locations. When the development of a

given component at a certain plant was delayed or otherwise disrupted, HPC developers had little choice but wait, try to use political means to speed development, or develop the component in-house. Either option led to delays. Acquiring the same component at a different plant was usually not an option. Given captive customers and production plans cast in terms of indicators such as numbers of units or volume produced, plants compromised on quality, causing difficulty for customers and downstream industries that depended on their outputs. The El'brus computers in particular experienced frustrating delays in construction and testing due directly to the unreliability of components.

These features of the environment affected all Soviet HPC projects, but not all equally. Features of the HPC technology itself played a significant role in determining the degree to which the hindrances could be overcome. Table 8-1 outlines some of the elements which have differentiated projects and shaped their R&D cycles. Of particular importance was the degree to which the HPC project stretched the technological capabilities of the supporting infrastructure. The projects with some of the longest development times–the El'brus and Elektronika SSBIS computers–were driving forces for the computer industry, forcing the development of a broad spectrum of new constituent and supporting technologies. In the terms of our conceptual framework, the strategy was to develop systems which drove domestic industry, which required building systems for which the technological availability was low.

It was one of the great dilemmas of the Soviet high-performance computing sector that to raise the technological level of the HPC infrastructure as a whole, the end product had to push the boundaries of technological capability of the entire infrastructure. This increased the complexity of R&D enormously–both technically and administratively–and ultimately slowed development. The paradox was that by pushing so many technological fronts simultaneously, the development cycle of a system was significantly stretched out

Environment Relationship with factories (established, long-term vs. weak, short-term) Existence of influential ‘‘champions’’ in industrial, policy making bodies	Technology Level of complexity Degree to which technology stretches capabilities of supporting industries Ease of manufacturing Complete system vs. add-on computational element
	Organizational structure Integrated, rigid organizational structures vs. flexible, autonomous units
Technological availability Nature of components, subsystems (series production vs. prototype) Availability of components (plentiful vs. difficult to acquire) Availability of necessary tools Availability of know-how (depth of experience in building computers)	Beliefs (design principles) Goals (design system for production vs. to demonstrate a concept)
Organizational slack Large-scale vs. small-scale funding Integrated vs. fragmented funding	Strategy Drive supporting industries vs. use existing technologies

Table 8-1 Elements Differentiating Soviet HPC Projects

because of the time needed to develop and acquire the technologies, and the delays involved in integrating a large number of prototype technologies simultaneously. In order to keep pace with the rate of advance in world-wide state-of-the-art, the next generation systems also required large, rather than incremental, advances in the supporting technologies. This was the paradox: the greater the reach forward, the greater the probability of significant delays.

Within the Soviet system as it existed through the mid-1980s, there probably was no good solution to this dilemma. The national security policies of both the East and the West forced HPC developers to rely on inadequate indigenous industries rather than take advantage of developments world-wide. Had the rate of development of supercomputers in the West not been as rapid, the pressure for advance in the Soviet Union might have

been less, and placed less stress on the infrastructure. While this might have made it possible for the supporting technologies to advance in a more continuous fashion, it is likely that given their incentive structures the supporting industries would have reduced their rate of advance to something less than what actually took place, yet still not improved reliability.

Most of the HPC projects discussed in this work did not push the technological capabilities of the infrastructure. With the partial exception of the multiprocessor computing systems with programmable architecture developed in Taganrog, systems developed with the Academy of Sciences and the Ministry of Higher Education used equipment already in series production. The MARS-M used existing El'brus technology, and most of the others were oriented around the technology used to build the ES mainframes.

Although these projects did not suffer delays directly resulting from the need to develop and test new components and supporting technologies, they did suffer from the lack of close ties with industry. The projects were greatly assisted by the efforts of NITsEVT and ITMVT which provided the technology necessary to build prototypes and lent supporting voices within the Ministry of the Radio Industry and policy-making circles. In spite of this, the academic projects experienced delays (of varying severity) through their relationships with the factories assigned to support their work. Although they used components in series production, they also suffered from low technological availability and waged on-going struggles to acquire the necessary components with the necessary quality. As the examples of the MARS-M and ES-270x demonstrate, from the factories' perspective, the construction of a prototype for an academic research institute was of secondary importance behind their main production, especially when the former required resources and facilities which could be applied to other products.

Although the Soviet economic system has been widely criticized for a lack of responsiveness to customer needs, the plans for each economic unit were based on the expressed needs of other economic units, which served as inputs into the planning process. If there was no expressed need for a given product, particularly those serving the military sector, there was little reason to incorporate it into the planning process. Supporting academic projects through to the prototype stage could be justified on the grounds that this served to advance the research of the field. Putting the systems into series production could be justified only if there were customers willing to take the machines, either because of a perceived need or because of higher-level pressure to do so.

The reality was that there were few customers for academic high-performance computers. Although offering high performance on paper, the systems did not deliver the performance advertised, were unbalanced, were too difficult to program, or otherwise did not suit users with real, demanding applications. As we will discuss below, during the reform period, interest in such unproven machines only declined. Furthermore, some of the systems, while built with standard components, were too complex in their construction to be easily incorporated into existing factory production lines. For these reasons, it was not in a factory's interests to assimilate production of the academic machines.

The technology and the guiding principles shaping their development were to a large extent responsible. A primary goal of the academic machines was to demonstrate the viability of novel architectural concepts. For many reasons, including professional integrity and the bias of funding organizations, the development of unique, sometimes radical, systems was imperative. While the industrial ministries sometimes sought to duplicate the efforts of Western computer manufacturers as in the case of the ES and SM families, such ''uninteresting'' efforts would have run counter to the mission of the academic research community and would have a difficult time finding high-level support. A machine was

considered successful if it demonstrated a concept, even if it did not meet the needs of users.

The lack of an industrial orientation was not a failing of academic circles per se. Although attitudes towards the appropriate mix of fundamental and applied research carried out in the Academy of Sciences and Ministry of Higher Education fluctuated over time, there has seldom been doubt that the purpose of these institutions is to advance new ideas and expand the base of knowledge. The academic orientation, however, did little to narrow the gaps between the academic researchers, the industrial facilities necessary to build machines, and the users needed to justify the production of such systems. Not only was the bureaucratic distance between an Academy of Sciences research institute and a Minradioprom factory great, the philosophical distance also discouraged active cooperation. The end result was that while they have made some contributions to the body of HPC research, purely academic projects have contributed little to the Soviet computer base.

High-performance computing research advances most quickly through experimentation. A design's strengths and weaknesses are never fully understood until they have been implemented and the effects of memory volumes and access times, cable lengths, signal propogation times, bus synchronization clocks, interconnect latency, and the myriad of other factors related to a physical implementation have their full impact. Advances are made through constructing prototypes, observing their behavior, and building new prototypes taking into account the lessons learned. Theory and conceptual design are necessary, but cannot fully compensate for a lack of experimentation. Similarly, the effectiveness of systems software and algorithms can only be fully evaluated when executed on a physical machine.

Considerable time and effort were expended in all Soviet HPC projects to acquire the resources and components necessary to construct physical machines. Thanks to the resulting delays, the research cycles of individual projects were extended, and the rate at which the lessons learned from building one machine could be applied to its successor was reduced. We can only speculate about the progress of Soviet HPC had the average development cycle been 2-4 years rather than 5-10. It is likely that new ideas would have been generated and tested more quickly, and unpromising lines of development rejected sooner. The state-of-the-art of the sector would probably have progressed much beyond its current state, both conceptually and in performance.

There are few examples of HPC projects which had the technological characteristics, development philosophy, and environmental conditions to be developed quickly and successfully. Thanks to a pragmatic development strategy oriented towards industry, a construction which lent itself to mass production, the use of existing, available technologies, close ties between the research and series production facilities, and considerable high-level support, the PS-2000 was one of the most successful Soviet HPC projects in terms of the length of the development cycle and the levels of series production. The El'brus-B, while not necessarily a high-end system at the time it was built, similiarly profited from the use of proven technologies, an industrial orientation, a construction of moderate complexity, and the close ties between ITMVT and the Moscow SAM Plant.

On the other hand, the PS-2000 successors demonstrate more of the difficulties characteristic of the majority of Soviet HPC projects. Although developed with the same industrial-orientation and manufacturability as its predecessor, the PS-2100 experienced delays because of the need to develop a new generation of gate-arrays for the processing elements.

8.4 Technological Paradigms and Trajectories

Is there a ‘‘technological paradigm’’ shaping machine architectures throughout the Soviet HPC sector as a whole? As we pointed out in chapter 2, the definition and scope of usage of this term is unclear in the work of Dosi and others. The term is most often applied to a scientific community; is it also applicable at the level of individual projects? If so, what is the relationship between the two? Dosi and others have not addressed the latter question satisfactorily.

At the community level, a paradigm should be founded on an ‘‘‘outlook,’ set of procedures, definition of the ‘relevant’ problems and specific knowledge related to their solution’’ [Dosi82, 148; Dosi84, 14] which predominate within the community. As we examine the set of Soviet HPC projects discussed in this study, we can identify some general characteristics which they share. All of the projects have a design objective of achieving maximal performance within the constraints of other design objectives and the technological base available (or projected to be available). A key direction of advance from one generation to the next is towards higher performance. To attain high performance rates, the overwhelming majority of systems rely on some form of parallelism.

Two features of the developmental environment are shared by all Soviet HPC projects. High performance has been a consistently important parameter to users and sponsors, policy makers who make decisions about project funding, and the broader scientific and technical community in both the Soviet Union and the West. Although peak performance figures are only marginally useful in describing a system’s utility or applicability to particular classes of problems, they are easy to compute and one of the few metrics which can be applied unambiguously to a broad spectrum of machines. In the case of the El’brus-1 and El’brus-2, a slightly more useful Gibson-3 benchmark performance figure is common. In either case, performance is unquestionably the most commonly mentioned

parameter in HPC. Appropriately or no, it has come to symbolize the level of technological advance within the sector.

A second feature of the environment shared by all Soviet projects has been the weakness of the supporting infrastructure and upstream industries relative to their counterparts in the West. For decades, developers have complained about the low quality of components and materials, the low functionality of the available tools, etc. Soviet HPC developers have not been able to build individual processing elements with a performance comparable to the Western state-of-the-art. They have been forced to take alternative approaches to machine architectures. The predominent characteristic of these approaches is that they incorporate parallelism. In theory, a given level of performance can be reached using a single fast processing element, or a greater number of slower elements. The reality is much more complex, but basic deficiencies in the technology available to designers all but forced them to pursue parallelism. Furthermore, parallelism consistently has been viewed as a means of overcoming reliability problems in the underlying hardware through redundancy.[2]

‘‘Achieving high performance and reliability through parallelism and modularity’’ sums up the technological paradigm impacting the Soviet HPC sector as a whole. There are few other goals or elements of machine architecture which are shared by Soviet designers. A dominant characteristic of Soviet HPC is the great diversity of approaches toward achieving parallelism, and the very small number of distinct projects which pursue any given one. Table 8-2 reviews the broad spectrum of architectures built within the So-

[2] The reality is that the goals of high performance and high reliability are not likely to be achieved simultaneously when the underlying component base is unreliable. Larry Snyder makes this point well: ‘‘It is often erroneously thought that connecting multiple copies of unreliable components together can achieve both reliability and performance. Though reliability *may* be achieved, that is, some parts *may* be functional at a give time, high performance is achieved only when all parts are functional all of the time. Thus, improved performance is achieved only when all parts are functional all of the time’’ [Dong92b, III-3].

Machine	Type of architecture
ES-2700	Attached array processor
ES-2701	Macro-pipeline/coarse-grain compositional language
ES-2703	Programmable architecture multiprocessor
ES-2704	Reduction/dataflow
ES-2705	Analog multiprocessor
El'brus-1,-2	Shared-memory multiprocessor / Stack-based
El'brus-3	Shared-memory multiprocessor / VLIW
MKP	Shared-memory multiprocessor / pipeline
Elektronika-SSBIS	Vector-pipelined
PS-2000	SIMD
Sibir'	Loosely-coupled array processor system
ES-1191	Mainframe host with vector processors

Table 8-2 Spectrum of Architectural Approaches in Soviet HPC

viet HPC community. While policy makers in the Soviet centralized, command economy made decisions about which projects to support, the technical features of Soviet HPC were not generally determined centrally. Although customers had requirements that had to be met by the HPC developers, the specific design was almost always determined, or at least suggested, by the researchers themselves. The systems reflect not only user requirements, but also the design philosophies of individual researchers.

Although the community as a whole has taken widely divergent approaches to achieving the goal of high performance through parallelism, we can find design elements which are shared by subsets of projects. For example, the El'brus and Elektronika SSBIS rely on shared memory; most of the academic projects including the Homogeneous Computing

Systems (OVS), dynamic architecture machines (MDA), and macro-pipeline processor systems use distributed memory. The Multiprocessor Computing Systems with Programmable Architecture (MCS PA) use a hybrid shared-distributed memory system. The PS-2x00 line also is a distributed memory system. The PS- series, OVS, and MCS PA in particular view reconfigurability as a promising means of improving the match between architectures and algorithms.

Other issues create different groupings. The OVS, MCS PA, El'brus, and PS- series use homogeneous processing elements. The MDA and macro-pipeline systems have homogeneous computational elements, but incorporate other types of processing elements for control functions. During Burtsev's tenure at ITMVT, the philosophy of incorporating a variety of special-purpose processors into an El'brus-2 configuration was pursued actively. The MARS-M and Sibir' projects are two extremely different systems which incorporate heterogeneous processing elements.

Soviet HPC can also be categorized by whether they operate in a single-instruction, multiple-date (SIMD) or multiple-instruction, multiple-date (MIMD) mode, the nature of the interconnect system, whether or not the system is designed to be attached to a general-purpose mainframe host, the use of horizontal architectures, the reliance on high-level language constructs, etc. Machines which are grouped together by one criterion are frequently not grouped together by another. In short, with regard to machine architecture, it is difficult to find design principles, or groups of principles, which are adhered to by the HPC sector as a whole. In the case of Soviet HPC, the number of principles shared by signficiant subsets is not large. In short, beyond achieving high performance and reliability through the use of parallelism and modularity, there does not appear to be a clearly identifiable paradigm for the Soviet HPC sector.

Although Soviet HPC projects differ greatly, they exhibit a great deal of internal design consistency from one generation to the next. In our study we have seen a striking continuity of many architectural approaches in the machines for which multiple generations have been built: the El'brus, the PS-2x00, the dynamic architecture systems, the multiprocessor computing systems with programmable architecture, etc. Each of these families have followed a technological trajectory characterized by continuity, rather than discontinuity. In fact, it is perhaps surprising how stable the technological trajectory has been during the reform period. The architecture of the El'brus-3,[3] the PS-2100, the successors to the ES-2703 and ES-2704 share the dominant characteristics of their predecessors and have few features which reflect the growing turmoil of the surrounding social, economic, and political systems.

We have discussed factors contributing to the stability of the technological trajectory. Each of these families was built under the influence of a stable set of users and user requirements and/or a set of design principles held very strongly by the main engineers. In the case of machines developed for specific customers, the selection environment– strongly shaped by the requirements of those users–remained stable, at least until the breakup of the Soviet Union and in some cases (the El'brus-3 in particular) later.

The technology itself played an important stablizing role in several different ways. First, requirements for compatibility forced a new machine to share many basic features with its predecessor. Second, the longer a machine had been under development, the more costly in time and money it would be to radically alter the design. The El'brus-3, for example, was initiated around 1984. By 1990 the machine had been rather completely designed and arrangements had been made with the supporting industries to provide the necessary components and subsystems. Altering the design at this point would have been

[3] We discuss the shift from stack-based to VLIW in the El'brus line below.

costly. Third, if an architectural approach was basically satisfactory yet allowed room for improvement, designers had little incentive to make the effort to master a dramatically new architectural approach.

From our study we can see that in the case of Soviet HPC, it is much easier to identify a meaningful technological trajectory and associated paradigm and selection environment for individual projects than for a community as a whole. What is viewed as an element of consistency within one line of development may not be shared by any other projects, and thus not an element of a more broadly held paradigm. In short, there are important elements of intra-project consistency which are not explainable by identifying a paradigm in effect for an entire community.

The case of the El'brus series gives us further insight into the nature of technological trajectories and paradigms. In chapter 4 we saw how the El'brus-1, -2, and -3 lie along a technological trajectory which has been quite consistent for over 20 years. In each generation, designers sought to increase performance through some combination of faster and improved components, reduced clock periods, greater volumes of primary and secondary storage, greater numbers of processors and functional units within processors, and improved processor architcture. Basic systems characteristics–coarse-grain parallelism through a moderate number of powerful processors with shared main memory, modularity, multiprocessing, independent I/O and data transmission processors, hardware support for high-level language constructions, software compatibility with previous generations–remained very similar throughout the generations.

One of the few points of sharp discontinuity in the technological trajectory was the design of the individual processors, as the stack-based architecture of the El'brus-1, -2 processors was replaced by a VLIW approach in the El'brus-3.

Did a paradigm shift take place? How much change is needed in a paradigm before one can say that a shift has taken place? These are perhaps the wrong questions. They assume that a paradigm, even at the level of a single product line, is a single, indivisible entity. The concept of a ‘paradigm’ must be modified to account for the fact that within a given project, certain elements can remain quite constant as the technology develops, while others can change dramatically.

A more useful way to view a paradigm is as a series of layers of finer-grained ‘‘‘models’ and ‘patterns’ of solution...’’ which cover the spectrum of technological problems to be addressed during the development of a complete system. We can refer to such layers as micro-paradigms. The micro-paradigms guide developers’ decisions about specific subsystems or parts of the complete system. The history of the El’brus series suggests that some micro-paradigms can shift dramatically without necessarily causing a shift in others. The shift to a VLIW approach with static scheduling did not cause major changes in the principles of modularity, coarse-grain parallelism, shared-memory, etc.

Because the micro-paradigms affect the development of portions of a complete system, they are in practice not independent. Unless subsystems function well together (or, in instances where inter-generational compatibility is a requirement), the computer will be unbalanced and not deliver the performance and functionality desired. There are hundreds of examples. A memory system which is slow, or has certain bottlenecks will not be able to provide a fast processing element with data rapidly enough to avoid excessive idle time. A lack of software compatibility between generations will require extensive recoding by software developers and users. A cooling system must be capable of dissipating the heat generated by the components. Similarly, decisions to alter one subsystem dramatically may force dramatic changes in other subsystems. For example, had El’brus designers determined that a fine-grained, massively parallel, distributed memory system

held the best promise for the future, most of the earlier ideas about processor architecture, memory structure, interconnect systems, etc.would have had to change significantly.

An important issue for technological innovation and advance is the conditions under which paradigms and micro-paradigms can change. Our study of Soviet high-performance computing indicates that several factors are necessary. First, there must be a mis-match, or growing incompatibility between the technological approach being taken and the selection environment which indicates which kinds of technology are acceptable. This can occur through a qualitative change in the requirements for a system, such as those specified by principal customers, or because a given technological approach is not able to meet the goals of quantitatively changing requirements. In the case of Soviet high-performance computing through the period covered by this study, there was little qualitative change in requirements of the principal sponsors, leading to a high degree of consistency in the technological trajectories of most HPC projects. The nature of the target applications changed little. The principal changes were quantitative--increased performance, improved functionality and reliability, and so forth.

Some systems, like the PS-2x00 series, could meet the requirements adequately through quantitative means. The essential architecture of the individual base modules differed little from that of the PS-2000. Advances were mostly extensional: uniting multiple base modules and providing for their interaction, increasing the amount of main and peripheral storage, increasing the word length, using an improved component base, etc.

Other systems, like the El’brus-3, could not meet the requirements solely through incremental extensions of existing approaches. The nature of the basic requirements had not changed, but existing technological approaches to the design of the processors could not meet the high performance demands.

Two other factors strongly affecting whether or not a shift in a paradigm or micro-paradigm occurs are the scope of impact of changes to the technology and the scope of decision-making needed to effect the change. The scope of impact and scope of decision-making are intimately related to the nature of the developmental environment and organizational structure, and the technology and associated paradigms. The scope of impact refers to the degree of coupling between elements of a technology. The broader the scope of impact, the greater the changes that need to be made in the systems which interact with the specific technology. The greater the scope of decision-making, i.e. the numbers and types of individuals and organizations involved in the decision-making process, the more difficult it is likely to be to make a decision for change.

The lower the scope of impact and scope of decision-making, the easier it will be to implement a change. In the case of the El'brus-3, the change from a stack-based to VLIW architecture had a relatively limited scope of impact and scope of decision-making. The scope of impact was limited by a series of well defined interfaces which insulated the processor from the surrounding systems, both technological and social. For example, the high-level programming languages defined the interface between applications and systems software and the underlying hardware. The compilers were the only pieces of software which directly reflected the underlying processor architecture. Hence, changes in the processor architecture required changes to the compilers, but not to existing systems or applications software. Implementing this change also required the decisions and cooperation of a relatively limited group of individuals, primarily those within the El'brus development team and selected individuals in the ITMVT and ministry hierarchy.

In contrast, similarly drastic changes to the PS-2x00 processor architectures would almost certainly have required changes to the assembler level language in which much ap-

plications software was written. Large bodies of existing software (and hence users) would have been impacted. The social cost of such a change would have been high.

In principle, Academic systems without a user community have greater feedom to make drastic changes to the architecture. They may, however, lack to financial and human resources to make changes to a many aspects of a design simultaneously.

The strategies of minimizing the scope of impact and decision-making have proved to be extremely powerful facilitators of technological advance in Western development as well. Network designers have employed layered approaches to protocol stacks as a means of reducing complexity and allowing individual vendors considerable freedom to alter implementation details of any given protocol layer. A similar strategy lies at the core of recent trends towards open systems. Open systems can be viewed as a collection of ‘‘black boxes’’ with well defined interfaces between them. As long as a ‘‘black box’’ adheres to the necessary interface standards, the internal implementation can be varied easily.

A fourth factor influencing paradigm shifts is the availability of ideas about alternative directions of advance and examples of successful implementation. In particular, we have discussed how the ILLIAC-IV, Burroughs 700 Series, FPS attached array processors with horizontal architectures, and Cray supercomputers served as sources of inspiration for the PS-2000, El’brus-1 and -2, El’brus-3, MKP, MARS-M, and Elektronika SS-BIS respectively. In each case, the Western models shaped Soviet developments in two ways. First, they demonstrated particular architectural approaches. Second, and perhaps more importantly, they showed that these architectural approaches were basically viable. The latter factor gave development teams much of the confidence they needed to pursue implementations, and, in some cases, proved valuable in obtaining higher-level approval. It is certainly not the case that all ideas found in Soviet HPC computers were inspired by

Western developments. Many ideas and their implementations are indigenous. Nevertheless, our study indicates that the availability of Western ideas has been a powerful catalyst for advance in Soviet high-performance computing.

A fifth factor is the feasibility of implementation. Feasibility is closely related to the scope of impact discussed above, but is also a function of cost, time to development, and other factors. Projects like the PS-2000 and academy projects with modest resources which stayed within the bounds of what the domestic industry could manufacture experienced greater feasibility constraints than did those, like the El'brus, which were designed to push technological boundaries.

Finally, a more human element is the tenacity with which principal designers hold on to their beliefs about the direction of advance. This is particularly evident in the work of NIIMVS in which there is only slight variation from one project to another in basic design philosophies of processor architecture, interconnect systems, etc. V. A. Kalyayev has exerted strong control over the research agenda and permitted little experimentation outside the framework of the established paradigm.

The factors we have just discussed play an important role in facilitating or hindering a paradigm shift. Our list is not exhaustive, particularly if we wish to generalize the discussion to other types of technologies. Studies of computing technologies more broadly, such as those by Kling [Klin82; Kling84] make it clear that for technologies which are socially complex, systems are more likely to evolve as a by-product of technological and social factors rather than through strictly rational decision-making. Although not socially simple, Soviet high-performance computers are not as socially complex as other systems, such as the management information systems discussed by McHenry [Mche85]. Characteristics of the technology and the immediate infrastructure necessary for their develop-

ment have considerble explanatory power in helping us understand the nature of their technological advance.

While it is beyond the scope of this dissertation to predict conclusively under what combinations of these factors a paradigm shift will occur, we are able to point to a number of changes in these factors which could very well lead to significant paradigm shifts in the future. We discuss these in our final chapter.

8.5 The Impact of the Reform Process on Organizational Structure

We have discussed how changes in legislation regarding state enterprises and associations, small enterprises, joint ventures, and cooperatives have dramatically altered the opportunities and mechanisms for organizational change. An important outcome of the reform process has been a decentralization of authority and responsibility for an organization's structure and domain(s) of activity. How have the organizational structures changed, and why? What has been their impact on the development of high-performance computers?

Throughout the former Soviet Union, there has been a pronounced trend towards the fragmentation of organizational structures at all levels of society. The role of the ministries and organizations such as the Academy of Sciences in the lives of their subordinate institutes has decreased dramatically. The State Committee on Science and Technology (later absorbed into the Russian Ministry of Science, Higher Education, and Technology Policy) has retained some influence through its funding practices, but lost much administrative ability to control S&T throughout the economy. The government is still the owner of land, buildings, and much capital equipment of most organizations, but has minimal direct influence in day-to-day decisions about the activities and structure of enterprises and institutes.

Within the institutes we have discussed, we have witnessed a transformation away from a unified hierarchy toward a loosely-coupled collection of smaller-scale organizations with greater autonomy. This trend has taken place more completely in some organizations than in others. ITMVT and NIIMVS maintain a hybrid structure of an integrated hierarchical core with a number of associated but autonomous organizations existing within the shell of the institute as a whole. NIIUVM consists almost entirely of financially independent ‘‘rental collectives.’’ ISI is currently a collection of laboratories, small enterprises, etc. pursuing research or contract opportunities independently of each other with little coordination from central administrators. Similar patterns are, we believe, taking place at the other R&D facilities mentioned in this study.

The transformation of organizational structure reflects the balance between forces for decentralization (driven by efforts increase the organizational slack and achieve greater autonomy) and centralization (driven by the desire to perserve the ability to conduct HPC R&D). In the cases we have studied, the former has dominated the latter.

The changes in legislature allowing alternative organizational forms and local decision-making have been a powerful enabling factor. Information about successful incorporation of new forms at other institutes often encouraged leaders at the institutes we have studied to make similar changes. The principal factor driving the changes, however, was a desire to increase ‘‘organizational slack,’’ or the level of resources at the disposal of the organization. The latter could be accomplished by working more efficiently, by generating additional revenues, and by converting existing resources into a more flexible, useful form, i.e. the conversion of accounting rubles *(beznalichnyye)* into cash *(nalichnyye).* A basic objective has been to find a way to retain the engineers who constitute the core of the institutes’ technical capability and keep them from seeking employment else-

where. Money for wages had to be generated, and restrictions on wage levels had to be skirted.

The creation of cooperatives, small enterprises, rental collectives, and temporary collectives served each of these purposes. They provided a way to get around legal restrictions on wage levels (or, more precisely, the amount of money available to pay wages), to enter into contracts with negotiated (i.e. higher) prices, to convert accounting rubles into cash, and bring together individuals best suited to carry out a particular task in an efficient, timely manner.

Our study indicates that the nature of the revenue stream and the opportunities for alternative organizational forms have a significant influence on organizational structure. In each of the organizations in our core cases a reduction and fragmentation of income has led to a fragmentation of organizational structure. We will discuss the changing nature of income streams and its implications for technology in the next section.

8.6 The Impact of Reform on the Development of HPC

The reform process initiated in 1985 by Mikhail Gorbachev has followed a complex, uncertain, largely uncontrolled path which has fundamentally changed most facets of Soviet economic, political, and social life. The goal of ‘‘democratic centralism,’’ of greater autonomy and use of economic mechanisms at the local level coupled with more comprehensive coordination by central government organizations has largely unravelled; the forces of decentralization and fragmentation have overwhelmed the forces seeking to improve centralization in both the political and economic spheres.

The reforms have brought about some changes which are likely to have a positive impact on innovation in Soviet HPC over the longer term. The administrative-directive form of economic management has to a significant degree been replaced by one based on eco-

nomic considerations. Enterprises have achieved the autonomy over transactions with other enterprises and organizations which Berliner called for [Berl76, 522]. The quality of feedback between suppliers and customers has increased as suppliers, facing declining markets and excess capacity, have been forced to court customers. The customers, thanks to their own weakening financial condition and control over finances, have become more demanding. The sensitivity to customers' needs has increased at factories and at research institutes whose ''products'' are pieces of research or technology development carried out under contract. In addition, in the case of ITMVT and NIIUVM, the research institutes have assumed much of the burden of marketing the high-performance systems. The creation of the Supercomputer Association, its composition, and the grass-roots nature of its operation are further indications of the strengthing of the customer feedback loop. An important feature of current inter-organizational transactions is that they are based on satisfying the requirements of customers rather than bureaucratic watchdogs who monitor planning indicators and procedures.

The reforms have brought a new flexibility to the organization of R&D. We have witnessed the creation of development teams and organizations which draw members from a variety of existing organizations to address specific tasks.

The reforms have created opportunities for expanded contacts with the international community, opening the doors for better professional interaction, foreign investment, and/or contracts for work or products. Joint efforts such as that between Sun Microsystems and ITMVT (Moscow SPARC Center) would have been unthinkable a decade ago. Although many aspects of technology transfer between Soviet and Western counterparts remain subject to export control restrictions, arrangements like these offer Soviet scientists the hope of accessing Western capital, technology, and know-how and the opportunity to observe Western practice more closely.

These changes have the potential for improving some critical parts of the innovation process. Idea generation can be improved through increased interaction between suppliers and demanding customers and idea ‘‘cross-pollination’’ across organizational and international boundaries. Gathering support for an idea can profit from the potential access to foreign funding sources and more open communication between individuals in different organizations (although the latter is partly offset by growing possessiveness over ideas of possible commercial value). The idea implementation process can be improved through the use of the more flexible organizational forms, the growing willingness of factories to manufacture products which they can sell, and the reduction of bureaucratic overhead in the development process.

These positive elements currently exist more as opportunities than as realities, however. While they will undoubtedly have a significant positive impact on innovation in the future, they are overshadowed by the negative consequences of the reform which have seriously undermined Soviet ability to carry out large-scaled R&D in advanced technologies. First, the landscape is dominated by the desperate state of the economy and a nonexistent market for Soviet HPC. Second, fundamental weaknesses remain in the Soviet infrastructure. Unless the economy improves dramatically and the structural weaknesses are corrected, there will be only limited, localized benefit from the improvements listed above; there will be no Soviet high-performance computing sector of any consequence.

8.6.1 Economic Considerations

The real demand for expensive, powerful systems at present is very low. The financial state of current and potential HPC users is so poor that few are able to acquire them. The ballooning federal budget deficits coupled with a policy of reducing the portion of production carried out directly for the state has caused the volume of state orders to de-

crease dramatically. Series production of Soviet HPC systems ground to a halt by the end of 1992.

The depressed economy has prevented a market for Western high-performance systems from growing as well. Western vendors with permission to sell to the former Soviet Union have found the market virtually non-existent. A Convex executive responsible for sales in Russia has stated that in spite of a number of inquiries from Russian organizations, ‘‘we haven’t seen any money yet, so they haven’t seen any computers’’ [Huds92].[4] A healthy HPC sector cannot exist without a market.

In cases where users are able to afford large-scale systems, available Soviet machines face increasingly fierce competition from Western models. Current CoCom regulations still shelter the market for high-end Soviet HPC systems such as large configuration El’brus-2s, the Elektronika SSBIS and the MKP, at least in terms of performance levels. Mid-range workstations and mainframes can be imported with few restrictions, but larger systems are either categorically prohibited, or are sold with cumbersome restrictions. In practice, users who have hundreds of thousands or millions of dollars available to spend on computing equipment are attracted by Western mid-range systems and workstations which are highly functional, reliable, and can sit beside an engineers desk. Given the rapid advances in technology,[5] the number (hundreds of thousands) of units manufactured, and the pressure to acknowledge global trends in computer technology evolution and relax export control restrictions, Western workstations will provide a viable alternative to indigenous systems for most Soviet HPC users. This will be a truism if, as appears to be case, production of the older generation of Soviet HPC systems (the El’brus-2 and

[4] Convex recently installed its first (legal) unit in Russia at the Joint Institute of Nuclear Research in Dubna, Russia [Hpcw930628].

[5] Witness the recent introduction of machines based on the Alpha microprocessor which compete favorably with Western mainframes or even supercomputers. In a recent test, an Alpha AXP system performed a sort benchmark six times faster than the Cray YMP which set the record in 1992 [Hpcw930405].

PS-2100) has ended before successor systems reach volume production. Series production of high-end systems has all but ended, yet demand for the new generation of systems is too weak to support series production of systems like the MKP and Elektronika SSBIS which have reached the prototype stage. By the time the economy recovers to the stage where potential HPC users have the funds necessary to acquire large-scale systems, the current models will be obsolete, especially in comparison with Western models which will continue their rapid rate of technological advance for the forseeable future.

8.6.2 Structural Considerations

Even if the economy improves, Soviet HPC developers face a number of structural weaknesses which fundamentally compromise their ability to develop new machines over the long term. We have mentioned how the reforms have brought increased flexility to organizational structure and inter-organizational ties. They have also brought fragmentation, or the breakdown of ties which we discuss in this section.

A major structural challenge facing R&D facilities is the nature of the indigenous infrastructure supporting HPC development. Large-scale development still requires the participation of hundreds of upstream organizations. As a consequence of the reforms, these links are now, for the most part, not administrative (vertical), but economic (horizontal). Their existence depends on whether or not the upstream organizations feel that it is in their best interests to partipate in the development or production of a particular good, and their ability to deliver. Thanks to declining production levels throughout the economy and the creation of a fair amount of idle production capacity, organizations are more willing now than previously to consider new orders. At the same time, they are not likely to take on orders which will not be profitable for them, as might well be the case for highly specialized pieces of advanced technology for which the market is limited. Even if they

wish to engage in such work, under current conditions acquiring the necessary inputs of the necessary quality is quite difficult.

More fundamentally, the Soviet economy still has a highly monopolistic nature for many products. While the number of factories making consumer electronics has mushroomed, the number of facilities able to carry out advanced microelectronics R&D, for example, has remained constant or declined. The indigenous infrastructure for Soviet HPC still lacks redundancy. As a result, the problems of delay and quality which result from non-competitive, monopolistic industrial organization will continue to plague Soviet HPC developers and overshadow some of the benefits of more flexible inter-organizational ties. We will discuss possible alternatives to this situation in the next chapter.

A second structural weakness is the administrative and financial gap between research and production facilities. Soviet policy-makers have tried for many years to bridge the gap between research and production, chiefly through scientific production associations (NPO) and research clusters such as in *Akademgorodok* in Novosibirsk which encourage R&D to be carried through the combined efforts of Academy and industry organizations. At no time have the component entities of the HPC sector been linked so that a portion of the profit realized through series production was poured directly back into R&D for the development of the next generation of products.

In the cases of NPO Impul's and ITMVT, funding for development of high-performance systems came through state funding (from the state budget or principal sponsors) ear-marked for the development of a specific system. These funds were used to support R&D and production in upstream industries as well. When a prototype was completed, factory documentation was turned over to the series production facility which built units in response to a production Plan, or individual orders from customers. Pro-

ceeds of series production were not channeled back into the R&D facility. The development of new generations of systems had to be supported by specific funding from government or wealthy customers.

The forces of decentralization unleashed by the reform efforts penetrated below the boundaries of the NPO. Factories became administratively and financially independent of the research facilities. The former, themselves struggling for solvency, had little inclination and few resources to support the R&D of advanced technology in a different organization. To the extent that research funds were available, they were redirected towards goods which would cost little to develop and which would enjoy an immediate market. Such goods, like telephones and washing machines, were developed inside the factory.

V. V. Rezanov, deputy-director of NPO Impul's calls this situation ''a hole in *perestroika*.'' One of the goals of perestroika was to incorporate decentralized mechanisms in the management of the economy. However, decentralization penetrated below the boundaries of the NPO to the level of individual institutes and factories and even to the level of divisions and laboratories within institutes. At a time when prospects for continued government support for R&D projects were becoming very uncertain, the R&D facilities had less opportunity to profit from the manufacture of systems which they had developed.

It is absolutely essential for R&D facilities in the Soviet HPC sector to find an arrangement in which the research and production facilities are tightly integrated, both financially and administratively–in effect combined into one organization–so the organization can generate revenue through the sales of series production goods, which can be used to fund in-house development of new products. The alternative is for R&D always to be funded directly from the government budget, or by customers willing to fund a large portion of the entire R&D bill themselves. In the first case, the link between customer prefer-

ences and R&D efforts will be weak and/or not lead to commercially viable products, and administrative barriers between research and production will remain problematic. Under the new economic conditions, it is unlikely that customers or investors will be willing to fund massive R&D efforts unless they can be assured of recovering their investment. The number of such customers is already extremely limited, and the prospects for long-term, adequate, stable funding through them are not good.

A third structural problem is that the fragmentation of organizational structures will impact Soviet ability to conduct R&D on large-scale projects. We have discussed the fact that development of large-scale projects depends on an appropriately extensive and integrated organizational structure to provide the direction and coordination necessary to build a functional system. Such a structure need not be rigid, but mechanisms must be in place to enable a variety of organizational units which complement each other to work together towards a common goal. While a moderate reduction in the income stream helped move organizations towards more flexible structures, fragmentation resulted from a drastic decrease in and fracturing of the income stream. As funding levels for HPC R&D decreased in relative terms, institutes were left with few alternatives but to find other sources of funding, mostly through contract work or, in rare instances, through joint efforts with Western companies. In the desperate economic climate, such contracts were, as a rule, small scale and short term. Whether smaller contracts caused smaller organization units to form, or smaller organizational units sought out contracts which matched their capabilities can be debated. The influence probably ran in both directions. Either way, a rough equivalence developed. Given the difficulty in securing large-scale contracts, a strategy adopted by the leadership in nearly all the institutes we have examined was to distribute the burden of finding alternative funding sources, placing much of the responsibility for survival on the shoulders of the small autonomous units themselves.

Contract work tends to lead to a fragmented income stream. First, payment is very closely tied to the execution of specific tasks. Unless the profit margins are quite high, such work generates a minimal amount of slack resources which can be applied to other development projects. Although in principle the cost of a contract is negotiated between customer and provider, the poor economy and past practice under the Soviet system tend to keep profit margins low. Customers with little money to begin with and who are used to contract values' equaling the cost of doing the work plus a small margin are reluctant to grant providers large profits. Second, contract work tends to be customized for the consumer and does not result in a product which can be broadly marketed. Furthermore, as a function of the poor financial state of most organizations in the depressed economy, year-to-year or even month-to-month budgeting, and the multiplicity of customers, individual contracts currently are often relatively small, supporting the work of a team of engineers for a relatively short period of time. Such projects do not generate sufficient income to support the development of large-scale high-performance projects. In other words, contract-based work has limited potential for generating the volume and type of income needed to support large-scale R&D in the future.

A necessary, although not sufficient, condition for conducting large-scale R&D is a unified flow of income sufficient to meet the costs. We have mentioned possible sources: government funding, support from individual, principal customers, production and sales of goods which generate revenue.

It is an unfortunate paradox that the measures needed to preserve capability and keep development teams employed–the use of more autonomous organizational structures–threatened to fragment the structure necessary for large-scale development. It can be argued that if funding is restored the resources needed to support such a structure will again be available. This is probably so, but the ease with which Humpty-Dumpty can be put

back together is also a function of the length of time the smaller organizational units evolve autonomously and of the diversity of their directions. Over time, they will probably drift apart in terms of shared technology, shared research goals, and possibly shared culture. The ability to conduct large-scale HPC R&D will be compromised.

The preservation of some semblance of an integrated structure has depended on the persistence of the directorate, and the level of funding available to support a basic level of research on advanced projects. ITMVT, for example, continues to receive funding to finish existing projects and support the core R&D teams. Funding for the PS-2300 had, at the time of this writing, dried up completely, and the development team was returning to the institute's traditional emphasis on control systems. Itenberg's rental collective retained some ties to other organizations within NIIUVM thanks to a small, but existing market for the institute's control systems. The ISI divisions and laboratories are carrying out indpendent research and HPC development has ended.

CHAPTER 9. WHAT IS TO BE DONE?

If the Soviet states are to occupy leading positions in science and the international advanced technology industries, their scientists and engineers must have access to high-performance computing far beyond what they now have. Functionality, cost, and time-to-market are critical competitive variables. Western automotive, aerospace, computing, and other industries rely heavily on high-performance computers to improve each of them. Widespread access to powerful computing facilities by the scientific community is today a prerequisite for broad-based advance in many spheres of research.

To date, the Soviet high-performance computing sector has not delivered nearly the computing resources needed by science and industry. Furthermore, the sector is at a critical stage, fighting for survival. Production of high-performance systems has all but stopped. There are many potential users for HPC, but few who are willing and able to purchase HPC systems, particularly indigenous ones. Competition from Western computers is increasing. Funding continues for major industrial projects nearing completion, but at levels which are inadequate to do much more than keep development teams together. Funding for new large-scale systems has not materialized. The leading developers of HPC, ITMVT and NIIUVM, are struggling to maintain an organizational structure and expertise which can continue HPC development in the future. Under these conditions, Can Soviet HPC survive? If so, in what form? What are some options for giving the scientific and industrial computing communities access to needed high-performance computational resources?

9.1 Implications for HPC developers

Without funding, there will be no Soviet HPC sector. Without a market, it will only struggle, and not become a world-class player. There are three basic sources of revenue: direct government funding, revenues generated from internal activities, i.e., the sale of

finished products, product re-sale, contract work, etc., and foreign or domestic private investment. Until the economy as a whole improves to the point where the government and industry have resources which could be devoted to high-performance computers, the opportunities for funding to support the development of complete large-scale systems are very limited.

The government is currently not willing to fund new projects on a large scale; industry does not have the resources (or is not willing to spend them) to purchase Soviet high-performance computers, let alone fund expensive R&D programs. The R&D facilities themselves are ill-equipped to earn revenue from the sale of finished products manufactured by factories. We have discussed how contract work is not likely to generate large income streams. Foreign investment such as that provided by Sun Microsystems will serve a useful purpose in keeping Soviet engineers engaged in work which will hone and advance their skills. However, foreign firms investing in Soviet R&D will be interested in research results or the development of portions of systems, and not the development of complete systems. They may provide tools (e.g. workstations and CAD systems) which are useful for other development, but funding for other projects will have to come from elsewhere. Income may be generated through strictly commercial activities if R&D facilities depart from their traditional domains and serve as re-sellers of Western equipment. Even if this is highly profitable, how willing are organizations going to be to invest the proceeds in high-performance computing development if there is no market?

A consequence of decreasing revenue stream and more highly focused small scale funding is that the HPC sector will not be able to carry out large-scale development projects which support R&D across the entire infrastructure of upstream industries. The types of projects undertaken in the future are likely to smaller scale, relying less on new developments in other industries or industrial sectors.

To be healthy, the Soviet HPC sector must have a market. Obviously, a market provides sources of revenue and should provide a demanding customer base which helps shape the direction of technological advance. If the economy does improve, or national priorities shift, it is possible that the government will fund Soviet HPC at higher levels than it does now. Such funding by no means guarantees that the end results will be competitive on either the foreign or domestic markets, however. As a principal customer, the government has helped individual companies survive; but direct government funding has not had a good track record of producing commercially viable products, either in the West or in the Soviet Union. If Soviet high-performance computers are not competitive, the market will be limited to those organizations which, for foreign and domestic national security reasons, are forbidden to use non-domestic advanced technologies. Such a market is smaller now than it once was, and will not lead to a vibrant HPC sector.

Regardless of the level of government funding, the Soviet HPC sector will, for its own health, have to try to develop a commercial presence. To do this, R&D and production must be united so that the proceeds from the sale of finished products can be reinvested in research and development. This will not be accomplished easily, however. As decentralization and privatization progress, providing R&D facilities with large-scale production facilities will have to be accomplished by acquisition, construction, or strategic alliance. The first two require funds currently unavailable; the latter is unlikely in light of the poor market for HPC products. It would be easier for factories to hire individuals from existing R&D institutes, but this will only happen if the factories determine that HPC is a better investment than other, less advanced, technologies.

Assuming the economy improves and funding from some quarter is available for R&D, how can Soviet HPC become competitive? By their nature, high-performance computers of all types and sizes depend on a large number of highly sophisticated up-

stream industries. Although some of the component products may be considered ubiquitous, such as certain types of microprocessors in the West, they still require advanced R&D and manufacturing capabilities.

Our study has shown how strongly the nature of the Soviet economic system and the technologies available from upstream industries have influenced Soviet HPC. Although in some respects it is currently easier for HPC developers to interact with other organizations and suppliers, the traditional infrastructure is not yet well suited to support HPC development.

It is critical that the Soviet HPC sector reduce its dependence on indigenous industries. Although cultivating a complete infrastructure for HPC was necessary in the past, the Soviet HPC sector cannot afford to serve as the principal driving force for upstream industries. It also must take advantage of advances in computer technologies as they occur elsewhere in the world. This is not to say that current efforts to keep leading teams in all facets of computer technology employed should be terminated or that the HPC sector should not use indigenous products when they are of acceptable functionality, quality, and quantity. Such efforts have a role in maintaining expertise and should be continued. It is the case, however, that if the HPC sector is forced to rely on inadequate or immature domestic technologies, the resulting HPC system will have long development times and will not be internationally competitive. Developers, in industry and academia, will not be able to enjoy the benefits of rapid iterations in computer design and development. The HPC sector cannot afford to wait for the domestic industries to de-monopolize and rise to world-class levels.

Incorporating Western technology is no panacea either. Most advanced systems require at least some customized components. Establishing the necessary business relations with Western manufactures will be problematic for financial, geographic, and political

reasons, although less so now than during the Cold War era. Even acquiring off-the-shelf components requires much more effort and expense for a Moscow-based organization than one located in Silicon Valley. We have discussed the additional complications for organizations such as NIIUVM in Ukraine who have only indirect access through Russia to the Western currency needed to purchase Western goods.

The basic point, however, is that the greater the use of advanced, off-the-shelf technology–either Soviet or Western–the easier it will be to build advanced systems in a timely, competitive manner. This will be the easiest way for the Soviet HPC sector to ride the wave of technological advance in the international community.

Such a strategy has implications for the types of systems developed. One cannot build a system directly comparable to a Cray using off-the-shelf components. Soviet industry will probably have to relinquish the objective of competing with such machines for the time being. Recent Western experience shows that high performance can be obtained through high levels of parallelism using non-exotic technologies. For example, Intel's Supercomputer Systems Division has built multiple generations of systems based on off-the-shelf Intel processors such as the 286/287, 386/386, and i860 microprocessors. Furthermore, Intel has tried to incorporate commercial technologies to the highest degree possible, using commodity CMOS memory components, the 5.25" hard-disks used in most workstations, as well as widely used I/O, networking, operating systems, and computer language standards and technologies. Transputer-based systems can also be constructed from readily available technology. Transputers, communicating via built-in serial links and running widely available software, can be configured into multiprocessor configurations relatively easily, as demonstrated by the many value-added resellers of transputers throughout the world.

The reasoning driving the development of massively parallel HPC in the West is essentially the same as that underlying a key element of the Soviet HPC paradigm: parallel processing is a viable route to high performance. For this reason, developing massively parallel systems based on Western technology would not be a severe philosophical departure for many Soviet groups involved in HPC. Unlike those of the Soviet industry, however, the components used in Western parallel systems, even though commercial, are very close to the state-of-the-art.

Using existing technology will have a significant impact on system design. No longer will developers have the freedom to create exotic processor architectures or memory modules. Many issues of interconnect structure and construction will be determined, or at least bounded, by the nature of the technology available. Should this strategy be pursued, it is likely that we will see a confluence of the approaches taken to hardware design.

The major challenge in designing systems along these lines is not the hardware, however, but the systems and applications software needed to use it effectively. Companies like Intel have spent millions of dollars and hundreds of man-years researching the most appropriate way of managing system resources, taking advantage of the computing power the hardware offers, decreasing software development time, and providing computational results in a useful form. Without sophisticated (and often proprietary) systems software, the hardware is all but useless.

Fortunately, software development plays to Soviet strengths (or, plays less to their weaknesses) than hardware development. Although software development does require a basic parallel hardware platform to be tested realistically, it does not require nearly the network of supporting industries and products that hardware development does. Adequate software development platforms are readily available for commercial personal computers

and workstation. A great deal of development and simulation can be done by engineers with these tools alone.

The effort required to develop software to make the hardware discussed above functional should not be underestimated. Given the tools and intellectual capacity currently available in the Soviet states, however, it is possible to make the effort.

Soviets often claim that they have been able to coax great utility out of weak hardware through the use of more sophisticated algorithms and models. It has not been a goal of this study to confirm or disprove this claim. If it is true, however, parallel systems like those just described should offer an excellent opportunity to capitalize on this strength. Advances in massively parallel systems are now less a hardware issue than a software and algorithms issue. This fact offers Soviet engineers one of the best opportunities to make their systems internationally competitive. It remains to be seen whether they can capitalize on it.

9.2 Implications for HPC Users

The issue of accessibility depends on at least two factors: the availability of systems, and the number and type of individuals who are granted access to them. In the past, most high performance computers like the El'brus and PS- series have been installed either at closed military sites, or at non-military sites with specific, intensive data processing tasks, such as seismic data processing. In either case, the machines have not been generally available for the scientific community at large. The latter has had to work on outdated BESM-6 style machines or ES mainframes. The goal of using the El'brus systems at the center of extensive collective-use computing centers did not materialize.

Two recent trends could help reverse this state of affairs. First, influential individuals like G. G. Ryabov, the director of ITMVT, have expressed a strong design to make ma-

chines available to the broader research community through the creation of computer centers based on machines like the MKP. In this manner, several organizations could pool resources to acquire a powerful system, reducing the cost to each. Although there are few indigenous machines which could serve this purpose and the financial state of most organizations (including the government) is so precarious that establishing a center would be difficult, the intent is a good one, and is likely to serve the scientific community well in the long-term.

Second, the central planning organizations no longer dictate which organizations will acquire which machines. One of the reasons so few civilian organizations had El'brus computers was that those that had been built were taken by privileged customers. It is now possible in principle for any organization with the necessary resources to purchase Western technology, provided it does not fall under CoCom export control restrictions, and sometimes even if it does. The issue is no longer one of allocation priorities.

Of course, such a center requires an extensive telecommunications infrastructure to be widely accessible, but significant efforts are underway in both the private and public sector to improve telephone lines, establish satellite links, lay fiber optic cable, etc. This access will probably not be a major barrier to access for long.

The issue of hardware availability remains problematic. Soviet HPC users have not been well served in the past. The short-term prospects for acquiring systems from the indigenous HPC sector are very poor. What are the prospects for acquiring Western systems?

Although individual Soviet users have been able to accomplish much in weapons design, real-time mission control, etc., the CoCom export control regime has been very effective in denying Soviets access to high-performance computing, or at least making ac-

quisition and use of foreign systems very difficult. The Soviet HPC user community has been hurt much more than the development sector by export control regulations.

Because of recent advances in technologies, some opportunities will be available to Soviets regardless of the export control regulations. For example, powerful workstations are now being manufactured in quantities of half a million or more. Thanks to public domain software, these can be clustered together to execute a variety of jobs or portions of jobs in parallel. Although the peak performance for a single task may not be at supercomputer levels, such arrangements greatly increase system throughput.

Although the Cold War and relationships between the Western countries and those of the former Soviet Union have improved dramatically, significant national security concerns remain. The West has a legitimate concern that its high-performance systems will be diverted to military use, or, through lack of effective proliferation control efforts, finds their way into the hands of restricted third countries.

At the same time, there is a widespread desire in the West to modify the export control regulations to take into account a) the realities of global advances in computer technologies and b) the possibilities that the Soviets themselves can become part of the export control solution rather than the problem. Computer technologies have advanced so quickly and are manufactured in such quantities that efforts to control the diffusion of many types of mass-produced technologies are futile. If the Soviets can demonstrate their ability and to work together with Western governments, vendors, and users to establish a control regime which will keep sophisticated technologies from being diverted to military use or restricted destination countries, it is possible that the iron-clad controls of the past can be eased to the benefit of commerce, scientific progress, and the Soviet transition to a market economy. The issues of export control of dual-use technologies, including HPC, has been taken up in a number of forums, including [Good88; Schm91; Good93; Nrc93].

The question is how to build sufficient trust that formerly restricted technologies can be exported to the Soviet states. This issue is addressed in [Good93]:

> In any relationship, including that between countries, the reduction of confrontation does not lead immediately to an establishment of trust. The latter can be accomplished only through a) the multilateral establishment of procedures and mechanisms to achieve the goals of non-diversion and non-proliferation, and b) a series of small and incremental steps taken over time in which both parties demonstrate trust, trustworthiness, and a willingness to work together in mutually beneficial ways. These will necessarily involve an element of risk, since measures which give one party complete control over the actions of the other (e.g. iron-clad control over high-performance computer installations) give the latter no opportunity to demonstrate independent good faith and cooperation. [Soviets] must be given the opportunity to demonstrate understanding of and respect for the national security concerns of the United States and cooperate in preventing diversion and proliferation of sophisticated technology.

This document proposed a three-track framework for confidence-building in which systems could be selectively installed and used in the Soviet states. The three tracks are: application domains, institutional arrangements, and the means for controlling or monitoring HPC technologies.

> For each track, one can envision an evolution, conditional on continued cooperation and trustworthiness, from safer, more secure positions to those which involve greater risk of diversion. The tracks are loosely coupled in the sense that movement from a more secure to riskier positions on each track can be made at different rates. This flexibility makes a wide range of potential confidence-building sequences possible [Good93].

Successful confidence-building will not be easy. While there is reason to believe that the willingness of many Soviets to cooperate has increased, the government's ability to control has decreased. This means that it is not sufficient to rely on government assur-

ances alone; individual users and organizations must have a vested interest in cooperation as well.

There are numerous possible confidence-building sequences. Two are suggested in [Good93]. Russian scientists frequently claim that they have excellent algorithms. A joint project could be established to enable them to implement these algorithms on Western hardware platforms. At a first stage, a team of Russians could undergo training at a Western university in software development for a particular Western massively parallel system. At stage two, the Russians could implement their algorithms, developing programs to run on the parallel machine. This could take place in Russia on workstations with the appropriate software development environment. At stage three, the Russian team could work on debugging and tuning their algorithms together with Western colleagues on the Western machine. At stage four, a small configuration could be installed in Russia under the joint supervision of Russian and American researchers and Russian and Western export control administrations. As long as the Russians comply with all associated agreements, the installation could be upgraded annually, increasing the number of processors, the amount of main memory, etc.

A second example could be the creation of a computer center which would provide computer time to individuals conducting civilian research in a variety of application domains. At the first stage, a low-end, general-purpose machine from a leading Western supercomputer manufacturer could be installed at a prominent Russian university or Academy of Sciences computer center under the exclusive control of representatives of Western export control organizations and the computer's manufacturer. At this stage the system would run Western applications, or specifically approved Russian applications. At stage two, a set of research projects, conducted jointly by collaborating Western and Russian colleagues, would be selected and granted access to the machine. An international

commission could be established with the task of guaranteeing its appropriate use. Such a commission would have to include the full participation of the principal researchers using the system. Additional members would include a representative of the computer vendor, a representative of a Russian monitoring agency, and a representative from the Western export control establishment. The arrangement would rely for its success on the personal relationships and interests of the researchers, and the personal stake each has in ensuring an enduring, successful collaboration.

At stage three, the set of users and applications could be selectively widened. The international commission would retain a permanent core, with pairs of Western and Russian researchers participating for the duration of their projects.

At subsequent stages, the center could evolve in a number of different directions. The installation itself could be upgraded; the Russians could be given greater and greater monitoring responsibilities; the requirement that all projects be collaborations between Russian and Western colleagues could be removed; the center could be made available to a broader circle of users and/or applications, including deserving university students.

There are many more possible examples. In all cases, however, if they would join the international computing community, Soviet users will have to work together with the West to demonstrate an understanding of legitimate Western national security concerns and their willingness and ability to address those concerns.

9.3 Implications for Policy Makers

It is beyond the scope of this study to make the argument that it is in the West's best interests to draw the Soviets into the international community. If this argument is accepted then in light of the above discussion a number of changes to policy should be made [Good93]. Specifically,

- Controls on technologies of which 100,000 units or more have been sold should be significantly reduced, unless there are compelling reasons to the contrary.

Regardless of one's view of the geo-political aspirations of the Soviet states, the reality of export controls is that technologies produced in this volume are very difficult to control. From an economic perspective, it is in the interests of American industry to permit the sale of such products to the Soviet states. The market for workstations and chips which meet this criterion is much larger in dollars than that for high-end machines. In both the East and the West it is easier to sell one hundred $10,000. systems than one $1 million unit.

- Western export control administrations should consider plans for the installation of individual pieces of technology within the context of a series of measures, possibly leading up to the approval of otherwise restricted technology, conditional on compliance with prior agreements.
- Western export control administrations should give favorable consideration to a number of test-case sequences of confidence-building measures.
- Soviet administrations should establish export control regimes which satisfy the legitimate security concerns of Western nations and demonstrate their continued effectiveness.
- A variety of ''soft'' controls, or means of verification of the end-use of high-performance computer technology should be evaluated as a part of a sequence of confidence-building measures.

A variety of measures can be used to monitor the use of individual systems. These include, but are not limited to, using the computers to store detailed logs about certain aspects of computers usage such as which programs are being used by whom for how

long, patterns of system resource usage by individual programs, etc. Although such information is not sufficient to identify the higher-level problems being solved by a particular program, it is very useful in giving a general idea of how a system is being used.

9.4 Directions of Future Research

This study has concentrated on high-performance computing technologies and their associated R&D facilities. Although we have provided some information about the broader context, we have not presented a complete picture of all elements in the HPC sector. In particular, our discussion of the policy issues surrounding HPC has been sparse. Over the years, extensive discussions about HPC have been held at the highest levels of the Academy of Sciences, the State Committee on Science and Technology (the Council on Supercomputing in particular), the industrial ministries, and the Military-Industrial Commission, whose function was to coordinate activities in areas of strategic importance of the various ministries. The latter controlled most of the financing and production facilities needed to bring HPC research to fruition. A study of these discussions could reveal important information about how decisions were made to fund one line of development and not another, which external events (e.g. advances in Western technology, the U.S. Strategic Defense Initiative, etc.) had the greatest impact on HPC policy and why, and how the views of policy-makers served as a selection environment for individual research teams.

Similarly, a study of Soviet HPC would not be complete without studies of the production facilities, upstream industries, and users. Studies of these groups of players could contribute important information about how Soviet HPC developed in the past and, more importantly, how it might develop in the future.

The Soviet states have not completed their transition from the social, economic, and political systems of the Soviet Union. Neither have the R&D facilities discussed in this

study. While we have observed certain trends, the complete process of transition will only become apparent in some years when conditions have stabilized. A number of Western researchers have examined issues of organizational transformation, concentrating specifically on the path from one organizational form to the next. For example, Hinings and Greenwood discuss organizational archetypes and categorize paths by which organizations make a transition from one archetype to another [Hini88; Gree88]. Gemmill and Smith and others discuss the application of dissipative structures theory to the area of organizational transformation [Gemm85; Leif89]. Our study offers initial data on the transformation of selected organizations. Continued study will provide valuable data with which to evaluate or improve the theories of organizational transformation.

We have discussed the ability of the Soviets to develop advanced technology within one industrial sector. Further study is needed to determine how generalizable our conclusions are to other advanced technology sectors, and to prosaic technology sectors as well.

We have postulated some possible future directions for the Soviet HPC sector. Will it survive these turbulent years? In what form will it emerge? What role will this sector play in the international community? In some respects, the HPC sector is at the edge of a new era. The old HPC projects are dying, but new ones have only begun to emerge. Although there have been some encouraging changes, dawn has not yet arrived; we will have to wait to see what the new day will bring.

APPENDIX A. HIGH PERFORMANCE COMPUTING: CONTROLLABILITY AND COOPERATION

Joint Statement of the U.S. National Academy of Sciences and Russian Academy of Sciences Working Groups on High-Performance Computing[1]

by

Seymour E. Goodman
Vladimir K. Levin
Ivan D. Sofronov
Peter Wolcott
Aleksey V. Zabrodin

April, 1993

[1] Reprinted with permission from *High-Performance Computing: Controllability and Cooperation. Joint Statement of the U.S. National Academy of Sciences and Russian Academy of Sciences Working Groups on High-Performance Computing.* Courtesy of the National Academy Press, Washington, D.C.

A.1 Introduction

With the end of the Cold War, the relationship between the United States and the Russian Federation has become less adversarial on many fronts, from foreign policy to commerce to science and technology. The hope is that the two countries' basic principles and goals about the nature of relationships between nations have drawn closer. In this context, some of the key features of the Cold War, such as the regimes governing the export of sophisticated dual use technology from West to East and from North to South, should be reviewed and possibly modified.

At the heart of the issue are questions of which technologies or uses should be controlled and which technologies can practically be controlled in terms of their manufacture or proliferation. Controllability depends strongly on the nature of the technology, its availability on global markets, and the organizational arrangements governing its use and distribution. The control of high performance computing (HPC) is particularly problematic because controllability characteristics of constituent technologies (software, microcircuits, networks, integrated systems, etc.) vary widely, and thanks to the extraordinarily rapid rate of technological advance and diffusion throughout the world, leading-edge technologies move into the mainstream only a few years after their introduction.

In the past, control efforts have consisted of measures taken by the U.S. and its allies in the Coordinating Committee for Multilateral Export Controls (COCOM) which assumed active attempts by agencies of the Soviet Union to divert sophisticated technology. Reliance on Soviet cooperation was minimal. In recent years, remarkable progress has been made diplomatically in recasting the relationship between the U.S. and the Soviet Union/Newly Independent States from one that has been fundamentally confrontational to one that is more mutually beneficial. The cooperation of Russians can be a powerful factor in the export control equation. If the Russians can demonstrate their ability and

willingness to work with Western governments, vendors, and users in keeping sophisticated technologies from being diverted to military uses or restricted destination countries, it is possible that the iron-clad controls of the past can be eased to the benefit of commerce, scientific progress, and Russian transition to a viable market economy.

In any relationship, including that between countries, the reduction of confrontation does not lead immediately to an establishment of trust. The latter can be accomplished only through a) the multilateral establishment of procedures and mechanisms to achieve the goals of non-diversion and non-proliferation, and b) a series of small and incremental steps taken over time in which both parties demonstrate trust, trustworthiness, and a willingness to work together in mutually beneficial ways. These will necessarily involve an element of risk, since measures which give one party complete control over the actions of the other (e.g. iron-clad control over high-performance computer installations) give the latter no opportunity to demonstrate independent good faith and cooperation. Russians must be given the opportunity to demonstrate understanding of and respect for the national security concerns of the United States and cooperate in preventing the diversion and proliferation of sophisticated technology.

In the past, the Soviets' willingness to control diversion and proliferation was questioned, but their ability to do so was not. Strong, centralized political and military institutions effectively regulated sensitive technologies. Today, there are reasons to suspect that the willingness to cooperate has increased. However, it should be noted that the Russians' ability to control has decreased. Partly as a result, concerns about North-South proliferation of technologies to such countries as Iraq and Iran have grown. The Western community should acquire assurance that under the current conditions of fragmentation and decentralization of lines of authority the Russians have the ability to establish an effective, civilian, control regime.

This paper examines the present nature and inherent controllability of high-performance computing technologies. It discusses means of control in the context of broader efforts to create an environment in which the need for controls is reduced. Specifically, it sketches a three-track approach–focusing on application domains, institutional arrangements, and technologies and controls–to building confidence without unduly compromising national security or economic interests.

A.2 Controllability of High-Performance Computing Systems

The High-Performance Computing Act of 1991 defines high-performance computing as "advanced computing, communications, and information technologies, including scientific workstations, supercomputer systems (including vector supercomputers and large scale parallel systems), high-capacity and high-speed networks, special-purpose and experimental systems, and applications and systems software." The ability to store and manipulate large volumes of data and make the results accessible to local and remote users has made such systems powerful enabling factors in a wide variety of civilian and military applications. HPC systems have contributed to leading-edge developments in such diverse applications as the weapons design, integrated-circuit simulation, weather analysis, automobile crash simulation, seismic prospecting, and drug design. Computational research has become a third pillar of scientific advance, together with theoretical and experimental research, and is increasing in importance over time.

In this section we examine some of the trends in high-performance computing development in the United States and Russia. We concentrate on those technologies which could be, or are, most easily obtained in Russia.

A.2.1 Trends in HPC

Over the last half decade in particular the high-performance computing sector has been highly dynamic, witnessing remarkable advances in performance throughout all classes of machines, component bases, storage devices, architectures, software and software environments, data transmission technologies, etc. While systems with traditional vector-pipeline architectures continue to evolve steadily, the emergence of a variety of commercial massively parallel machines with performances in some instances rivaling or exceeding those of traditional vector-pipelined supercomputers has signaled an important era of transition in the field in which massively parallel processors (MPP) have a significant role in complex and increasingly heterogeneous computing environments. Building on dramatic advances in microprocessor technology, workstations, servers, and accelerator boards have blossomed into a $15 billion market in recent years. On the order of 500,000 workstations are sold worldwide annually.

A.2.1.1 HPC Trends in the United States

Some of the high-performance computer systems which have been introduced in just the last two years by U.S. companies are listed in Table A-1.

The rapid evolution of microprocessor technology is one of the main factors fueling the construction of hardware with impressive theoretical performance. Thanks to manufacturing technologies which can place well over one million transistors on a chip and advances in microprocessor architectures, single-chip microprocessors in volume manufacture today offer 50-200 MIPS. They are small enough that thousands can be combined in a reasonably sized, air-cooled cabinet. Individual microprocessors are the engines for powerful, user-friendly engineering workstations. Several microprocessors can be placed on a single printed-circuit board which can be slipped into standard slots in workstations or even personal computers.

Year	Company	Machine	# Processors	Peak Performance(64-bit)
(64-bit) VECTOR-PIPELINE SUPERCOMPUTERES				
1991	Convex	C-3 series	1-8	34.4-960 Mflops
	Cray Research	Y-MP C90	16	16 Gflops
		Y-MP/EL	1-4	133-532 Mflops
1992	Cray Research	Y-MP M90	2-8	666-2664 Mflops
MASSIVELY PARALLEL COMPUTERS				
1991	Intel	Paragon XP/S	66-4096	5-300 Gflops
	Thinking Machines	CM-5	32-1024	4-128 Gflops
	Kendall Square Research	KSR1	32-1088	40 Mflops/node
1992	nCUBE	nCUBE 2E,2S	8-8192	27-34,000 Mflops
OTHER				
1991	IBM	Power-visualization System	8-32	1280 Mflops
	Convex, HP	Meta series		
	IBM	RS/6000 cluster		
WORKSTATIONS				
1992	Sun Microsystems	SPARCstation10	4	400 MIPS (32-bit)
		SPARCcenter 2000	2-20	100 MIPS - 2.19 GIPS (32-bit)
ADD-IN BOARDS				
1992	Transtech	TTM110	1	60 Mflops
	Transtech	PARAstation	4	240 Mflops
	Sky Computers	SKYbolt	16	960 Mflops
	CSPI	SuperCard Quad-860	4	320 Mflops (32-bit)

Table A-1 Recent U.S. High-Performance Computers

Many MPP manufacturers have chosen to use commercial, off-the-shelf microprocessors to save design and development resources. These include the Intel family of parallel systems (based on 286, 386, and i860 microprocessors), Thinking Machines CM-5 (SPARC), and the Parsytec GmbH Parsytec GC (transputers). Other manufactures, including nCUBE, Kendall Square Research, and MasPar, have chosen to use highly integrated, customized processors on the argument that by tailoring the design to include only needed functionality more processors can fit in a given space. Although such processors cost more to design and develop, these companies feel the improved performance and

functionality outweigh the drawbacks. In general, the industry has not come to a consensus over whether customized or off-the-shelf ICs are preferable.

During the past decade the performance, functionality, and availability of storage systems, high-speed networks, software engineering environments, graphics workstations and visualization software, etc. have increased tremendously (although not necessarily keeping pace with advances in microprocessor technology). High-speed networks which provide access to HPC configurations and support the transfer of large volumes of data have transformed the way in which individuals in the HPC community conduct their research and collaborate with one another.

A.2.1.2 HPC Trends in the Soviet Union/Russia

The Soviet Union has conducted research and development of digital computers since the late 1940s. Although not without achievements, the HPC industry has not been able to keep up with the scope and pace of Western development, for a variety of systemic and technological reasons. Two of the most significant hinderances have been the complex and cumbersome political and economic structures needed to support the development of complex technology, and, correspondingly, a technology base unable to support the development and manufacture of machines with world-class performance. In particular, the Soviet/Russian microelectronics industry has been unable to achieve large-scale, reliable production of chips with less than 1.5 micron technology (approximately 30,000 transistors per chip). The manufacture of single-chip microprocessors with the level of integration of even the 386 has not been achieved. Table A-2 lists some of the principal Soviet/Russian HPC computers prototyped or put into series production over the last decade.

In part to try to achieve high performance using relatively slow components, Soviet designers concentrated their efforts on parallel systems. Collectively, the Soviet/Russian

Year	Machine	# Processors	Peak Performancs (64-bit)	Status
1992	El'brus-3	1-16	8.96 Gflops	near prototype
	Elektronika-SSBIS	1-2	500 Mflops	prototype
1990	MKP	1-2	1 Gflops	prototype
1988	PS-2100	64-640	1.5 GIPS (32-bit)	series prod.
1985	El'brus-2	1-10	125 MIPS	series prod.

Table A-2 Soviet/Russian High-Performance Computers

projects cover a spectrum of architectural approaches nearly as broad as that in the West, although not as deep. Two recent exceptions, initiated after the success of vector-pipeline computers in the late 1970s, are the vector-pipelined MKP and Elektronika-SSBIS.

Besides those listed, many projects, carried out chiefly within institutes of the Academy of Sciences or the ministry of higher education focused on homogeneous distributed systems. During the 1980s hardware prototypes were built using available indigenous technology, but in recent years developers have been calling such systems "transputer-like" and have focused their efforts on software, often using real transputers as a development base. Most of these researchers have become members of a newly-formed Russian Transputer Association. Many of these projects have been oriented towards developing parallel co-processors or accelerators for general-purpose host computers.

In the past, national security policies in both the Soviet Union and the Western countries forced the Soviet high-performance computing industry to develop to a large extent independently. Basic information about development trends in the West was available, but developers were forbidden to use Western components and users were forced to rely almost exclusively on indigenous computers. Some architectural innovations served partially to compensate for the weak component base, and users invested much effort in de-

veloping models, algorithms, and systems software which would compensate for computer deficiencies. The Russian party feels that such efforts have been very successful for high-priority applications. Little has been published in the West about Soviet/Russian innovations in models and algorithms in particular.

As a whole, the HPC community in Russia continues to suffer from deficiencies in storage devices, although individual high-priority configurations may be reasonably adequately supplied with moderate-capacity storage systems. Facilities for remote, interactive access to HPC installations are at best extremely limited. Only in the 1990s has electronic mail become generally available in Russia.

In recent years, orders for HPC technology and financing for development has been reduced significantly as a result of the conversion of defense industries to civilian production and the deterioration of the Russian economy in general. At the same time, many Russian policy barriers to international contact and cooperation have been lifted, making it likely that the Russian HPC community will, in the future, be more integrated into the world HPC community.

A.2.2 Controllability of HPC

High-performance computing systems have become a particularly problematic element of the export control regime. The extraordinarily rapid rate of technological advance means that products move quickly from leading-edge to the mainstream, threatening to make specific features of export control regulations obsolete before they are published. High-performance computing as a whole encompasses a wide variety of technologies from components to networks to large-scale systems to sophisticated software which have very different controllability properties. The field, dominated by American and Japanese companies is growing more international (e.g., C-DAC in Pune, India has recently started production of the transputer-based PARAM computer), as the

user base expands, production technologies are licensed outside the principal countries, and international networks provide access and rapid communications across national boundaries.

Off-the-shelf ICs are not easily controlled. The microprocessors mentioned in the previous section have been manufactured in volumes of 100,000 or more, are small enough that tens or hundreds can be packed in a suitcase, are sufficiently self-contained that units are replaced rather than repaired, and are widely available outside the COCOM member countries. While not all of them, strictly speaking, can be considered commodities according to the criteria spelled out in [Schm91], they will be within very few years.

All of the massively parallel and conventional vector-pipeline use some customized components. Customized components are easier to control than off-the-shelf or industry standard ones. They are manufactured in smaller quantities, have limited distribution, and not easily substitutable. The ease with which one could acquire the hardware necessary to build a given machine varies with the degree to which customized components and subsystems are used. We examine three types of high-performance systems which are among the least controllable from the perspective of denying the capability to construct them. We consider other systems not discussed explicitly to be more difficult to construct than these.

A.2.2.1 Intel Parallel Systems

The Intel systems grew out of work on the Cosmic Cube at the California Institute of Technology in the 1980s, a system with a hypercube structure using the off-the-shelf 8086 and 8087 microprocessors as nodes. Intel's Supercomputer Systems Division was formed in 1984 to commercialize large-scale parallel computer systems based on an implementation of standard Intel microprocessors. Between 1985 and 1992, Intel intro-

duced three more generations of machines, based on the 286/287, 386/387, and i860 microprocessors, and currently has an installed base of over 300.

In order to keep manufacturing costs low and leverage the enormous amount of research done in the workstation and personal computer sectors of the computer industry, Intel has sought to use commercial and non-exotic technologies to the greatest extent possible. For example, the iPSC/860, introduced in 1990, is constructed using the i860 commercial microprocessors, commodity CMOS memory components, the 5.25" disks used in most workstations, as well as widely used I/O, networking, operating system, and computer language standards.

An exception to this principle is the direct-connect inter-node communications system based on the proprietary VLSI communications chips which route messages from one node to another. These chips were developed and manufactured by Intel based on designs by researchers at the California Institute of Technology. Of all the hardware used in the iPSC/860, these chips would be the most difficult to acquire.

It is, however, a mistake to assume that simply acquiring and assembling the hardware is sufficient to build a high-performance system. Actual performance depends directly on the efficiency with which system resources are managed and data are moved from one location to another. Intel has invested hundreds of man-years and millions of dollars researching the most appropriate ways of managing system resources, taking advantage of the computing power the hardware offers, decreasing software development time, and providing computational results in a useful form. Without the proprietary Concurrent File System (CFS), System Resource Manager (SRM), the NX/2 operating system, and other important pieces of systems software and firmware, the hardware is all but useless. The effort and know-how required to develop the necessary systems software should not be underestimated.

A.2.2.2 Transputer-based Systems

A second type of system worth examining is transputer-based systems. Transputers are microprocessors developed by INMOS Ltd. in England. Two important qualities are the ease with which they can be configured into systems of various sizes, and the clear and stable interface between hardware and software which enables software to run on a single transputer or a network of transputers without change or even re-compilation. Because communications between processes on the same transputer or on different transputers uses identical instructions (with the inter-transputer communications taken care of by the hardware), the precise configuration of the hardware is largely transparent to the software.

The core of the transputer market currently is fault tolerant systems and embedded controllers. Significant numbers are also used in large multiprocessors and accelerator boards which can be plugged into commercial personal computers and workstations. Specialized boards are being marketed for basic computation, video frame-grabbing, graphics, A/D conversion, and more. Some boards even incorporate an i860 processor.

Construction of such boards does not require highly sophisticated, proprietary technology. One leading vendor does board design using commercially available CAD systems, purchases commercially available components and subcontracts out the PCB manufacturing and system assembly. The technology to manufacture eight-layer boards used here does not have to be state-of-the-art, and is within the capabilities of Russian manufacturers. The assembly is a combination of surface-mount (automated) and through-hole (manual) processes, also within the capabilities of Russian manufacturers.

Multiple transputers can be placed on one board (up to 32 for a board to be used in a workstation). Boards are placed into racks which are attached to the workstations via an Sbus-VME converter card which also is commercially available. No customization of

software is required for such an installation. Standard transputer software–the operating system, compilers, a toolkit, etc.–is supplied by INMOS and the vendor serves solely as a distributor. Systems with several hundred transputers and theoretical peak performances of several hundred Mflops can be configured in a straightforward manner.

Because transputers, communicating via built-in serial links and running widely available systems software, can so easily be configured into multiprocessor configurations it is difficult to prevent configurations such as those mentioned above from being assembled. To disable the hardware, the serial links connecting transputers must be disabled. Since the communications facilities are built into the transputers themselves, this is impractical to do, short of physically isolating individual transputers. In multi-board configuration, it would be possible to manufacture specialized boards in which the communications links on the board could be terminated before they reach the edge connector. This would prevent transputers on different boards from communicating with each other, while permitting a modest amount of parallel processing on each board individually.

For large systems consisting of several hundreds or thousands of nodes, the most difficult challenge is not the construction of the hardware, but the development of the systems and applications software necessary to use it effectively. Network operating systems, debuggers, and performance monitoring systems are crucial, and hard to develop. Much time must be spent porting or developing applications. Not yet widely available, such sophisticated software is still reasonably controllable.

The transputers and their basic systems software are not easily controlled, however. Over one million transputers have been manufactured, and annual world-wide sales are over 250,000. SGS Thomson, the principal transputer distributor, has offices in India and the Pacific Rim countries as well as Western Europe and North America. Nearly two

dozen companies are value added resellers, building complete systems based on the transputer.

A.2.2.3 RS/6000 Clusters

The RS/6000 workstation clusters recently announced by IBM represent an alternative path to high performance computing based on commercially available technology. IBM has widely advertised the fact that a cluster of RS/6000 workstations supplanted a Cray X-MP supercomputer at Lawrence Livermore National Laboratory (LLNL). The hardware consists of standard workstations equipped with boards manufactured by IBM for Ethernet, Token Ring, or FDDI fiber optic networks, and the associated cabling.

The heart of the cluster is in the software. Currently, the cluster can be organized as a serial-batch, or a parallel system. In the first case each program runs on only one machine, but programs can be submitted to any available computer in the cluster. Supporting this mode are the Network Queuing System (NQS) or the DQS system, a public-domain system developed at Florida State University. Supporting the parallel execution of a single program across multiple machines are the Network Linda, Parasoft Express, and PVM (Parallel Virtual Machine) environments. The latter is public domain, developed at Oak Ridge National Laboratory.

The most controllable parts of an RS/6000 cluster are the workstations and the individual network boards which are IBM proprietary. Although not yet commodities, with a worldwide installed based approaching 100,000 the RS/6000 is not easily controlled.

As the technology advances, clusters of computers increasingly will be applied to a single job. Gordon Bell, the well-known former Vice President for R&D at DEC and founder of the Gordon Bell Prize for achievement in parallel computation, states that "Important gains in parallelism have come from software environments that allow networks of computers to be applied to a single job. Thus every laboratory containing fast

workstations has or will have its own supercomputer for highly parallel applications. The rapid increase in microprocessor power ensures that the workstation will perform at near super speed for sequential applications. LAN environments can provide significant supercomputing for highly parallel applications by 1995" [Bell92].

A.2.3 Controlling the Acquisition of HPC

The number of high-end HPC installations is still quite small. For example, Intel has shipped only 300-400 parallel systems; Convex has over 1100 systems in the field; and Cray Research Inc. has an installed base of just over 300 mainline computers plus nearly 100 Cray YMP-ELs. The small numbers of units, single sources, and considerable effort required to install them are among the factors which make it relatively easy to keep track of where each system is located.

From the discussion above we can see that it is possible to construct high-performance systems with a high percentage of generally available, off-the-shelf hardware which is difficult to control. However, the construction of almost all massively parallel and vector pipeline high-performance computers requires some customized hardware and/or software. These components and the technologies necessary to produce them are the best candidates for control efforts. Software design tools for application-specific integrated circuits are commercially available from over a dozen firms (see Smith, 1992), but the technology to manufacture chips based on the designs is still controllable and should be a high control priority.

In addition, the sophisticated systems software needed to make systems run effectively is still a reasonable control target. In spite of the ease with which software can be copied, it is a very difficult task to port it from one type of hardware platform to another. It is nearly impossible without access to source code. This is particularly true of most par-

allel systems. Proprietary and closely held, the source code is perhaps the most controllable part of such a machine.

Workstations, with global production rates approaching half a million annually from a dozen or more vendors, are not easy to control; neither will computing clusters based on them. While leading vendors are cooperating with government policies to limit direct sales of restricted technology effectively, the installed base is so large that controls are "leaky", at best. Large numbers of workstations can be obtained easily in the Far East. U.S.-based workstation manufacturers have set up factories abroad. Particularly notable are Hewlett-Packard and Silicon Graphics Inc. SGI's Iris Indigo workstation, introduced in the U.S. in 1991, will soon be manufactured in China; HP built a factory in Shanghai in 1990 to manufacture the Apollo 9000 series 400 workstations.

Workstations constitute perhaps the most rapidly evolving sector of the computer industry. With development cycles under a year in many cases, prices on given models drop rapidly following introduction. Users replace models after only a few years of use, creating a large secondary market which is uncontrollable. Developing countries like India appear to have little problem acquiring workstations.

Currently the principal barriers to workstations in Russia have little to do with export control regimes. First, few organizations can afford to purchase machines costing tens of thousands of dollars each. Second, the support infrastructure is poorly developed. Western workstations will run for months without maintenance, but they do fail, and failures are more difficult diagnose and repair than is the case for PCs. As the technology becomes more available and the economic climate improves, these hindrances will ease.

A.3 A Framework for Confidence-Building Measures

We have examined the controlability of HPC technology to restricted countries. We now consider a three-track framework for confidence-building by which systems could be selectively installed and used in Russia. The three tracks are: application domains, institutional arrangements, and the means for controlling or monitoring HPC technologies. For each track, one can envision an evolution, conditional on continued cooperation and trustworthiness, from safer, more secure positions to those which involve greater risk of diversion. The possibilities discussed below are necessarily riskier than what has been permitted for HPC in the past. The tracks are loosely coupled in the sense that movement from more secure to riskier positions on each track can be made at different rates. This flexibility makes possible a wide range of possible confidence building sequences.

The framework does not assume that cooperation at any one level of society or government, or within any particular sector is sufficient for establishing confidence. Russia today is characterized by the decentralization and fragmentation of lines of authority. This creates both difficulties and opportunities, since governments are not as able to regulate the activities–for good or for ill–of individuals and organizations as they once were. The framework requires the cooperation of all individuals and organizations involved, from users up through national governments.

The success of confidence-building measures will depend to no small measure on the creation of incentives to cooperate for all parties involved. The main idea behind the sequence of confidence-building steps is that observing the agreements at the previous step will open new opportunities or capabilities for the next step. Inherent in the framework with its emphasis on a sequence of confidence-building steps is the notion that appropriate behavior today will be rewarded by increased opportunity or capability tomorrow.

A.3.1 Application Domains

The applicability of high-performance computing applications to military concerns varies considerably. Confidence-building measures should initially focus on applications which have little importance to the military and gradually move towards those which are marginally important.

According to [Gart91], the following are examples of applications with little direct military applicability:

- Design of pharmaceuticals through the simulation of proteins and molecules.
- Structural biology. The use of simulation and molecular dynamics methods to study the time-dependent behavior of biologically important macro molecules.
- Human genome project. Computer-assisted comparison of normal and pathological molecular sequences for understanding genomes and the basis for disease.
- Computational Ocean Sciences. The development of a global ocean prediction model.
- Astronomy. The processing of the large volumes of data generated by Very Large Array or Very Long Baseline Array radio telescopes.
- Quantum Chromodynamics (QCD). Simulation of QCD yield insight into the properties of strongly interacting elementary particles.
- Computational Chemistry. Simulation of molecules and chemical reactions are critical to the development of new materials.
- Financial applications. In the West, sophisticated econometric models and vast databases consume enormous amounts of computing power. The Russian financial infrastructure is still immature, but will, hopefully, strengthen.

- Commercial applications. Reservation systems, point-of-sales systems, etc. require fast access to large databases.

Other application domains having a greater, but indirect, relevance to military capability are crucial to economies in general, and the Russian economy in particular. These include:

- Transportation. Modeling of fluid and gas dynamics in three dimensions, such as the airflow around vehicles, fluid flows within engines.
- Superconductivity. Superconductivity can be a critical factor in future power transmission technologies, instrumentation. The basic properties of superconducting materials are not well understood.
- Efficiency of combustion. Studying the interplay between flows of various substances and the quantum chemistry principles governing the reactions between them.
- Oil and gas exploitation. Utilization of improved seismic analysis techniques and modeling the flow of fluids through geological structures.
- Nuclear fusion. Understanding the behavior of ionized gasses under high-temperature conditions with very strong magnetic fields.
- Prediction of weather, climate, and global change. Development and use of models regarding the interaction between atmosphere, ocean and biosphere system enabling long-range predictions.
- Engineering applications. The structural analysis of products.

Because of the potential military application of these areas, confidence-building measures here in the areas of institutional arrangements and control regimes might proceed more slowly than in the non-military cases listed earlier.

Nevertheless, because of their importance, the greatest efforts to make progress should, perhaps, be concentrated exactly in these areas.

Some application domains have direct and critical implications for both economic and military competitiveness. Efforts should be made to explore confidence-building measures here, but with greater caution than in other application domains.

- Material sciences. The understanding the atomic nature of materials and the development of new kinds of materials.
- Semiconductor design. The modeling of how semiconductors constructed out of faster materials operate.
- Vehicle dynamics. The analysis of the aeroelastic behavior of vehicles and their stability and ride characteristics.

Finally, application domains which have great military importance, but marginal economic importance. There is little reason to seek confidence-building measures in, for example:

- Vehicle signature. The reduction of acoustic, electromagnetic, and thermal characteristics of vehicles.
- Undersea surveillance. Tracking undersea vehicles.
- Cryptography.

Movement from safer to riskier applications could take place on one machine, as the set of allowable applications grows, or in multiple installations. In the latter case, one installation might be devoted to a safe application, while in subsequent ones riskier applications might be allowed. One of the problems that has plagued this approach, and that of permitting remote access to a machine under physical U.S. control, in the past is the difficulty of carefully monitoring and distinguishing between applications as they are run.

A.3.2 Institutional Arrangements

The technical composition and technological content of an installation influence the degree to which it might be diverted. In general, installations can vary in the scope of use, i.e. the spectrum of applications, the size and composition of the user community, the degree of openness about systems use, and the physical distribution of the hardware. In addition, one can categorize installations according to whether they are managed and used by people from one country or from several. The risk of diversion increases as the scope of use increases, the hardware becomes geographically distributed, and the management of the installation becomes more closed, private, and under the control of just one country. If Russians and non-Russians are working together on the same system, it is less likely that sensitive military applications will be run.

Least subject to diversion would be government-run facilities physically located in the U.S. or another NATO country, where citizens of other countries would be permitted either on-site or remote access. The risk of reverse engineering could be kept minimal, and the risk of diversion during times of international conflict would be essentially eliminated.

A higher risk might well be suitably controlled at public, centralized, tightly managed, international computing centers sponsored and managed by individual governments or international agencies such as the United Nations for the purpose of providing advanced computing resources for non-military research in specific application domains. Over time, the center could expand the user community, possibly offering time to the international community on a competitive basis. Researchers might submit detailed proposals for systems use; individual projects, selected on the basis of appropriateness to the center's mission, could be run under the supervision of the center's staff. Fundamental to

such an arrangement would be the on-going surveillance of system activities by the international community.

International installations could also be created within private joint ventures. Companies which routinely use high-performance computing such as Boeing, Chevron, and Sun Microsystems have established joint ventures in aerospace, oil and gas exploration, and computing. These companies could work together with their Russian partners to ensure non-diversion of imported advanced technologies. To a large extent, the success of the export control regimes in the past has been due to self-policing by Western companies; reputable firms have refused to deal with suspect customers because of the possible legal, financial, and negative publicity consequences of illegal transactions. This principle should be applicable in the the case of joint ventures as well. Western partners could be given permission to import high-performance computing technologies for use in the joint venture with the understanding that they will be held responsible for any diversion of the technology.

Centers under the management of a single country could be established at state-owned institutions such as universities and government research facilities and at new or newly privatized corporations. The opportunities for diversion would be less at facilities operating with non-proprietary data and applications, where activity could be monitored by a broader circle of observers. It is not clear *a priori*, however, whether a government organization is to be preferred over a private firm. On the one hand, the Russian government could be enlisted as a partner in preventing diversion; other the other hand, a private firm involved in non-military commercial activities would likely have weaker ties to the military. Questions such as these would likely have to be answered on an organization by organization basis.

Fundamental to the success of any of these scenarios is the establishment of mutually beneficial collaborative efforts using high-performance computing between Russian and Western researchers. The degree of commitment to the relationship (and, correspondingly, the willingness to avoid actions which threaten it) will be a function of the longevity of the relationship, the promise of future benefit, and the importance of the efforts to individual researchers as well as industrial or scientific sectors and the country as a whole.

Russia has rich and extensive pools of data in many branches of science listed in the previous section. In many cases these have suffered from inadequate computing facilities to process and analyze the data properly. A fruitful area of cooperation would be the application of Western computing technology to this data. Cooperation would be more closely knit and longer-term as researchers work together to conduct studies and experiments which generate new data as well.

It is important to note that collaborative work can begin prior to hardware installation in Russia. Russian researchers could literally bring data tapes to the West for processing. Alternatively, such data could be sent electronically to Western machines monitored by both countries.

Institutional arrangements also include government-level monitoring and control mechanisms. In the past, the Soviet government was able to exert effective control over the use and distribution of sensitive technologies through strong military, Party, and State Security structures. Each of them had strong military components and worked to a large extent, at least in the area of high-technology, on behalf of military interests. If Russia is to work together with Western countries to control diversion and proliferation, an effective civilian mechanism must be established which will not only exert control in the cases

of individual installations, but also provided continuity and consistency of control from one installation to another and over time.

No such mechanism currently exists with regard to high performance computing. There is doubt in the West that such a mechanism could perform an effective job under Russia's current economic and political conditions. First, the traditional pillars of Soviet society have partially lost their ability to control as a result of decentralization measures and the unregulated activities of powerful groups such as organized crime. Second, the dire economic straits are forcing individuals and organizations at all levels of society to skirt regulations merely to survive. The sale of advanced technology for hard currency to unauthorized customers is by no means inconceivable. Third, the Russian government has stated that it intends to maintain significant military capability and the ability to re-convert enterprises to military production if necessary.

Under these circumstances it is difficult to envision a civilian authority which could effectively control the diversion and proliferation of high performance computing technologies, even though in principle COCOM-like structures and procedures could be established within Russia as they have been within Hungary. Yet such structures must exist and be effective if Russia is to be a partner with the West in this area in the longer term. It is incumbent upon the Russians to design such structures and convince the West that they are effective.

A.3.3 Technologies and Measures for Control and Monitoring

Given any combination of application domain and institutional arrangement, a variety of control measures can be implemented to regulate and monitor system activities. "Hard" controls are those which seek to prevent diversion by making it difficult to carry them out. Measures which physically or logically control access to a given computer or otherwise restrict performance are of this nature. "Soft" controls, on the other hand, are

designed to detect violations, rather than prevent their occurrence. Confidence building measures involve a series of steps which increasingly reduce first hard controls and then soft controls. Soft controls in most cases will have to be in place longer than hard controls to provide objective verification that violations have not occurred.

A.3.3.1 Hard Controls

Hard controls seek actively to limit what can be accomplished on a computer and by whom. The Supercomputer Safeguards Plan (SSP) (Export Administration Regulations 15 CFR 776.11(f)(4)) places very stringent hard controls on systems with Composite Theoretical Performance (CTP) equal to or exceeding 195 million theoretical operations per second (MTOPS). The measures are designed to prevent unauthorized use through denial of physical access to systems by restricted nationals, strict control over the issuing of passwords, precise selection of which applications can be run and under what conditions, complete lack of connection to networks or remote terminals, etc.

While these restrictions do and will continue to accomplish the goal of controlling access, at a time when we are examining alternatives to such restrictions it is important to keep in mind that access is a necessary, but not sufficient condition for performing useful work. The performance and usefulness of a computer are dependent on many things, only some of which are taken into account in the computation of the CTP. Performance depends not only on the raw processing rate of individual processors, but also on the amount of memory available, the throughput of the interconnect system, the amount and speed of external memory, and the throughput of the I/O system. Unless processors are supplied with data at a high enough rate, they sit idle and accomplish no useful work. Software also plays a critical role. The overhead of the operating system, the efficiency with which it manages systems resources, and the effectiveness of the compiler can have a significant impact on performance. Instances in which the performance of the same

program on the same hardware is increased 100% or more, simply through the use of an improved compiler, are not uncommon.

In a real-world setting, human factors–the ease with which a user can accomplish a desired task–play a crucial role in determining a system's usefulness. The amount of time spent programming and debugging, and the time needed to analyze and interpret results strongly influence the utility of the machine to the user.

Each of these factors provides a means of regulating the effective performance and usefulness of a system. If a system is installed with insufficient external storage, a non-mature software development environment, a lack of sophisticated applications, inadequate tools to support the visualization and interpretation of results, or is used in an environment in which the ratio of software development to execution is high, the true performance indicated by the CTP will not be realized.

As confidence is built, given installations can be enhanced by selectively relaxing the constraints just mentioned. Processing elements can be added, more external or main memory can be installed, upgraded software packages can be provided, the number of applications authorized for execution can be increased, etc. At each installation, the prospect of future upgrades provides an incentive to cooperate.

A.3.3.2 Soft Controls

Soft controls make it possible to monitor the use of a system without necessarily preventing unauthorized use. The Supercomputer Safeguards Plan requires extensive soft controls to be used in conjunction with the hard controls. These include maintaining in a secure fashion usage logs and inspecting them daily, detecting attempts to gain unauthorized access, recording execution characteristics of each program run, and the monitoring of CPU and I/O usage.

Soft controls also serve as guards against proliferation, since the physical location of a system can be easily determined.

Some sort of soft control should be used until a high level of confidence in adherence to non-diversion agreements has been reached.

A.4 Recommendations

The recommendations in this section augment those presented in the "Joint Statement of the Delegations of the RAS and NAS on Dual Use technologies and Export Administration" and should be considered within the context of the latter document.

High-performance computing technologies are evolving very rapidly, particularly in the workstation arena where new generations are introduced every 2-3 years and equipment five years old is often considered obsolete. Recommendations for the control or decontrol of specific technologies are similarly quickly outdated.

- **Recommendation #1: Significantly reduce controls on technologies of which 100,000 units or more have been sold, unless there are compelling reasons to the contrary.**

Currently, microprocessors such as the i860 and T800 and many workstations fall into this category. While not necessarily commodities in the strict sense, such technologies are so widely available that control measures are very "leaky" at best.

From an economic perspective, the greatest benefit to American industry will come through the sale of large-volume products. In the West, the total size of the workstation market is an order of magnitude larger than the supercomputer market; the personal computer market is many times larger than the workstation market. In general and in Russian in particular it is much easier to sell one hundred $10,000. units than one $1 million unit.

With a threshold of 100,000 units, great economic gains can be made without severely compromising national security.

A basic premise of this paper is that Russians should be able to participate with Western countries in regulating the diffusion and use of high-performance computing systems, and that they should be given the opportunity to demonstrate their willingness and ability to do so. One means of accomplishing this is through the use of carefully selected sequences of confidence-building measures. Ideally, such sequences would serve as a testing ground for a variety of Russian, Western, and combined control measures, and serve as a model which could in the future be replicated. An additional benefit would be the placement of technology in Russia which could help stem the drain of computational scientists from Russia. But it is critical that the object of export control review be an entire sequence of steps, rather than an isolated installation.

- **Recommendation #2: Consider plans for the installation of individual pieces of technology within the context of a series of measures, possibly leading up to the approval of otherwise restricted technology, conditional on compliance with prior agreements.**
- **Recommendation #3: Give favorable consideration to a number of test-case sequences of confidence-building measures.**

We offer the following sequence as an example. Russian scientists frequently claim that they have developed methods of solving a variety of computational problems which are better in some sense that those developed in the West. As their contribution, in the interests of mutual cooperation in advanced high-performance computing technologies, the Russian scientists can adapt these methods to Western machines. At the first stage of a joint project, at team of Russians would undergo training at a Western university in software development for a particular Western massively parallel system. At the second

stage, the Russian team would implement their algorithms, developing programs to run on the parallel machine. This could be carried out in Russia on workstations with the appropriate software development tools. At the third stage, the Russian team would work on debugging and tuning their algorithms in concert with Western colleagues on the Western machine. At the fourth stage, a small configuration would be installed in Russia under the joint supervision of the Russian and American researchers, and Russian and Western export control administrations. Each subsequent year, as long as non-diversion agreements are not violated, the installation would be upgraded through adding more processing elements, memory, external storage, software, etc.

A second example could be oriented towards the creation of a prominent computer center which would provide computer time to individuals conducting civilian research in a variety of application domains. At the first stage, a low-end, general-purpose machine from a leading Western supercomputer manufacturer could be installed at a prominent Russian university or Academy of Sciences computer center under the exclusive control of representatives of Western export control organizations and the computer's manufacturer. At this stage the system could be used to run Western applications, or specifically approved Russian applications.

At a second stage, a set of research projects, conducted jointly by collaborating Western and Russian colleagues, would be selected and granted access to the machine. An international commission could be established with the task of guaranteeing its appropriate use. Crucial to the composition of this commission would be the full participation of the principal researchers using the system. Additional members would include a representative of the computer vendor, a representative of a Russian monitoring agency, and a representative from the Western export control establishment. Having both Russian and Western researchers involved would ensure that the commission contained the expertise

necessary to understand the applications being run. The arrangement would rely for its success on the personal relationships and interests of the researchers, and the personal stake each has in ensuring an enduring, successful collaboration.

At a third stage, the set of users and applications could be selectively widened. The international commission would retain a permanent core, with pairs of Western and Russian researchers participating for the duration of their projects.

At subsequent stages, the center could evolve in a number of different directions. The installation itself could be upgraded; the Russian could be given greater and greater monitoring responsibilities; the requirement that all projects be collaborations between Russian and Western colleagues could be removed; the center could be made available for a broader circle of users and/or applications, including deserving university students.

This second example assumes that successful use of an installation must be based on participants from the individual researchers up through the national government having a strong interest in guarding the system against inappropriate use. Although the existence of a Russian governmental structure with oversight over export control and the use of imported high-performance technology is not a sufficient condition, it is necessary.

- **Recommendation #4: Evaluate a variety of "soft" controls, or means of verification of the end-use of high-performance computer technology as a part of a sequence of confidence-building measures.**

The confidence-building measures will lead to fewer iron-clad controls over the use of particualar systems, but means of verification of use should be kept in place until sufficiently high levels of trust have been established, or technological developments make them unnecessary or impractical. Computer systems can store detailed logs about certain aspects of computer usage, such as which programs are being used by whom for how long, patterns of system resource usage by individual programs etc. Although such infor-

mation is not sufficient to identify the higher-level problem being solved by a particular program, it is very useful in giving a general idea of how a system is being used. Initially, such information would be gathered by Western systems managers on location. At later stages, such information could be gathered and transmitted automatically through satellite or other communications links to individuals monitoring the system. This would provide a relatively unobtrusive form of soft control.

APPENDIX B. GLOSSARY OF ORGANIZATIONAL ACRONYMS

ARPA	Advanced Research Projects Agency
CIS	Commonwealth of Independent States
CMEA	Council for Mutual Economic Assistance
CoCom	Coordinationg Committee for Multilateral Export Controls
DEC	Digital Equipment Corporation
FPS	Floating Point Systems Corporation
GKNT	USSR All-Union State Committee for Science and Technology *Obshchesoyuznyy Gosudarstvennyy komitet SSSR po nauke i tekhnike*
GKVTI	All-Union State Committee for Computer Technology and Informatics *Obshchesoyuznoy Gosudarstvennyi komitet SSSR po vychislitel'noy tekhnike i informatike*
HP	Hewlett Packard Corporation
IBM	International Business Machines Corporation
IK AN ESSR	Institute of Cybernetics of the Academy of Sciences of the Estonian SSR *Institut kibernetiki AN ESSR*
IK AN UkSSR	Institute of Cybernetics, Academy of Sciences of the Ukrainian SSR *Institut kibernetiki AN UkSSR*
INEUM	Institute of Electronic Control Machines *Institut elektronnykh upravlyayushchykh mashin*
IPM	Kel'dysh Institute of Applied Mathematics *Institut prikladnoy matematiki imeni Kel'dysha*
IPU	Institute of Control Problems *Institut problem upravleniya*
ISI	Institute of Informatics Systems *Institute system informatiki*
ITMVT	Institute of Precision Mechanics and Computer Technology *Institut technoy mekhaniki i vychislitel'noy tekhniki*
Impul's	Scientific Production Association Impul's *Nauchno-proizvodstennoye ob"edineniye "Impul's"*
KGB	State Committee for Security *Komitet gosudarstvennoy bezopasnosti*
KamAZ	KamAZ Giant Automotive Plant
LFIA	Lisichansk Subsidiary of the Institute of Automation *Lisichanskiy filial instituta avtomatiki*
LIAP	Leningrad Institute of Aviation Instrument Building *Leningradskiy institute aviapriborstroyeniya*
LIIA	Leningrad Institute of Informatics and Automation *Leningradskiy institut informatiki i avtomatizatsii*
MGU	Moscow State University *Moskovskiy gosudarstvennyy universitet*

NIIVK Scientific Research Institute of Computing Systems
Nauchno-issledovatel'skiy institut vychislitel'nykh kompleksov

Minaviaprom Ministry of the Aviation Industry
Obshchesoyuznoye Ministervo aviatsionnoy promyshlennosti SSSR

Minelektronprom USSR All-Union Ministry of the Electronics Industry
Obshchesoyuznoye ministerstvo elektronnoy promyshlennosti SSSR

Minelektrotekhpribor USSR All-Union Ministry of the Electrical Equipment Industry and Instrument Building
Obshchesoyuznoye Ministerstvo elektrotekhnicheskoy promyshlennosti i priborostroyeniya SSSR

Minelektrotekhprom USSR All-Union Ministry of the Electrical Equipment Industry
Obshchesoyuznoye Ministerstvo elektrotekhnicheskoy promyshlennosti SSSR

Minpribor USSR All-Union Ministry of Instrument Construction, Means of Automation, and Control Systems
Obshchesoyuznoy Ministerstvo priborostroyeniya, sredsva avtomatizatsii i upravlyayushchikh sistem SSSR

Minradioprom USSR All-Union Ministry of the Radio Industry
Obshchesoyuznoye Ministerstvo radio promyshlennosti SSSR

NATO North Atlantic Treaty Organization

NIIMVS Scientific Research Institute of Multiprocessor Computer Systems
Nauchno issledovatel'nskiy institut mnogoprotsessornykh vychislitel'nykh sistem

NIITT Scientific Research Institute of Precision Technology
Nauchno-issledovatel'skiy institut tochnoy tekhnologiy

NIIUVM Scientific Research Institute for Control Computers
Nauchno-issledovatel'skiy institut upravlyayushchikh vychislitel'nykh mashin

NIIUVM Scientific Research Institute for Control Computers
Nauchno-issledovatel'skiy institut upravlyayushchikh vychislitel'nykh mashin

NITsEVT Scientific-Research Center of Electronic Computer Technology
Nauchno-issledovatel'skiy tsentr po elektronno-vychislitel'noy tekhniki

NITsSN Scientific Research Center of Supercomputers and Neurocomputers
Nauchno-issledovatel'skiy tsentr super-EVM i neyrokomp'yuterov

OIVTA Department of Informatics, Computer Technology, and Automation
Otdeleniye informatiki, vychislitel'noy tekhniki, i avtomatizatsii

OZVM Orel' Control Computing Machines Factory imeni K. H. Rudnev
Orlovskiy zavod upravlyayushchikh vychislitel'nykh mashin imeni K. N. Rudneva

RSFSR Russian Soviet Federal Socialist Republic
Rossiskaya sovetskaya federal'naya sotsialisticheskaya respublika

SAM Moscow Calculating-Analytic Machines Plant

	Moskovskiy zavod schetno-analiticheskikh mashin
SO AN SSSR	Siberian Department, Academy of Sciences *Sibirskiy otdel, Akademii nauk SSSR*
SPZ	Severodonetsk Instrument Building Plant *Severodonetsk priborostroitel'nyy zavod*
VEM	Computing Electronic Machines Plant *Zavod vychislitel'nykh electronikh mashin*
VNIIEF	All-Union Scientific Research Institute of Experimental Physics *Vsesoyuznyy nauchno-issledovatel'skiy institut eksperimental'noy fiziki*
VPK	Military-Industrial Commission *Voyenno-promyshlennaya komissiya*
VTs SO AN SSSR	Computer Center of the Siberian Department of the Academy of Sciences *Vychislitel'nyy tsentr SO AN SSSR*
VUM	Control Computing Machines Plant *Zavod vychislitel'nykh upravlyayushchikh mashin*
YERNIIMM	Yerevan Scientific Research Institute of Mathematical Machines *Yerevanskiy nauchno-issledovatel'skiy institut matematicheskikh mashin*
ZEMZ	Zagorsk Electronic-Mechanical Factory *Zagorsk elektro-mekhanicheskiy zavod*

APPENDIX C. GLOSSARY OF ACRONYMS

AAP	Attached Array Processor
AI	Artificial Intelligence
Akademgorodok	Academic City, Novosibirsk
Akademset'	Academic Network
BESM	High-speed Electronic Calculating Machine *(Bystrodeystvuyushchaya elektronnaya schetnaya mashina)*
CISC	Complex Instruction Set Computer
CMOS	Complementary Metal-Oxide Semiconductor
DAN	Dynamic Automata Network
DRAM	Dynamic Random Access Memory
Dacha	Country Home
ECL	Emitter-Coupled Logic
EFRNT	Unified Fund for the Development of Science and Technology *(edinyy fond razvitii nauki i tekhni)*
ES	Unified System of computers *(edinaya sistema)*
FFT	Fast Fourier Transform
FIFO	First-In-First-Out
Gflops	Giga (billion) FLoating-point Operations Per Second
Goszakaz	State order
HPC	High-Performance Computing
IC	Integrated Circuit
I/O	Input/Output
KByte	Kilo (1024) bytes
Khozdogovor	Economic contract
Khozraschet	Economic self-accounting
KOPS	Kilo (thousand) Operations Per Second
LSI	Large-Scale Integration
MARS	Modular, Asynchronous, Extendable, System *(modul'naya, asinkhronnaya, razvivayushchaya sistema)*
MAYaK	Macro-pipeline Language *(MAkro-konveyernyy YazyK)*
MByte	Mega (1024*1024) bytes
MESM	Small Electronic Calculating Machine *(Malaya elektronnaya schetnaya mashina)*
MIMD	Multiple-Instruction, Multiple Data
MIPS	Million Instructions Per Second
MNTK	Inter-branch Scientific-Technical Complex *Mezhotraslevoy nauchno-tekhnicheskiy kompleks*
MSI	Medium-Scale Integration
Mflops	Mega (million) FLoating-point Operations Per Second

NPO	Scientific Production Association *(nauchno-proizvodstvennoye ob"edineniye)*
PEPE	Parallel Element Processing Ensemble
Perestroika	Restructuring
PO	Production Association *(proizvodstvennoye ob"edineniye)*
PS	Reconfigurable System *(perestrayevayemaya sistema)*
PSS	Procedure-oriented Static Scheduling
PTsM	Program-goal method *(programmno-tselevoy metod)*
RISC	Reduced Instruction Set Computer
S&T	Science and Technology
SIMD	Single-Instruction, Multiple-Data
SKB	Special Design Bureau *(spetsialnyy konstruktorskiy buro)*
SM	system of Small Computers *(sistema malykh)*
SOLOMON	Simultaneous Operation Linked Ordinal MOdular Network computer
SSI	Small-Scale Integration
TPP	Theoretical Peak Performance
TTL	Transistor-Transistor Logic
VLIW	Very-Long-Instruction-Word
VLSI	Very-Large-Scale Integration
VNTK	Temporary Scientific-Technical Collective *(vremennyy nauchno-tekhnicheskiy kolektiv)*
VUZ	Higher-education Institution *(vysshoye uchebnoye zavedeniye)*

REFERENCES

[Aben82] Aben, Kh K., “Cybernetics: Forecasts and Reality,” *Molodezh’ Estonii* (Apr. 17, 1982), 3.

[Afan87] Afanasyev, G., “"START" Project for Developing 5th-Generation "MARS" Computer Described,” *Moskovskaya Pravda* 66 (March 17, 1987), 3. (trans.: SNAP, Apr., 2, 1987)

[Agam91] Agamirzian, Igor, “Computing in the USSR,” *Byte* (Apr., 1991), 120-129.

[Akhm82] Akhmetov, R., “Computer: 200 Million Operations per Second,” *Sotsialisticheskaya Industriya* (Nov. 28, 1982), 1. (trans.: JPRS, 82695, Jan. 21, 1983, 9-11)

[Alek83] Alekseyev, A. S., Kotov, V. Ye., Kuznetsov, Ye. P., Metlyayev, Yu. V., Orlov, L. Kh, Pogrebnyak, V. F., Fishelev, V. I., “The Academic Regional Network of Siberia,” *Referativnyy Zhurnal, Avtomatika, Telemekhanika i Vychislitel’naya Tekhnika* 1 (1983), 21.

[Alek84] Aleksandrov, A. P., Suryabin, G. K., “From the USSR Academy of Sciences,” *Izvestiya* (Sep. 14, 1984), 3.

[Alek88] Aleksandrowicz, Piotr, “In Midstride: Metronex Has Reached Sakhalin, Siberia, and the Transcaucasus, but the Soviet Market Is Still Waiting To Be Discovered,” *Przeglad Tygodniowy* 17 (Apr. 17, 1988), 3.

[Alek91] Alekseyeva, M. B., Belyayev, S. P., Sorvachev, A. M., Sofronov, I. D., “Difficulties in Solving Continuous Medium Problems in Modular Pipeline Computers,” *Programmirovaniye* 5 (Sep.-Oct., 1992), 57-65. (trans.: JPRS, UCC-92-009-L, May 27, 1992, 30-40)

[Aman77] *The Technological Level of Soviet Industry*, Amann, R., Cooper, J., Davies, R. W., Eds., Yale University Press, London, 1977.

[Aman82] *Industrial Innovation in the Soviet Union*, Amann, R., Cooper, J., Eds., Yale University Press, New Haven, CT, 1982.

[Anan87] Anan’yev, A. V., Chochia, G. A., “Basic Mathematical Software of the High-Speed Complex ES-1037, ES-2706 in the Virtual Machines Operating System,” in *The ES-1037 - ES-2706 Multiprocessor Computer System: the Experience of Systems Development and Digital Modeling of Non-linear Physical Problems*, Sagdeyev, R. Z., Shevchenko, V. I., Eds., Institute of Space Research, Moscow, 1987, 7-19.

[Anto81] Antonov, V. S., Orayevskiy, K. S., Avtonomov, B. B., “Concept of ES-1060 Computer Design and Development,” in *Computer Technology of the Socialist Countries*, Rakovskiy, M. E., Ed., Statistika, Moscow, v. 9 (1981), 12-19. (trans.: JPRS, CCAT/L/10387)

[Apok74] Apokin, I. A., Maystrov, L. Ye., *The Development of Computers*, Nauka, Moscow, 1974.

[Arkh86] *Scientific-Technical Programs and the Management of the Scientific-technical Progress in Socialist Countries*, Arkhangel'skiy, V. N., Ed., Progress, Moscow, 1986.

[Astr88] Astrov, V., "Quality of Association's Ship Automation Equipment Defended," *Vodnyy Transport* 126 (Oct. 20, 1988), 1,3. (trans.: SNAP, Nov. 18, 1988)

[Baba77] Babayan, B. A., ''Basic Principles of Software for the El'brus Multiprocessor Computing Complexes,'' ITMVT (Moscow), Preprint No. 5, 1977, 11 pp.

[Baba89] Babayan, B. A., Ryabov, G. G., Chinin, G. D., "Elbrus Software Methodology: Instrumentation-Experience," in *Information Processing 89, Proceedings of the IFIP 11th World Computer Congress, San Francisco, USA, Aug. 28–Sep. 1, 1989*, Ritter, G. X., Ed., North-Holland, Amsterdam, 1989, 879-882.

[Baba89b] Babayan, B. A., "Basic Results and Prospects for Development of the "El'brus" Architecture," *Prikladnaya Informatika* v. 15 (1989), 100-131.

[Baba90] Babayan, B. A., Bocharov, A. V., Volin, V. S., Gavrilov, S. S., Groshev, A. S., Gruzdov, F. A., Yeremin, M. V., Zotov, S. M., Plotkin, A. L., Pshenichnikov, L. Ye., Ryabov, G. G., Chudakov, M. L., Shevyakov, V. S., *Multiprocessor Computers and the Methods of their Design.*, Smirnov, Yu, Ed., Vysshaya shkola, Moscow, 1990.

[Babe88] Babenko, L. K., Makarevich, O. B., Matveyeva, L. N., *Principles of the Organization of Computation in Multiprocessor Systems for Real-Time Applications*, preprint no. 16-88, IPPMM, AN UkSSR, L'vov, 1988.

[Babe89] Babenko, L. K., Makarevich, O. B., Sukhinov, A. I., Ovcharenko, O. I., "Means of Implementing Basic Computational Algebra Operations in Multiprocessor Computer Systems," *Upravlyayushchiye Sistemy i Mashiny* 5 (1989), 25-28.

[Babe91] Babenko, L. K., Makarevich, O. B., Chefranov, A. G., "Parallel Programming Technology and Resource Management in Multiprocessor Systems," in *Proceedings of the International Conference Parallel Computing Technologies, Novosibirsk, Sep. 7-11, 1991*, Mirenkov, N. N., Ed., World Scientific Publishing Company, Singapore, 1991, 495-502.

[Bakh89] Bakhmurov, A. G., Smelyanskiy, R. L., "The Design of an Architecture of a Specialized High-performance Computing System," in *Hardware-software Systems and Systems Software of Computing Systems*, Moscow State University, Moscow, 1989, 19-28.

[Bala82] Balakirev, N. Ye., Bartenev, Yu. G., Zel'dinova, S. A., Tyurin, V. F., Schlesinger, P. Ya., "DISPAK Operating Systems for the Integrated BESM-6," in *Raising the Effectiveness of the Use of High-Throughput Computers: Materials on Software*, Kopytov, M. A., Zel'dinova, S. A., Kobelev, V. V., Lipatova, Ye F., Mikhaylov, G. M., Samofalov, V. V., Eds., ANSSSR, Urals Science Center, Institute of Mathematics and Mechanics, Sverdlovsk, 1982, 10-21. (trans.: JPRS, UCC-84-007-L, May 30, 1984, 28-35)

[Bald82] Baldin, B. Yu., Govorun, V. N., Rybakov, V. G., Sytin, A. N., "SUMMA Microcomputer System Based on Intel 8080 Microprocessor Family," Abstract in conference proceedings *All-Union Scientific-Technical Conference on Individual Dialog Systems Based on Microcomputers* (Personal Computers): Dialog-82-Micro, Science Center for Biological Research AN SSSR, Pushchino, Oct. 25, 1982, 58-59. (trans.: JPRS, CCAT/L/11469, Jul. 21, 1983, 28)

[Bara82] Barabanov, V. A., Gerasimenko, V. Ye., Goncharova, A. F., Goncharov, V. G., Levitshiy, G. N., Sokolova, N. A., "SM-2M Computing Complex," *Pribory i Sistemy Upravleniya* v. 9 (Sept. 1982), 22-26. (trans.: JPRS, CCAT/L/10987, Dec. 3, 1982, 22-26)

[Bard87] Bardizh, V. V., *85 Years After the Birth of S A Lebedev*, Institute of Precision Mechanics and Computer Technology, Moscow, 1987.

[Bard88] Bardizh, V. V., "Academician Sergey Alekseyevich Lebedev," in *From BESM to Supercomputer*, Ryabov, G. G., Ed., Institute of Precision Mechanics and Computer Technology, Moscow, 1988, 5-16.

[Bbc86] ''Serial Production of the New ES-1066 Computer'', BBC Summary of World Broadcasts, Part I: The USSR, 1986, No. 8160C, C2.

[Beer80] Beer, Michael, *Organization Change and Development: A Systems View*, Goodyear Publishing Company, Inc., Santa Monica, CA, 1980.

[Bell92] Bell, Gordon, "Ultracomputers: A Tereflop Before Its Time," *Communications of the ACM*, Vol. 35, No. 8, Aug., 1992, 27-47.

[Bely80] Belyakov, M. I., Grinkrug, Ye. M., Krol', L. A., "General-Purpose Timesharing Operating System for the M-10 Multiprocessor Digital Computer," *Upravlyayushchiye Sistemy i Mashiny* 1 (Jan.-Feb., 1980), 37. (trans.: JPRS, CCAT/76313, Aug. 27, 1980, 57-59)

[Bere80] Berezovskiy, M. A., Ivanov, M. F., Petrov, I. V., Shvets, V. F., "Problems of Programming Modeling Problems on the M-10," *Programmirovaniye* 6 (1980), 37-45.

[Bere87] Berezovskiy, M. A., Minkin, A. L., Frolova, Ye. I., "Means of Automatic Parallelization of Programs for the ES-2706 Array Processor," in *The ES-1037 - ES-2706 Multiprocessor Computer System: the Experience of Systems Development and Digital Modeling of Non-linear Physical Problems*, Sag-

deyev, R. Z., Shevchenko, V. I., Eds., Institute of Space Research, Moscow, 1987, 53-68.

[Bere88] Berezhnoy, M., Mazalov, P., "A Rookie With Character," *Pravda* (Dec. 24, 1988), 1. (trans.: SNAP, Jan. 9, 1989, 3)

[Berk82] Berkovich, D. M., Mamedli, E. M., "All-Union Conference on High-Throughput Computing Systems," *Avtomatika i Telemekhanika* 10 (Oct. 1982), 166-173. (trans.: JPRS, CCAT/L/11202, Mar. 21, 1983, 121,124)

[Berk85] Berkovich, D. M., Mamedli, E. M., "The Second All-Union Conference on High-Performance Computing Systems," *Avtomatika i Telemekhanika* 9 (Sept., 1985), 168-175. (trans.: JPRS, UCC-86-004-L, Feb. 26, 1986, 56-64)

[Berl76] Berliner, Joseph S., *The Innovation Decision in Soviet Industry*, MIT Press, Cambridge, MA, 1976.

[Bern84] Berney, K., "IBM Eyes Niche in Burgeoning Supercomputer Market," *Electronics* (Jul. 12, 1984), 45-46.

[Bobk78] Bobko, I. M., *Automated Management Systems and Their Adaptation*, Nauka, Novosibirsk, 1978.

[Bobk86] Bobkov, V. A., Sukhoparov, A. I., Guzik, V. F., Kalyayev, A. V., Popov, Yu. P., Stanishevskiy, O. B., "The K1815VF3 Microprocessor for Digital Signal Processing," *Mikroprotsessornyye Sredstva i Sistemy* 2 (1986), 21-23,85.

[Bork91] Borkovsky, Arkady, Computers in the USSR: posted on TPS-L@Indycms, Mar. 5, 1991.

[Bosg811230] "US Allies Called Key To Pipeline Ban," *The Boston Globe* (Dec. 30, 1981).

[Boya77] "Automated Work Station of Designer Based on a Minicomputer System," *Pribory i Sistemy Upravleniya* 11 (1977), 7.

[Bras83] Brasch Jr., F. M., Yau, S. S., "Integrated Circuitry," in *Encyclopedia of Computer Science and Engineering, 2/e*, Ralston, A., Reilly Jr., E. D., Eds., Van Nostrand Reinhold Company, New York, NY, 1983, 772-777.

[Buch84] Bucher, I. Y., "The Computational Speed of Supercomputers," in *Supercomputers: Design and Applications*, Hwang, K., Ed., IEEE Computer Society Press, 1984.

[Bulm77] Bulman, D. M., "Stack Computers: An Introduction," *Computer* (IEEE) (May, 1977), 18.

[Burn61] Burns, T., Stalker, G. M., *The Management of Innovation*, Tavistock Publications, London, 1961.

[Burr72] ''Burroughs B 6700 Information Processing Systems Reference Manual (1058633)(5-72),'' Burroughs Corporation, 1972.

[Burt75] Burtsev, V. S., “Prospects for the Creation of igh-Performance Computers,” *Vestnik Akademii Nauk SSSR* 3 (Mar., 1975), 25-31.

[Burt78] Burtsev, V., “Computers: Relay-Race of Generations,” *Pravda* (Apr. 4, 1978), 3.

[Burt78b] Burt, Richard, “Carter, Reacting to Trials in Soviet, Will Control Oil-Technology Sales,” *New York Times* (July 19, 1978), A1,A4.

[Burt85] Burtsev, V. S., Krivosheyev, Ye. A., Asrieli, V. D., Borisov, P. V., Tregubov, K. Ya., “A Vector Processor with a Programmable Structure,” in *Computer Processes and Systems*, Marchuk, G. I., Voyevodin, V. V., Bakhvalov, N. S., Belykh, L. N., Dymnikov, V. P., Kotov, V. Ye, Kuznetsov, Yu A., Lebedev, V. I., Sarkisyan, A. S., Eds., Nauka, Moscow, v. 2 (1985), 63-72.

[Burt87] Burtsev, V. S., *Reliability Characteristics of Multiprocessor Systems and an Analysis of the Reliability of the El’brus-2 MVK. Preprint # 169*, USSR Academy of Sciences, Moscow, 1987.

[Burt89] Burtsev, V. S., Krivosheyev, Ye. A., Asrieli, B. D., Borisov, P. V., Tregubov, K. Ya., “The MVK El’brus-2 Vector Processor,” in *Supercomputers*, Burtsev, V. S., Ed., Computation Mathematics Department, USSR Academy of Sciences, Moscow, 1989, 1-29. (trans.: JPRS, UCC-90-002, Mar. 27, 1990, 18)

[Byak90] Byakov, A. Yu., Kropachev, Yu. A., *Modular Pipeline Processor (Preliminary Description), Part 1. Architecture, Data, Control*, ITMVT, Moscow, 1990.

[Byak91] Byakov, A. Yu., “Modular Pipeline Processor Architecture and Organization of Programming,” *Programmirovaniye* 5 (Sep.-Oct., 1991), 34-56. (trans.: JPRS, UCC-92-009-L, May 27, 1992, 3-29)

[Cart80] “Transcript of President’s Speech on Soviet Military Intervention in Afghanistan,” *New York Times* (Jan. 5, 1980), 6.

[Chai88] Chaikovskiy, M. G., “Architecture, Instruction Set and Software of the "El’brus-B"," in *Software and Hardware of High-Performance Computing Complexes*, Palagin, A. V., Ed., IKANUKSSR, Kiev, 1988, 25-29.

[Cher86] Chernis, Ye. N., “Specifics of the Realization of Programming Languages on the El’brus Microcomputer,” in *Proceedings of the III All-Union Conference on Automation of the Production of Programming Systems*, Pryugi, K., Ed., Institute of Cybernetics of the Estonian Academy of Sciences, Tallinn, v. 3 (1986), 94-96.

[Cher91] Chernyayev, A. P., “Programming Systems for Vectorizing and Parallelizing FORTRAN Programs for Several Vector-pipelined Computers (Overview),” *Programmirovaniye* 2 (Mar.-Apr., 1991), 53-68.

[Cher92] Chernozhukov, Ye. S., “There Were Differences of Opinion,” *Impul’s* 3 (Nov., 1992), 1.

[Clar87] Clark, Norman, “Similarities and Differences Between Scientific and Technological Paradigms,” *Futures* (Feb., 1987), 26-42.

[Clem84] Clementi, E., Corongin, G., Detrich, L., Chin, S., Domingo, L., “Parallelism in Quantum Chemistry: Hydrogen Bond Study in DNA Base Pairs as an Example,” in *International Journal of Quantum Mechanics: Quantum Chemistry Symposium*, v. 18 (1984), 601-618.

[Coop88] Coopersmith, Jonathan C., Interview with Valetin Ivanovich Shevchenko, Jul. 14, 1988.

[Crow93] Crowe, G. D., Goodman, S. E., ‘‘S. A. Lebedev and the Birth of Soviet Computing,’’ work in progress, 1993.

[Dani84] Danil’chenko, Igor’, “Computer Technology: To A New Level,” *Ekonomicheskoye Sotrudnichestvo Stran-Chlenov SEV* (Moscow) (Nov., 1984), 14-18.

[Dani84b] Danilov, G. A., Lebedev, V. N., “Data Base Computers,” *Prikladnaya Informatika* 1 (1984), 94-106.

[Dani86] Danilchenko, I., “Up-to-date Computing Facilities Boost the Economic Development of Socialist Countries,” *Elorg Information* v. 11, 1 (1986), 2-4.

[Davi78] Davis, N. C., Goodman, S. E., “The Soviet Bloc’s Unified System of Computers,” *Computing Surveys* (ACM) v. 10, 2 (June, 1978), 93-122.

[Denn68] Dennis, J. B., “Programming Generality, Parallelism and Computer Architecture,” in *Proceedings of the IFIP Congress, 1968*, North Holland, Amsterdam, 1968, 484-492.

[Dijk68] Dijkstra, E. W., “The Structure of the THE Multiprogramming System,” *Communications of the Association for Computing Machinery* v. 11, 5 (May, 1968), 341-347.

[Dimi82] Dimitriyev, Yu. K., Khoroshevskiy, V. G., *Computer Systems with Minicomputers*, Yevreinov, E., Ed., Radio i svyaz’, Moscow, 1982. (trans.: JPRS, CCAT/L/11341, 1-73)

[Doly82] Dolya, V. I., Tyutyunnik, L. I., Chuprin, I. F., “Results of Field Tests of Two Algorithms for Solving Nonlinear Optimization Problems of Oil Mixing,” *Avtomatizatsiya i Kontrol’no-Izmeritel’nye Pribory v Neftepererabatyvayushchey i Neftekhimecheskoy Promyshlennosti* 2 (Mar.-Apr., 1982), 13-16. (trans.: JPRS, CCAT/L/10973, 32-33)

[Dong88] Dongarra, Jack J., “The LINPACK Benchmark: An Explanation, Part I,” *Supercomputing Magazine* (Spring, 1988), 10-14.

[Dong90] Dongarra, Jack J., "Performance of Various Computers Using Standard Linear Equations Software," *Supercomputing Review* (Jul., 1990), 49-56.

[Dong92] Dongarra, J. J., ''Performance of Various Computers Using Standard Linear Equations Software,'' CS-89-85, May 30, 1992, Mathematical Sciences Section, Oak Ridge National Laboratory

[Dong92b] Dongarra, J. J., Snyder, L., Wolcott, P., *Parallel Processing Research in the Former Soviet Union*, SAIC, McLean, VA, Mar., 1992.

[Dons90] Donskov, S. V., Inyakin, A. V., Mikhailov, Yu. V., Sin'govskiy, A. V., Sugonyayev, V. P., Shagin, P. M., Shtannikov, A. V., "Special-purpose Processor for Particle Sampling by Momentum in an Experiment to Study Central Hadron Collisions," *Pribory i Tekhnika Eksperimenta* 4 (Jul.-Aug., 1990), 96-100.

[Doro92] Dorozhevets, M. N., Wolcott, P., "The El'brus-3 and MARS-M: Recent Advances in Russian High-Performance Computing," *The Journal of Supercomputing* 6 (1992), 5-48.

[Doro92b] Dorozhevets, M. N., ''The MARS-M Computer: Overview,'' Working paper, Institute of Informatics Systems, Novosibirsk, 1992.

[Doro92c] Dorozhevets, M. N., Conversation with PW, Mar. 6, 1992.

[Dosi82] Dosi, Giovanni, "Technological Paradigms and Technological Trajectories," *Research Policy* v. 11 (1982), 147-162.

[Dosi84] Dosi, G., *Technical Change and Industrial Transformation*, The MacMillan Press, Ltd., London, 1984.

[Dosi88] Dosi, G., Orsenigo, L., "Coordination and Transformation: An Overview of Structures, Behaviors and Change in Evolutionary Environments," Chapter 2 in *Technical Change and Economic Theory*, Dosi, G., Freeman, C., Nelson, R., Silverberg, G., Soete, L., Eds., Pinter Publishers, London, 1988, 13-37.

[Dosi88b] Dosi, Giovanni, "The Nature of the Innovative Process," Chapter 10 in *Technical Change and Economic Theory*, Dosi, G., Freeman, C., Nelson, R., Silverberg, G., Soete, L., Eds., Pinter Publishers, London, 1988, 221-238.

[Dpro71] ''Burroughs B 5700, B 6700, B7700,'' Datapro Research Corp., June, 1971.

[Dpro77] ''Burroughs B6700 and B7700'', Datapro Research Corp., Jan. - March 1977.

[Dpro82] ''All About Plug-Compatible Disk Drives,'' Datapro Research Co., Jul., 1982.

[Dpro86] "IBM 3090 Series," 70C-504MK, Datapro Research Corporation, Apr., 1986.

[Draz85] Drazin, R., VandeVen, A. H., "Alternative Forms of Fit in Contingency Theory," *Administrative Science Quarterly* v. 30 (1985), 514-539.

[Ecot85] "ES-1066 Computer," *Ecotass* 33 (1985), 24.

[Eko79] "Sigma Automated Management System," *Ekonomika i Organizatsiya Promyshlennogo Proizvodstva* (Novosibirsk) 5 (May, 1979), 59-60. (trans.: JPRS, CCAT/76465)

[Ekon89] "Developer of Multiprocessor With Dynamic Architecture," *Ekonomicheskaya Gazeta* (Sept. 1989), 19. (trans.: SNAP, Sep. 19, 1989, 2)

[Elec87] "Computers (USSR)," *Electronique Actualites* (Paris) (Dec. 11, 1987). (trans.: FBIS S&T Perspectives, Vol. 3, No. 2, Mar. 7, 1988, 2)

[Elek91] Unattributed note. From a brochure received at CeBIT'91 in Hannover, GE, 13-20 Mar., 1991. Received Apr., 1991.

[Es76] *Documentation System of Unified Computer System*, Statistika, Moscow, 1976. (trans.: JPRS, CCAT/L/10928)

[Ezh9008] "On measures for creating and developing small enterprises," *Ekonomika i Zhizn'* 33 (Aug., 1990), 4.

[Ezh9008b] "On the general foundations of the activity of small enterprises," *Ekonomika i Zhizn'* 34 (Aug., 1990), 18.

[Farr91] Farrens, M. K., Pleszkun, A. R., "Implementation of the PIPE Processor," *Computer* (IEEE) v. 24, 1 (Jan., 1991), 65-70.

[Fauc68] ''First All-Union Conference on Programming,'' 11 volumes, Kiev (Nov., 1968).

[Favo81] Favorskaya, A., "Each Estonian Computer Scientist Pursues a Topic Which Will Contribute to Science," *Sovetskaya Estoniya* (Dec. 16, 1981), 2. (trans.: JPRS, CCAT/80411, 1982)

[Favo86] Favorskaya, A., "Institute's Work on Artificial Intelligence and Expert Systems," *Sovetskaya Estoniya* 178 (Aug. 2, 1986), 2. (trans.: SNAP, Aug. 15, 1986)

[Feig84] Feigenbaum, E. A., McCorduck, P., *The Fifth Generation. Artificial Intelligence and Japan's Computer Challenge to the World*, 2nd ed., New American Library, New York, 1984.

[Fish84] Fisher, Joseph A., "The VLIW Machine: A Multiprocessor for Compiling Scientific Code," *Computer* (IEEE) v. 17, 7 (Jul., 1984), 45-53.

[Fish91] Fisher, Joseph A., Rau, B. Ramakrishna, "Instruction-Level Parallel Processing," *Science* v. 253, 5025 (Sep. 13, 1991), 1233-1241.

[Ford77] Ford, J. D., Slocum Jr., J. W., "Size, Technology, Environment and the Structure of Organizations," *Academy of Management Review* (Oct., 1977), 561-575.

[Fort86] Forescue, S., *The Communist Party and Soviet Science*, Macmillan Press, London, 1986.

[Fort90] Fortescue, Stephen, *Science Policy in the Soviet Union*, Routledge, London, 1990.

[Gakh82] Gakhariya, L. G., "Control of Functions of the Multimode Operating System for the M-10 Computer," *Voprosy Radioelektroniki Seriya EVT* 6 (1982), 125.

[Gart91] The Gartner Group, *High Performance Computing and Communications: Investment in American Competitiveness,* prepared by the Gartner Group, Inc. for the U.S. Department of Energy and Los Alamos National Laboratory (1991),

[Gavr86] Gavrilov, B. P., "Development of Leading Technology and Automation of Production in Instrument Building," *Pribory i Sistemy Upravleniya* 2 (Feb., 1986), 42-44.

[Gemm85] Gemmill, G., Smith, C., "A Dissipative Structure Model of Organization Transformation," *Human Relations* v. 38, 8 (1985), 751-766.

[Gigl921217] Giglavyy, A., Tchashchin, K., "Russian Supercomputers Stay on Track," *Newsbytes* (Dec. 17, 1992).

[Gigl930122] Giglavyy, A., Tchashchin, K., "Russian High-Tech Hermits Hit the Scene," *Newsbytes* (Jan. 22, 1993).

[Glus74] Glushkov, V. M., Myasnikov, V. A., Ignat'yev, M. B., Torgashev, V. A., "Recursive Machines and Computing Technology," in *Information Processing 74, Proceedings of IFIP Congress 74*, Rosenfeld, Jack L., Ed., North-Holland Publishing Company, Amsterdam, 1974, 65-71.

[Glus78] Glushkov, V. M., Kapitonova, Yu. V., Letichevskiy, A. A., "Toward a Theory of Design of Hardware and Software for Multiprocessor Computers," *Kibernetika* 8 (1978), 1-15.

[Glus78b] Glushkov, V. M., Burtsev, V. S., Rabinovich, Z. L., Tsukernik, L. V., *Sergey Alekseyevich Lebedev*, Khazanet, S. M., Chusov, A. T., Eds., Naukova dumka, Kiev, 1978.

[Glus79] Glushkov, V. M., "USSR, Computing in," Chapter in *Encyclopedia of Computer Science and Technology*, Belzer, J., Holzman, A. G., Kent, A., Eds., Marcel Dekker, Inc., New York and Basel, v. 13 (1979), 498-507.

[Golo80] Golovkin, B. A., *Parallel Computer Systems*, Nauka, Moscow, 1980. (trans.: JPRS, CCAT/L/9703, Apr. 30, 1981, 78-93)

[Golo85] Golovkin, B. A., "Structures of High-Performance Computer Systems and Relationships to Algorithm and Program Structures," *Izvestiya Akademii Nauk SSSR: Tekhnicheskaya Kibernetika* 5 (1985), 194-229. (trans.: JPRS, UCC-86-016-L, Jul. 10, 1986, 1-47)

[Golo86] Golovkin, B. A., "Multiprocessor Computer Complexes El'brus," *Programmirovaniye* 4 (1986), 76-87. (trans.: JPRS, UCC-87-014-L, Apr. 10, 1987, 7-20)

[Golo87] Golovkin, B. A., Yegisapetov, E. G., Makeyev, V. G., "Software for Robustness of Real-Time Computer Systems Using the M-10 Computer," *Elektronnoye Modelirovaniye* 3 (IZVANTK), 58-62.

[Golo88] Golovin, O. V., *Elektrosvyaz'* 2 (Feb. 1988), 42-46. (trans.: JPRS, UEE-88-007, Dec. 14, 1988)

[Goly83] Golyas, Yu. Ye., Tikhonov, V. A., "Microprocessor Devices in Radio Engineering," *Radiotekhnika* 1 (1983), 11-15. (trans.: JPRS, CCAT/L/11342, 14-21)

[Good79] Goodman, S. E., "Software in the Soviet Union: Progress and Problems," *Advances in Computers* v. 18 (1979), 231-287.

[Good87] Goodman, S. E., "The Information Technologies and Soviet Society: Problems and Prospects," *IEEE Transactions on Systems, Man, and Cybernetics* v. SMC-17, 4 (Jul.-Aug., 1987), 529-552.

[Good88] Goodman, Seymour E., et al., *Global Trends in Computer Technology and Their Impact on Export Control*, National Academy Press, Washington, D.C., 1988.

[Good93] Goodman, S. E., Levin, V. K., Sofronov, I. D., Wolcott, P., Zabrodin, A. V., "High-Performance Computing: Controllability and Cooperation. Joint Statement of the U.S. National Academy of Sciences and Russian Academy of Sciences Working Groups on High-Performance Computing," (Apr., 1993).

[Gorb85] Gorbachev, M. S., "Report at the Plenary Session of the Central Committee of the CPSU," *Pravda* (Apr. 24, 1985), 1-2. (trans.: CDSP, Vol. XXXVII, No.17, May 22, 1985, 1-11)

[Gorb85b] Gorbachev, M. S., "The Fundamental Question of the Party's Economic Policy," *Pravda* (Jun. 12, 1985), 1-2. (trans.: CDSP, Vol. XXXVII, No. 23, Jul. 3, 1985, 1-7)

[Gorb87] Gorbachev, M. S., Moscow Television Service in Russian at 0531 GMT on 16 April 87. Live address by Mikhail Sergeyevich Gorbachev (UR), General Secretary of the CPSU Central Committee, at the 20th KOMSOMOL Congress in the Kremlin Palace of Congresses. (trans.: JPRS, USSR Daily Report, Apr. 17, 1987)

[Gorb89] Gorbachev, M. S., "Law of the Union Of Soviet Socialist Republics: About Making Changes in the USSR Law 'About the Council of Ministers of the USSR'," *Moskovskaya Pravda* (July 7, 1989), 1.

[Goro84] Gorokhovskiy, S. S., Kapitonova, Yu. V., Letichevskiy, A. A., Molchanov, I. N., Pogrebinskiy, S. B., "An Algorithmic Language MAYAK," *Kibernetika* 3 (1984), 54-75.

[Goro84b] Gorokhovskiy, S. S., Kapitonova, Yu. V., Letichevskiy, A. A., Molchanov, I. N., Pogrebinskiy, S. B., "The Algorithmic Language MAYAK," *Kibernetika* 3 (1984), 54-77.

[Gors89] Gorshkov, N. V., "Twenty Years of Cooperation," *Vychislitel'naya Tekhnika Sotsialisticheskikh Stran* 26 (1989), 3-11.

[Grah75] Graham, L. R., "The development of science policy in the Soviet Union," Chapter 2 in *Science Policies of Industrial Nations*, Long, T. Dixon, Wright, C., Eds., Praeger, New York, 1975.

[Grea83] Great Soviet Encyclopedia, Macmillan, New York, 1983.

[Gree88] Greenwood, R., Hinings, C. R., "Organizational Design Types, Tracks and the Dynamics of Strategic Change," *Organization Studies* v. 9, 3 (1988), 293-316.

[Grin82] Grinkrug, Ye. M., "Parallel Processes in the Time-Sharing Operating System of the M-10 Computer," *Voprosy Radioelektroniki Seriya EVT* 6 (1982), 24-38.

[Grin90] Grinkrug, Ye. M., *Programming Authorization in the M-13 Computer*, Institute of Precision Mechanics and Computer Technology, Moscow, 1990.

[Groc88] Grochowski, J. B., "Partnership: A Soviet Initiative, but the Deal Was Good for Both Sides," *Zolnierz Wolnosci* (Warsaw) (July 7, 1989), 1,2.

[Grub77] Grubov, V. I., Kirdan, V. S., *Handbook of Computers and Analog Devices*, Pukhov, G. E., Ed., 2nd ed., Naukova Dumka, Kiev, 1977.

[Grub80] Grubov, V. I., Kirdan, V. S., *Handbook of Electronic Computer Equipment*, Vishcha Shkola, Kiev, 1980.

[Grub89] Grubov, V. I., Kirdan, V. S., Kozubovskiy, S. F., *Computer Catalog*, Naukova dumka, Kiev, 1989.

[Grue81] Gruenewald, A., "Improved ES-1055M Computer," *Pribory i Sistemy Upravleniya* 11 (Nov., 1981), 7-9. (trans.: JPRS, CCAT/L/10466)

[Gryz88] Gryzlov, A. A., "The BESM-4 Computer," in *From BESM to Supercomputer*, Ryabov, G. G., Ed., Institute of Precision Mechanics and Computer Technology, Moscow, 1988, 93-98.

[Gudi82] Gudim, I., "Efficiency of Computer Use Questioned," *Sovetskaya Estoniya* (July 8, l982), 3. (trans.: JPRS, CCAT/81919, 5-9)

[Gure85] Gurevich, V., Semenor, V., "Dependence on West for Computerization Denied," *Foreign Broadcast Information Service* v. 3, 15 (1985), U1.

[Guse83] Gusev, O., "Informatics Changes Its Colors," *Pravda* (Nov. 12, 1983), 3. (trans.: JPRS, UCC-84-003)

[Gust80] Gustafson, T., "Why Doesn't Soviet Science Do Better Than It Does?," in *The Social Context of Soviet Science*, Lubrano, L. L., Solomon, S. G., Eds., Westview Press, Boulder, 1980, 31-67.

[Gust83] Gustafson, T., *The Soviet Gas Campaign*, The Rand Corporation, Santa Monica, 1983.

[Hage80] Hage, J., *Theories of Organizations*, John Wiley & Sons, New York, 1980.

[Hamm84] Hammer, C., Dale, A. G., Feldman, M. B., Goodman, S. E., McHenry, W. K., Schwartz, J., Walker, S. T., Winograd, S., ''Soviet Computer Science Research,'' FASAC-TAR-2020, Washington, D.C., July 31, 1984.

[Harv83] Private Communications at Soviet Science and Technology: Eyewitness Accounts Seminar (Soviet Computing), Harvard University, Nov. 18-19, 1983.

[Hini88] Hinings, C. R., Greenwood, Royston, *The Dynamics of Strategic Change*, Basil Blackwell, New York, 1988.

[Hoar78] Hoare, C. A. R., "Communicating Sequential Processes," *Communications of the Association for Computing Machinery* v. 21, 8 (Aug., 1978), 666-677.

[Hock88] Hockney, R. W., Jesshope, C. R., *Parallel Computers*, 2nd ed., Adam Hilger, Bristol, 1988.

[Hord90] Hord, R. M., *Parallel Supercomputing in SIMD Architectures*, CRC Press, Boca Raton, 1990.

[Hpcw930405] "Alpha AXP Breaks Supercomputer Record for Fastest Sort," *HPCwire Select News* (Apr. 5, 1993).

[Huda89] Hudak, Paul, "Conception, Evolution, and Application of Functional Programming Languages," *Computing Surveys* (ACM) v. 21, 3 (Sep., 1989), 359-411.

[Huds92] Hudson, Richard L., "Western Supercomputer Firms Begin Seeking Customers in Eastern Europe," *Wall Street Journal* (Aug. 28, 1992), A7B.

[Igna84] Ignatushchenko, V. V., *Organization of the Structure of Multiprocessor Control System*, Trapeznikov, V. A., Voronov, A. A., Mamikonov, A. G., Aven, O. I., Berkovich, D. M., Eds., Energoatomizdat, Moscow, 1984.

[Igna86] Ignat'yev, M. B., Sheynin, Yu. Ye., "Recursive Computational Machines and New Generation Computers," in *The Development of New Generation Com-*

puters: Architecture, Programming, and Intellectualization, Yershov, A. P., Kotov, V. Ye, Eds., Computer Center of the Siberian Department of the USSR Academy of Sciences, Novosibirsk, 1986, 27-38.

[Igna89] Ignat'yev, M. B., Sheynin, Yu. Ye., "Highly Parallel Architecture for New Generation PPC," in *Sixth Symposium on Microcomputer and Microprocessor Applications*, 1989, 467-474. (trans.: JPRS, UCC-90-004, May 9, 1990, 6-7)

[Iht910629]"A Bloc Dies: East Puts Comecon to Rest," *International Herald Tribune* (Jun. 29-30, 1991), 9.

[Ikan90] ''The V. M. Glushkov Institute of Cybernetics,'' brochure, Vneshtorgizdat, 1990.

[Ilif68] Iliffe, J. K., *Basic Machine Principles*, American Elsevier Publishing Company, Inc., New York, 1968.

[Ilyu82] Ilyukovich, A. A., Zveryugo, S. M., "Problems of the Design and Creation of Computer Centers and Collective Use Systems," *Upravlyayushchiye Sistemy i Mashiny* 3 (May-June, 1982), 126-27.

[Impu85] ''The PS-3000 Computing System,'' prospectus, NPO Impul's,Severodonetsk, 1985.

[Impu89] PS-2100 Computer Complexes, Prospectus, 1989.

[Impu89b] PS-1001 Control Computer System, Prospectus, 1989.

[Impu89c] Promotional Brochures from NPO Impul's, 1989.

[Impu91] Plenary Session, 35th Anniversary Jubilee Conference, NIIUVM, Sep. 18, 1991.

[Impu91b] PS-2300 Multiprocessor, Brochure, NPO Impul's, 1991.

[Info88] "A New Personal Computer," *Informatika i Obrazovaniye* 2 (1988), 81.

[Iosh87] Ioshpa, A. Z., Miroshniko, P. G., Pabinovich, V. M., Rozenberg, A. Ye., Trakhtengerts, E. A., Shcherbakov, Ye. V., "The Functional Structure of the PS-3000 Operating System," *Upravlyayushchiye Sistemy i Mashiny* 5 (Sep.-Oct., 1987), 114-117.

[Iten85] Itenberg, I. I., Lobak, M. A., Bernatskiy, V. P., Largin, V. A., "Vector Processor of PS-3000 Control Computer Complex," *Pribory i Sistemy Upravleniya* 6 (June, 1985), 10-11.

[Iten89] Itenberg, I. I., Pivovarov, G. Yu., Rezanov, V. V., "The PS-2100 High-Performance Problem-oriented Computing System," *Pribory i Sistemy Upravleniya* 6 (1989), 1-4.

[Iten90] Evaluation of the Performance of the PS2100 on Linpack Benchmark Tests (draft), NPO Impul's, Sep. 1990.

[Iten90b] Itenberg, I. I., Nabatov, A. S., Conversation with PW, Sep. 17, 1990.

[Iten91] Itenberg, I. I., Conversation with PW, Sep. 19, 1991.

[Iten91b] Itenberg, I. I., ‘‘The PS-2300 multiprocessor,’’ MVS ’91, Sep. 10, 1991.

[Itmv90] *Institute of Precision Mechanics and Computer Technology*, ITMVT, Moscow, 1990.

[Ivan85] Ivanov, V. K., Fokin, A. F., Tseloval’nikov, A. M., Yakovlev, V. N., “Application of the High-Throughput Electronic Geophysical Computer Complex PS-2000 in Solving Mining Geophysics Tasks,” *Razvedka i Okhrana Nedr* 5 (May, 1985), 45-49.

[Ivan87] Ivanov, I., “Prospects for Domestic Supercomputers,” *NTR: Problemy i Resheniya* (Apr. 21-May 4, 1987), 2. (trans.: SNAP, Jun. 9, 1987, 2-3)

[Ivan88] Ivanova, I., “Bulgaria’s Growing Technological Importance to the USSR,” *Radio Free Europe Research* (July 27, 1988).

[Izv841224] *Izvestiya* (Dec. 24, 1984), 3. (trans.: SNAP, Jan. 5, 1985, 2-3)

[Izv880329] “From the Committee On Lenin and State Prizes of the USSR (In the Area of Science and Technology Under the USSR Council of Ministers),” *Izvestiya* 89 (Mar. 29, 1988), 2.

[Izv890811] “On the introduction of amendments to the Law of the USSR ’On the State Enterprise (Association)’,” *Izvestiya* (Aug. 11, 1989), 1.

[Izv900606] “The New Insitute in Academgorodok,” *Izvestiya* 158 (Jun. 6, 1990), 1.

[Izv910720] “Drop in Production Has Been Halted, According to a Report from the USSR State Statistics Committee,” *Izvestiya* (July 20, 1991), 3. (trans.: CDSP, Vol. XLIII, No. 29, Aug. 21, 1991, 23-24)

[Jako85] Jakobson, G., *Soviet Artificial Intelligence Research: The Tallinn Institute of Cybernetics*, Delphic Associates, Falls Church, VA, 1985.

[K86] K, A., “What is the ES 1036 Capable Of?,” *Szamitastechnika* (Budapest) 6 (June 1986), 2,4. (trans.: JPRS, EST-86-020, Sep. 4, 1986, 41-43)

[Kale87] Kalenchuk-Porkhanova, A. A., Yakovleva, T. S., “Hardware and Software of Complexes and Systems for Joining Non-Homogenous Computers,” *Upravlyayushchiye Sistemy i Mashiny* 2 (Mar.-Apr., 1987), 124-125.

[Kalm72] Kalmykov, V. D., “Computers of the Third Generation,” *Ekonomicheskaya Gazeta* 10 (Mar., 1972), 7.

[Kalo86] Kaloshkin, E. P., Basil’yev, L. V., “An LSI Microprocessor Chip Set for Pipeline Digital Signal Processing Systems,” *Mikroprotsessornyye Sredstva i Sistemy* 2 (1986), 14-15.

[Kaly88] Kalyayev, A. V., Makarevich, O. B., Nikolayev, I. A., Babenko, L. K., "A Supercomputer With a Programmable Architecture," in *EVT Electronic Computer Technology*, Przhiyalkovskiy, V. V., Ed., Radio i svyaz', Moscow, v. 2 (1988), 161-171.

[Kaly88b] Kalyayev, A. V., Nikolayev, I. A., "Multiprocessor Computer Systems with Programmable Architecture," chapter 3 in *High-performance Systems for Parallel Processing of Information*, Gritsyka, V. V., Ed., Naukova dumka, Kiev, 4 (1988), 100-148.

[Kant88] Kanter, R. M., "When a Thousand Flowers Bloom: Structural, Collective, and Social Conditions for Innovation in Organization," in *Research In Organizational Behavior*, Staw, Barry M., Cummings, L. L., Eds., JAI Press, Inc., Greenwich, CN, v. 10 (1988), 169-211.

[Kart79] Kartsev, M. A., "The M-10 Computer," *Doklady Akademii Nauk SSSR* 245 (Mar. 1979), 309-312.

[Kart80] Kartsev, M. A., Marshalko, B. G., "Some Questions on the Structural Organization of Special Coded Optoelectronic Computer Complex," *Avtometriya*2 (1980), 3.

[Kart81] Kartsev, M. A., "Principles of Organizing Parallel Calculations, Structures of Calculation Systems, and Their Realization," *Kibernetika* 2 (Mar.-Apr., 1981), 68-74.

[Kart81b] Kartsev, M. A., Brik, V. A., *Computer Systems and Synchronous Arithmetic*, Ivanushko, N. D., Ignat'yev, N. A., Eds., Radio i svyaz', Moscow, 1981.

[Kart83] "M. A. Kartsev (obit.)," *Moskovskaya Pravda* 95 (Apr. 27, 1983), 4.

[Kash79] Kashun, I. N., Sedukhina, L. A., "Parallel Data Processing in the Homogeneous SUMMA Computer System," *Vychislitel'nyye Sistemy* 80 (1979), 85-89. (trans.: JPRS, CCAT/L/9692, Apr. 27, 1981, 82)

[Kata88] ''Adaptive Local Hierarchical Network Architecture,'' *Software for Computers*, Tsentrprogrammsistem, 1988.

[Katk78] Katkov, V. L., ''Software for the El'brus-1 and El'brus-2 MVK at the Novosibirsk Subsidiary of the ITMVT,'' ITMVT (Moscow), Preprint No. 25, 1978, 14 pp., *Referativnyy Zhurnal, Avtomatika, Telemekhanika i Vychislitel'naya Tekhnika* 1 (Abstract 1B129, 1979).

[Kezl86] Karpov, B. V., Kezling, G. B., Leont'yev, D. I., Lozinskiy, N. N., Lavrusenko, R. S., Palachev, V. Ya., Tyurin, R. A., Filippov, A. M., Beleyevskaya, T. M., *Hardware of Automated Management Systems, Handbook in Two Volumes. 1. Unified System Hardware*, Kezling, G. B., Ed., Mashinostroyeniye, Leningrad, 1986.

[Khom87] Khomenko, L. G., “Stages and Directions of the Development of Domestic Computer Technology,” *Upravlyayushchiye Sistemy i Mashiny* 6 (Nov.-Dec., 1987), 123.

[Khom89] Khomenko, L. G., “History of the Creation in the AN UkSSR of the First Indigenous EVM,” *Ocherki istorii estestvoznaniya i tekhniki* v. 36 (1989), 74-81.

[Khor82] Khoroshevskiy, V. G., et al., “SUMMA Computer System Architecture,” *Elektronnoye Modelirovaniye* 3 (1982), 11-15. (trans.: JPRS, CCAT/L/10928, Nov. 9, 1982, 25)

[Klin82] Kling, R., Scacchi, W., “The Web of Computing: Computer Technology as Social Organization,” *Advances in Computers* v. 21 (1982).

[Klin84] Kling, R., Iacono, S., “The Control of Information Systems Development After Implementation,” *Communications of the Association for Computing Machinery* v. 27, 12 (Dec., 1984), 1218-1226.

[Klio89] Kliorin, M. I., Kadulin, V. Ya., Smolkin, V. M., *SM-2M Control Computer Systems*, Rezanov, V. V., Ed., Energoizdat, Moscow, 1989.

[Komp91] Brochure received at Softools ’91, Oct., 1991.

[Korn92] Korneyev, V. V., Katalov, Yu. T., “Computing Systems on the Basis of a Set of RISC-processor LSI Chips,” *Vestnik Rossiyskoy Transp’yuternoy Assotsiatsii* 6 (1992), 24-28.

[Koto66] Kotov, V. Ye., Narin’yani, A. S., “Asynchronous Computing Systems with Common Memory,” *Kibernetika* 3 (1966), 64-71.

[Koto86] Kotov, V. E., Marchuk, A. G., “Architecture of the MARS System Layout,” in *Computer Systems and Software*, Kotov, V. E., Vishnevskiy, Yu L., Lel’chuk, T. I., Marchuk, A. G., Eds., Siberian Division of the Academy of Sciences of the USSR, Novosibirsk, 1986, 5-12.

[Koto86b] Kotov, V. Ye., Marchuk, A. G., Vishnevskiy, Yu. L., “MARS - A Hierarchical Heterogeneous Modular System,” in *Fifth Generation Computer Architectures. Proceedings of the IFIP TC 10 Working Conference on Fifth Generation Computer Architectures, Manchester, U.K., Jul. 15-18, 1985*, Woods, J. V., Ed., North-Holland, Amsterdam, 1986, 277-289.

[Koto87] Kotov, V. Ye., “On the Path to the New Generation of Computers,” *Ekonomika i Organizatsiya Promyshlennogo Proizvodstva* (Novosibirsk) 3 (March, 1987), 39-54. (trans.: JPRS, UEO-87-001, Jun. 9, 1987, 25-35)

[Koto91] Kotov, V. Ye., “Concurrency + Modularity + Programmability = MARS,” *Communications of the Association for Computing Machinery* v. 34, 6 (Jun., 1991), 32-45.

[Koto91b] Kotov, V. Ye., “Project Start,” *Communications of the Association for Computing Machinery* v. 34, 6 (June, 1991), 30-31.

[Kova87] Kovalenko, V., “Computers: Steps to the World Level,” *Sotsialisticheskaya Industriya* (Aug. 4, 1987), 2. (trans.: JPRS, UCC-88-002, Jul. 8, 1988, 1-12)

[Kras90] Krashenko, S. N., Conversation with PW, Sep. 19, 1990.

[Kriu80] Kriushin, V. N., Buravtseva, I. N., Pushkina, N. M., Chernyak, N. G., *Computers and Computer Networks*, Statistika, Moscow, 1980. (trans.: JPRS, CCAT/L/10502)

[Kron89] ‘‘Kronos 2.6WS,’’ Brochure from the Computer Center of the Siberian Department of the USSR Academy of Sciences, 1989.

[Kron93] Kronenberg, N., Benson, T. R., Cardoza, W. M., Jagannathan, R., Thomas, B. J., “Porting OpenVMS from VAX to Alpha AXP,” *Communications of the Association for Computing Machinery* v. 36, 2 (Feb., 1993), 45-53.

[Kuch81] Kucherenko, V. K., Kucherenko, I. K., “Minicomputer Voice Communication System,” *Pribory i Sistemy Upravleniya* 1 (Jan., 1981), 13-14. (trans.: JPRS, CCAT/L/9773, 45,49)

[Kuch85] Kuchukyan, A. T., Sarkisyan, Y. Ye., Ter-Israyelyan, V. A., “Development of ES-1045 Computer and Computer Complexes Based on It,” *Computer Technology in the Socialist Countries*, Finansy i Statistika, 17 (1985), 59-65. (trans.: JPRS, UCC-86-006-L, Apr. 10, 1986, 59-65)

[Kuch90] Kucherenko, Ye. I., Pettsol’d, A. Ye., Taraday, A. N., “Effective Means for Interfacing Mini- and Microcomputers to FMS,” *Pribory i Sistemy Upravleniya* 6 (1990), 21-22.

[Kudr84] Kudryashov, V. N., Mamzelev, I. A., “Switching Peripheral Devices to a Homogeneous Computer System,” *Elektronnoye Modelirovaniye* 6 (1984), 60-64.

[Kush86] Kushlin, V. I., *The Interaction Between Science and Production*, Ekonomika, Moscow, 1986.

[Kuzn61] Kuznetsov, V. I., “Computing Technology,” Chapter 5 in *Achievements in the Field of Technological Progress in the USSR*, Proftekhizdat, Moscow, 1961.

[Kuzn86] Kuznetsov, D. N., Nedorya, A. E., Osipov, A. V., Tarasov, E. V., “The Kronos Processor in a Multiprocessor System,” in *Computer Systems and Software*, Kotov, V. E., Vishnevskiy, Yu L., Lel’chuk, T. I., Marchuk, A. G., Eds., Siberian Division of the Academy of Sciences of the USSR, Novosibirsk, 1986, 13-19.

[Kuzn89] Kuznetsov, D. N., Nedorya, A. Ye., Tarasov, Ye. V., Filippov, V. E., “Kronos: A Family of Processors for High-Level Languages,” *Mikroprotsessornyye Sredstva i Sistemy* v. 1989, 6 (June, 1989), 2-4.

[Kyae86] Kyaeramees, K. I., Mytus, L. L., Myartin, K. O., Pall', I. V., Reytsakas, Yu. A., Tani, Kh I., "Experimental Object-Oriented Personal Computer," in *Computer Systems and Software*, Kotov, V. E., Vishnevskiy, Yu L., Lel'chuk, T. I., Marchuk, A. G., Eds., Siberian Division of the Academy of Sciences of the USSR, Novosibirsk, 1986, 20-31.

[Lakh90] Lakhtin, G. A., *The Organization of Soviet Science: History and The Present*, Sominskiy, V. S., Ed., Nauka, Moscow, 1990.

[Lari87] Larina, T., "Supercomputer in Operation," *Sotsialisticheskaya Industriya* 250 (Oct. 30, 1987), 1.

[Laut91] Laut, V. N., "BESM-6," in *From BESM to Supercomputer*, Ryabov, G. G., Ed., Institute of Precision Mechanics and Computer Technology, Moscow, v. 2 (1991), 5-12.

[Lawr67] Lawrence, P. R., Lorsch, J. W., *Organization and Environment: Managing Differentiation and Integration*, Harvard University, Cambridge, 1967.

[Laza91] Lazareva, Natalya, "A Billion Per Second," *Terminal* 21 (Oct. 18, 1991), 1.

[Laza91b] Lazareva, N., "A Billion Per Second," *Delovoy Mir* 161-162 (Jun., 1991), 5. (trans.: JPRS, UCC-92-004-L, Mar. 10, 1992, 8-10)

[Leif89] Leifer, R., "Understanding Organizational Transformation Using a Dissipative Structure Model," *Human Relations* v. 42, 10 (1989), 899-916.

[Leti85] Letichevskiy, A. A., "Software for a Macroconveyer Computer Complex," in *Computer Processes and Systems*, Marchuk, G. I., Voyevodin, V. V., Bakhvalov, N. S., Belykh, L. N., Dymnikov, V. P., Kotov, V. Ye, Kuznetsov, Yu. A., Lebedev, V. I., Sarkisyan, A. S., Eds., Nauka, Moscow, v. 2 (1985), 91-95.

[Li91] Li, V., Conversation with PW, Oct. 3, 1991.

[Liso88] Lisovskiy, I. M., "S. A. Lebedev and the MESM," in *From BESM to Supercomputer*, Ryabov, G. G., Ed., Institute of Precision Mechanics and Computer Technology, Moscow, 1988, 17-23.

[Litv81] Litvinov, G. L., Tolstykh, R. I., Shadurskaya, L. I., Dmitrieva, Ye. A., *Mathematical Algorithms and Programs for Minicomputers*, Finansy i statistika, Moscow, 1981.

[Loef83] Loeffler, W., "Integrated Circuits from the USSR," *Radio Fernsehen Elektronik* (East Berlin) 11 (1983), 690-691. (trans.: JPRS, ESA-84-007, Mar. 5, 1984, 3-8)

[Loma86] Lomazova, I. A., "Reducing Transformations for Functional Programs With Assignment," *Programmirovaniye* v. 12, 3 (May-June 1986), 17-24.

[Lomo86] Lomov, Yu. S., Slutskin, A. I., Zadykhaylo, I. B., "Issues of Building Supercomputers," in *The Development of New Generation Computers: Architecture, Programming, and Intellectualization*, Yershov, A. P., Kotov, V. Ye, Eds., Computer Center of the Siberian Department of the USSR Academy of Sciences, Novosibirsk, 1986, 60-74.

[Lomo87] Lomov, Yu. S., "The ES-1066 and ES-1065 High Capacity Computers," in *Electronic Computing Technology*, Przhiyalkovskiy, V. V., Ed., Radio i Svyaz', Moscow, v. 1 (1987), 177-188.

[Lomo91] Lomov, Yu. S., Przhiyalkovskiy, V. V., "Ryad-4 Program for the ES Computer. High Performance ES Computers," in *Cybernetics and Computer Technology*, Mel'nikov, V. A., Ed., Nauka, Moscow, 5 (1991), 67-81. (trans.: JPRS, UCC-92-004-L, Mar. 10, 1992, 16-23)

[Lube88] Lubeck, O. M., ''Supercomputer Performance: The Theory, Practice, and Results,'' LA–11204–MS, Jan. 1988

[Lusa78] Lusa, J. M., "IBM Delivers First 303xs, Announces Multiprocessor," *Information Systems* v. 25, 5 (May 1978), 25-27.

[Mair81] Mairov, L., "Computer Technology Bridge Between Scientific Technological Progress," *Nauka i Tekhnika* v. 17, 3 (1981), 4-7.

[Maly91] Malyshkin, V. E., "Organization of Parallel Computations in a Large-Block Linear Hierarchical Multiprocessor Computing System," *Programmirovaniye* 4 (Jul.-Aug., 1991), 82-94. (trans.: JPRS, UCC-92-006-L, Apr. 1, 1992, 69-85)

[Mamz84] Mamzelev, I. A., Nikolayenko, N. N., Rusakov, M. Yu., Chasovnikov, Ye. D., "The Computer Linkup Device for a Homogeneous Computing System," *Otkrytiya Izobreteniya* 8 (1984), 157.

[Mamz89] Mamzelev, I. A., Knyazev, K. G., "Parallel Programmable Computer Systems in Telecommunications," in *Proceedings of the First International Computers and Communications Conference, Budapest, Oct. 25-26*, 1989, 191-198. (trans.: JPRS, UCC-90-004, May 9, 1990, 1-2)

[Manu88] Manuylov, N. V., "On the Approach to Fifth Generation Computers," *Ekonomicheskaya Gazeta* (Aug., 1988).

[Marc78] Marchuk, G. I., Kotov, V. Ye., *A Modular Asynchronous Extendable System (Conception), Part I, Preprint 86*, Computer Center, SO AN SSSR, Novosibirsk, 1978.

[Marc78b] Marchuk, G. I., Kotov, V. Ye., *A Modular Asynchronous Extendable System (Conception), Part II, Preprint 87*, Computer Center, SO AN SSSR, Novosibirsk, 1978.

[Marc86] "Gury Ivanovich Marchuk, President of the USSR Academy of Sciences," *Pravda* (Oct. 17, 1986), 3. (trans.: CDSP, Vol. XXXVIII, No. 42, 28)

[Marc87] Marchuk, G. I., "Opening Address of the President of the Academy of Sciences of the USSR, G. I. Marchuk," *Vestnik Akademii Nauk SSSR* v. 7 (1987), 3.

[Mark86] Markov, S., Lazarov, V., "Specialized Highspeed Computer Systems," *Vychislitel'naya Tekhnika Sotsialisticheskikh Stran* 19 (1986), 3-12.

[Mark88] Markus, M. L., Robey, D., "Information Technology and Organizational Change: Causal Structure in Theory and Research," *Management Science* v. 34, 5 (May, 1988), 583-598.

[Mark92] Markoff, J., "Russian Computer Scientists Hired by American Company," *New York Times* (Mar. 3, 1992), 1.

[Mark92b] Markoff, J., "Russian Research Pact for Sun Microsystems," *New York Times* (September 2, 1992), D3.

[Mche81] McHenry, W. K., ''The Role of Western Computers in Soviet Energy Industries,'' Master of Science thesis, School of Engineering, University of Virginia, Charlottesville, VA, 1981.

[Mche85] McHenry, W. K., ''The Absorption of Computerized Management Information Systems in Soviet Enterprises,'' Ph.D. Dissertation, University of Arizona, 1985.

[Mche86] McHenry, W. K., Goodman, S. E., "MIS in USSR Industrial Enterprises: The Limits of Reform from Above," *Communications of the Association for Computing Machinery* v. 29, 11 (Nov., 1986).

[Mche88] McHenry, W. K., "Computer Networks and the Soviet-Style Information Society," Chapter 6 *The Future Information Revolution in the USSR*, Staar, R. F., Ed., Crane Russak & Company, New York, NY, 1988, 85-113.

[Mche88b] McHenry, W. K., ''Photos from Visit to VDNKh Computer Pavillion, Oct., 1988.''

[Mcma86] McMahon, F., ''The Livermore Fortran Kernels: A Computer Test of the Numerical Range,'' Lawrence Livermore National Laboratory report UCRL-53745, Dec. 1986

[Medv92] Medvedev, I. L., *Design of the Kernel of a Parallel Processor Structure*, Institute of Control Problems, Moscow, 1992.

[Medv92b] Medvedev, I. L., BEAR Parallel Supercomputer. Four Year Business Plan, Nov. 30, 1992.

[Meli79] Meliksetyan, R., "Development of First Soviet Matrix Processor, ES-2345," *Kommunist* 299 (Dec. 29, 1979), 1.

[Meln91] Mel'nikov, V. A., Mitropolskiy, Yu. I., Shnitman, V. Z., Ivannikov, V. P., Tomilin, A. N., Gaysaryan, S. S., "High Performance Computer System ''Elektronika SSBIS'','' in *Proceedings of the International Conference Parallel Computing Technologies, Novosibirsk, Sep. 7-11, 1991*, Mirenkov, N. N., Ed., World Scientific Publishing Company, Singapore, 1991, 47-55.

[Merk80] Merkel', G., Yungnikel', Kh G., "Modular Conception in the Designs of the GDR of Devices for Data Preparation and Terminals for ES and SM Computers," *Vychislitel'naya Tekhnika Sotsialisticheskikh Stran* v. 8 (1980), 13-19.

[Metl73] Metlyayev, Yu. V., Vishnevskiy, Yu. L., Makarov, G. P., *Vychislitel'naya tekhnika*, Novosibirsk, 1973, pp.65-88

[Meye82] Meyer, A. D., "Adapting to Environmental Jolts," *Administrative Science Quarterly* 27 (1982), 515-537.

[Meye88] Meyer, Ernest, "A Chip Off the Soviet Bloc," *Supercomputing Review* v. 1, 1 (Oct. 1988), 38-41.

[Meye90] Meyer, A. D., Brooks, G. R., Goes, J. B., "Environmental Jolts and Industry Revolutions: Organizational Responses to Discontinuous Change," *Strategic Management Journal* v. 11 (1990), 93-110.

[Mikh84] Mikhalevich, V. S., Kanygin, Yu. M., "Informatics: It's Time Has Come," *Pravda* (July 31, 1984), 2.

[Mikh86] Mikhalevich, V. S., Kapitonova, Yu. V., Letichevskiy, A. A., "Macropipelining of Computation," *Kibernetika* 3 (May-Jun., 1986), 3-10.

[Mikh88] Mikhalevich, V. S., Kapitonova, Yu. V., Kukharchuk, A. G., Klimenko, V. P., Letichevskiy, A. A., Losev, V. D., Molchanov, I. N., Pogrebinskiy, S. B., "Structural-Architectural Solutions of Multiprocessor Computers with Macro-Pipeline Data Processing," in *EVT Electronic Computer Technology*, Przhiyalkovskiy, V. V., Ed., Radio i svyaz', Moscow, v. 2 (1988), 151-160.

[Mile80] Miles, R. H., *Macro Organizational Behavior*, Goodyear Publishing Company, Inc., Santa Monica, 1980.

[Mint88] Mintzberg, H., "Opening Up The Definition Of Strategy," in *The Strategy Process*, Quinn, J. B., Mintsberg, H., James, R. M., Eds., Prentis Hall, Englewood Cliffs, 1988, 13-20.

[Mnw84] "Soviet Supercomputer," *Moscow News* 49 (Dec. 16, 1984), 10.

[Molc85] Molchanov, I. N., "Applied Software for an ES Multiprocessor Computer," in *Computer Processes and Systems*, Marchuk, G. I., Voyevodin, V. V., Bakhvalov, N. S., Belykh, L. N., Dymnikov, V. P., Kotov, V. Ye, Kuznetsov, Yu A., Lebedev, V. I., Sarkisyan, A. S., Eds., Nauka, Moscow, v. 2 (1985), 99-108.

[Morg86] Morgan, G., *Images of Organization*, Sage Publications, Inc., Beverly Hills, 1986.

[Moro75] Moroz, F., "How Are You, Minicomputer?," *Anilinokrasochnaya Promyshlennost* (Moscow) 6 (1975), 1-3.

[Mosc84] ''Production of New Fast Computers'', Moscow World Service Broadcast, Dec. 5, 1984.

[Moto82] *Fifth Generation Computer Systems: Proceedings of the International Conference on Fifth Generation Computer Systems (Tokyo, Japan, October 19-22, 1981)*, Moto-oka, T., Ed., North Holland, New York, 1982.

[Mpov91] ''ES-1066.90 Computer'', brochure obtained from MPOVT, Feb. 1991.

[Muen81] Muench, W., "ES-1055M," *Rechentechnik Datenverarbeitung* (GDR) v. 18, 1 (Feb., 1981), 1. (trans.: JPRS, EER/81621, 15-16)

[Muku93] Mukund, P. R., McDonald, J. F., "MCM: The High-Performance Electronic Packaging Technology," *Computer* (IEEE) v. 26, 4 (Apr., 1993), 10-12.

[Musa86] Musayelyan, V., "Millions of Operations Per Second," *Kommunist* (Yerevan) (Jun 14, 1986), 1. (trans.: JPRS, UCC-87-004, Mar. 31, 1987, 10)

[Mvke80] ''Multiprocessor Computer Complex (MVK) 'El'brus-1''' (Mnogoprotsessornyy Vycheslitel'nyy Kompleks 'El'brus-1'), ITMVT Brochure, AN SSSR, 1980.

[Mvs89] Multiprocessor Computer Systems '89, Sep. 11-16, 1989, Divnomorsk.

[Myam86] Myamlin, A. N., Smirnov, V. K., Golovkov, S. L., "A Specialized Symbol Processor," in *Fifth Generation Computer Architectures. Proceedings of the IFIP TC 10 Working Conference on Fifth Generation Computer Architectures, Manchester, U.K., Jul. 15-18, 1985*, Woods, J. V., Ed., North-Holland, Amsterdam, 1986, 301-318.

[Myas82] Myasnikov, V. A., "Present Status and Prospects for the Development of Computer Technology, Computer Networks, and Data Transmission Systems," in *The Design of Computer Complexes, Systems, and Networks*, Myasnikov, V. A., Ed., Moscow Order of Lenin and of the October Revolution Energy Institute,, 572 (1982), 3-8. (trans.: JPRS, UCC-84-001-L)

[Myas84] Myasnikov, V. A., Ignat'yev, M. B., Sheynin, Yu. Ye., "Architecture of a Modular Multi-Microprocessor Computer System," *Kibernetika* 3 (May-Jun., 1984), 48-53.

[Naba91] Nabatov, A. S., Largin, V. A., Conversation with PW, Sep. 19, 1991.

[Nari85] *Dialogue Interaction and Representation of Knowledge* Narin'yani, A. S., Ed., Novosibirsk, 1985.

[Neft91] ‘‘Multiprocessor Seismic Data Processing System:SOS-PS-2000,’’ Brochure, Neftegeofizika, 1991.

[Nels77] Nelson, R. R., Winter, S. G., “In Search of Useful Theory of Innovation,” *Research Policy* 6 (1977), 36-76.

[Nemo88] Nemov, A., “Academy’s Head Views Results of Restructuring, Rejuvenation Campaigns,” *Sovetskaya Rossiya* 135 (Jun. 11, 1988), 1-2.

[Netm88] “Editors of Byte Magazine Meet with the Soviet Delegation, ” *Netmonth* v. 13, 1 (Jul., 1988).

[Newm91] Newman, B., “Broken Crutch. These Days, Bulgaria Longs for Its Old Ties To the Soviet Union,” *Wall Street Journal* v. 124, 84 (Apr. 30, 1991), A1,A5.

[Niik92] Promotional brochure from NII Kvant, December, 1992. [Niim89b] “VLSI Macroprocessor Set,” brochure, Scientific Research Institute for Multiprocessor Computing Systems, 1989.

[Niim89] “Personal Multiprocessing Supercomputer (PMS),” brochure, Scientific-Research Institute of Multiprocessor Computing Systems, Taganrog, 1989.

[Niim89b] “Personal Macrocomputer,” brochure, Scientific Research Institute of Multiprocessor Computing Systems, 1989.

[Niim89c] “The PVK-460 Problem-Oriented Computer System,” brochure, Scientific Research Institute of Multiprocessor Computer Systems, Taganrog, 1989.

[Niko79] Nikolov, G. P., Lazarov, V. D., Dashkalov, G. P., Ivanova, J. V., Kirov, K. D., “ES-1035 System with Special Matrix Processor,” *Szamitastechnika* (Budapest) (Nov. 1979), 7. (trans.: JPRS, CCAT/75012)

[Niko81] Nikolov, G. P., Lazarov, V. D., Daskalov, P. P., Ivanova, I. V., Kirov, K. D., “The ES-2335 Array Processpr for the ES-1035 Computer System,” *Vychislitel’naya Tekhnika Sotsialisticheskikh Stran* v. 9 (May, 1981), 47-50.

[Niko82] Nikolov, G., “New Developmental Trends in Bulgarian Computers,” *Szamitastechnika* (Budapest) (Jan., 1982), 3.

[Nits91] “International Non-governmental Organization, Department of the All-World Laboratory in the USSR,” brochure, Scientific Research Institute of Multiprocessor Computing Systems, 1991.

[Nolt88] Nolting, L. E., *The Soviet View on the State of Technological Innovation in the USSR*, Center for International Research, U.S. Department of Commerce, 1988.

[Nove86] Nove, A., *The Soviet Economic System*, T , Allen & Unwin, Inc., Boston, 1986.

[Nrc93] *Joint Statement of the Delegations of the Russian Academy of Sciences and U.S. National Academy of Sciences on Dual Use Technologies and Export Administration*, National Academy Press, 1993.

[Ntr85] "Managing the Multi-Institute Project on 5th Generation Computers," *NTR: Problemy i Resheniya* 4 (Jul. 2-15, 1985), 6.

[Nyt770624] "Big-Computer Sale To Soviet Is Barred," *New York Times* (June 24, 1977), A3.

[Ohra84] Ohran, R., "Lilith and Modula-2," *Byte* v. 9, 8 (Aug., 1984), 181-192.

[Orga73] Organick, E. I., *Computer Systems Organization: the B5700/B6700 Series*, Academic Press, New York, 1973.

[Pano85] Panov, Ye., "The Branch Makes a Choice," *Sotsialisticheskaya Industriya* (July 19, 1985), 2.

[Parr80] Parrott, B., "The Organizational Environment of Soviet Applied Research," in *The Social Context of Soviet Science*, Lubrano, L. L., Solomon, S. G., Eds., Westview Press, Boulder, 1980, 69-100.

[Parr83] Parrott, B., *Politics and Technology in the Soviet Union*, MIT Press, Cambridge, 1983.

[Pent82] Pentkovskiy, V. M., *El'brus Machine Language*, Nauka, Moscow, 1982.

[Pent90] Pentkovskiy, V. M., Zaytsev, A. I., Kovalenko, S. S., Feld'man, V. M., "The Structure of a High-Performance Microprocessor with Hardware Support for High-Level Languages," *Programmirovaniye* v. 1990, 3 , 24-35.

[Perr67] Perrow, C., "A Framework for the Comparative Analysis of Organizations," *American Sociological Review* v. 32 (1967), 194-208.

[Pert86] Pertsovskiy, S. A., Butov, V. V., *Severodonetsk*, Donbas, 1986.

[Petr79] Petrov, B. N., "Most Important Directions for Increasing the Effectiveness of the Use of Computer Technology," *Avtomatika i Vychislitel'naya Tekhnika* (Riga) 2 (1979), 1-2.

[Pfei90] Pfeiffer, W., Alagar, A., Kamrath, A., Leary, R. H., Rogers, J., "Benchmarking and Optimization of Scientific Codes on the Cray X-MP, Cray-2, and SCS-40 Vector computers.," *The Journal of Supercomputing* 4 (1990), 131-152.

[Pipk83] Pipko, D., "PS-2000 High Speed Multiprocessor Complex," *Sotsialisticheskaya Industriya* 131 (June 9, 1983), 4.

[Pivo92] Pivovarov, G. Yu., "On the Question of the Structure of the Firm," *Impul's* 3 (Nov., 1992), 3.

[Pogr87] Pogrebinskiy, S. B., "Developments in the Architecture of High-Speed Computers," *Upravlyayushchiye Sistemy i Mashiny* 6 (Nov.-Dec., 1987), 39-46. (trans.: JPRS, UCC-88-003, Oct. 4, 1988, 1-13)

[Pois90] "We Implement:," *Poisk* 25 (Jun. 22-28, 1990), 2.

[Poll90] Pollack, A., "The Custom-Made Supercomputer," *New York Times* (May 27, 1990), F4.

[Pono83] Ponomarev, V. M., Plyasnin, V. U., Torgashev, V. A., *Distributed Computing and Machines with Dynamic Architecture*, Leningrad Research Computing Center, Leningrad, 1983.

[Pono87] Ponomarev, V. M., Torgashev, V. A., Didenko, V. D., "Distributed Computational Systems With Dynamic Architecture in Integrated Production Systems," in *Integrated Production Systems: A Collection of Articles*, Ponomarev, V. M., Aleksandrov, V. V., Barilov, A. A., Belyakov, I. P., Al, Et, Eds., Mashinostroyeniye, Leningrad, 1987, 27-34.

[Pran83] Prangishvili, I. V., Vilenkin, S. Ya., Medvedev, I. L., *Parallel Computer Systems with Common Control*, Energoatomizdat, Moscow, 1983. (trans.: JPRS, UCC-86-004-L, Feb. 26, 1986, 1-29)

[Pras89] Prasad, N. S., *IBM Mainframes. Architecture and Design*, McGraw-Hill Book Company, New York, 1989.

[Prat90] Pratt, C. R., 12 Mflops Array Processors, Foreign Availability Assessment, U.S. Department of Commerce, Bureau of Export Administration, May, 1990.

[Prav740508] "It Will Be Easier to Set Up Automated Management Systems," *Pravda* (May 8, 1974), 2. (trans.: CDSP, Vol. 26, No. 19, 1974, 20)

[Prav851219] "Comprehensive Program (Scientific-Technical Progress of the Country - And the Members of SEV Until the Year 2000) (The Basic Positions)," *Pravda* 353 (Dec. 19, 1985), 1,4. (trans.: SNAP, Jan. 1, 1986, 4-5)

[Prav850827] "Electronic Giant," *Pravda* (Feb. 22, 1985), 3. (trans.: JPRS, UCC-85-004, Aug. 27, 1985, 54-55)

[Prav860709] "On Serious Violations of Party and State Discipline by Certain Officials of the Ministry of the Machine Tool and Tool Industry and Other Ministries in Supplying the Economy With Advanced Forge and Press Equipment," *Pravda* (July 9), 3. (trans.: CDSP, Vol. XXXVIII, No. 27, Aug. 6, 1986)

[Prav870701] "Law of the Union of Soviet Socialist Republics on the State Enterprise (Association).," *Pravda* (July 1, 1987), 1-4. (trans.: CDSP, Vol. XXXIX, No. 30, Aug. 26, 1987, Vol. XXXIX, No. 31, Sep. 2, 1987)

[Prog86] "Synopses of Articles in Programirovanniye, 4, 1986," *Programmirovaniye* 4 (1986), 95-96.

[Prsu70] "Sirena Teleautomatic Mass Service System," *Pribory i Sistemy Upravleniya* 9 (1970), 1.

[Prsu87] "PS 1001 Computer Complex," *Pribory i Sistemy Upravleniya* 9 (Sept., 1987), 50.

[Przh89] Przhiyalkovskiy, V. V., "The Program for the Development of Hardware and Software for the Unified Series Computers," *Vychislitel'naya Tekhnika Sotsialisticheskikh Stran* 26 (1989), 32-40.

[Psas87] "Abstracts from Computer and Office Mechanization Facilities," *Pribory, Sredstva Avtomatizatsii i Sistemy Upravleniya: TS-2. Sredstva Vychislitel'noy Tekhniki i Orgtekhniki. Referativniy Sbornik* 6 (1987).

[Rabd88] "Use of a Supercomputer in a Research Center," *Rabotnichesko Delo* (Sofia) (Jan. 20, 1988), 1,3. (trans.: JPRS, EER-88-011, Feb. 12, 1986, 6)

[Rahm89] Rahmeyer, F., "The Evolutionary Approach to Innovation Activity," *Journal of Institutional and Theoretical Economics* v. 145, 2 (1989), 275-297.

[Rajm88] Rajman, J., "Data Processing in East Europe: The Robotron Line," *Zero Un Informatique Hebdo* (Paris) (May 2, 1988), 90. (trans.: JPRS, EST-88-005, Jul. 27, 1988, 49-51)

[Raki91] Rakitin, V. G., Keynote Address, 35 Year Jubilee, Scientific Research Institute of Control Computers, Severodonetsk, Sep. 18, 1991.

[Rama77] Ramamoorthy, C. V., "Pipeline Architecture," *Computing Surveys* (ACM) v. 9, 1 (Mar., 1977), 61-102.

[Rau89] Rau, B. Ramakrishna, Yen, David W. L., Yen, Wei, Towle, Ross A., "The Cydra 5 Departmental Supercomputer. Design Philosophies, Decisions, and Trade-offs," *Computer* (IEEE) v. 22, 1 (Jan., 1989), 12-35.

[Rdd82] ''Disk Drives Performance, Technology, and Architecture Status and Outlook,'' report by Memory and Media Sub-Committee, PTAC, Nov., 12, 1982.

[Reag81] "Text of Reagan's Announcement of Sanctions," *New York Times* (Dec. 30, 1981), A6.

[Reza71] Rezanov, V. V., Vinokurov, V. G., Kostelyanskij, V. M., Itenberg, I. I., Kadyrov, V. M., Nabatov, A. S., Lekhnova, G. M., Savvov, V. I., Smolij, V. G., Pronin, G. A., Margulis, D. S., "New Facilities Added to ASVT Line," *Mekhanizatsiya i Avtomatizatsiya Upravleniya* (Kiev) 4 (1971), 23-53.

[Reza83] Rezanov, V. V., Kostel'yanskiy, V. M., "Computer Technology Facilities for Controlling Technological Processes and Processing Geophysical Data - Status and Development Prospects (SM-1210 and PS-3000),' ' *Pribory i Sistemy Upravleniya* 1 (Jan., 1983), 7-12. (trans.: JPRS, CCAT/L/11330, 37-53)

[Reza92] Rezanov, V. V., “Structure–Oriented Towards Concrete Tasks,” *Impul’s* 3 (Nov., 1992), 1-2.

[Rich80] Richards, T. J., ‘‘An Examination of the Issues Affecting ADP Technology Transfer to the Soviet Union,’’ Ph.D. Thesis, George Washington University, Washington, D.C., Nov. 1980.

[Robo88] ‘‘Scientific-Technical Computer ES-1055.M with MAMO.1M’’, undated Robotron brochure, obtained in 1988.

[Roma90] Romano, C. A., “Identifying Factors Which Influence Product Innovation: A Case Study Approach,” *Journal of Management Studies* v. 27, 1 (Jan., 1990), 75-95.

[Rudi70] Rudins, G., “Soviet Computers: A Historical Survey,” *Soviet Cybernetics Review* (Jan., 1970), 6-44.

[Rumb75] Rumbaugh, J. E., ‘‘Parallel Asynchronous Computer Architecture for Data Flow Programs,’’ Ph. D. Thesis, MIT, Cambridge, 1975.

[Rumy88] *The Rhythm of Perestroika. The New System of Management of the Economy. 1987.*, Rumyantsev, A. F., Ed., Ekonomika, Moscow, 1988.

[Ryab91] Ryabov, G. G., Babayan, B. A., Conversation with PW, Sep. 25, 1991.

[Ryab91b] Ryabov, G. G., “Toward the Use of Supercomputers in the USSR,” *Programmirovaniye* 5 (Sep.-Oct., 1991), 32-33. (trans.: JPRS, UCC-92-009-L, May 27, 1992, 1-3)

[Ryab92] Ryabov, G. G., Repin, V. M., Conversation with PW, Dec. 17, 1992.

[Ryad88] Ryadchikov, V. A., ‘‘PS-2100 Internal Software System. Language Description,’’ Minpribor, 1988.

[Sagd87] *The ES-1037 - ES-2706 Multiprocessor Computer System: the Experience of Systems Development and Digital Modeling of Non-linear Physical Problems*, Sagdeyev, R. Z., Shevchenko, V. I., Eds., Institute of Space Research, Moscow, 1987.

[Saha81] Sahal, D., *Patterns of Technological Innovation*, Addison-Wesley, Reading, MA, 1981.

[Saha81b] Sahal, D., “Alternative Conceptions of Technology,” *Research Policy* v. 10 (1981), 2-24.

[Saha85] Sahal, D., “Technological Guideposts and Innovation Avenues,” *Research Policy* v. 14 (1985), 61-82.

[Sama85] Samarskyy, A. A., “Problems in the Use of Computer Technology and the Development of Informatics,” *Vestnik Akademii Nauk SSSR* 3 (1985), 57-69.

[Same85] Samedova, M. A., “About the Work of the Inter-Branch Scientific Seminar ’Theory of Information Systems’,” *Programmirovaniye* 2 (1985), 89-90.

[Samo89] Samoylis, S., "Dynamic-Architecture Computer Performs 100 Million Operations/second," *Leningradskaya Pravda* 156 (Jul. 8, 1989), 2. (trans.: SNAP, Jul. 24, 1989)

[Schl83] Schlegel, W. E., Blodszun, A., "1983 Leipzig Spring Fair Computer, Communications Equipment- Microelectronics, Semiconductor Components," *Radio Fernsehen Elektronik* (East Berlin) v. 32, 6 (Jun. 1983), 344-351. (trans.: JPRS, EER/84019, Jun. 8, 1984, 22-44)

[Schm91] Schmitt, R. W., et al., *Finding Common Ground: U S Export Controls in a Changed Global Environment*, National Academy Press, Washington D C, 1991.

[Schr80] Schreyögg, G., "Contingency and Choice in Organization Theory," *Organization Studies* v. 1, 4 (1980), 305-326.

[Schu34] Schumpeter, J., *The Theory of Economic Development*, Harvard University Press, 1934.

[Schu39] Schumpeter, J., *Business Cycles*, McGraw Hill, New York, 1939.

[Scot90] Scott, W. R., "Technology and Structure: An Organizational-Level Perspective," in *Technology and Organizations*, Goodman, P. S., Sproull, L. S., Eds., Jossey-Bass Publishers, 1990, 109-143.

[Sedu79] Sedukhin, S. G., "On an Analysis of the Data Transmission Modes in the SUMMA Computer System," *Vychislitel'nyye Sistemy* 80 (1979), 81-84. (trans.: JPRS, CCAT/L/9692, Apr. 27, 1981, 81)

[Seme82] Semerdzhyan, M. A., Arustamyan, V. Ye., "Armenian Institute Important for Soviet Computer Industry," *Promyshlennost' Armenii* (Yerevan) 6 (Jun., 1982), 25-27. (trans.: JPRS, CCAT/L/11079, 57-61)

[Seme84] Semerdzhyan, M. A., Nalbandyan, Zh. S., *The ES-2345 Array Processor*, Finansy i Statistika, Moscow, 1984.

[Seme85] Semerdzhyan, M. A., "Computer Technology: Stages of Development," *Promyshlennost' Armenii* (Yerevan) 4 (April, 1985), 27-30.

[Seme88] Semerdzhyan, M. A., Nalbandyan, Zh. S., Gasparyan, L. Kh., *Principles of Operations and Systems Software for the ES 2700 Array Processor*, Nauka, Moscow, 1988.

[Seme92] Semendyayev, S., "East Europe: New Economic Area," *Rossiskaya Gazeta* (Feb. 26, 1992), 1. (trans.: JPRS, USR-92-031, Mar. 20, 1992, 51-52)

[Sese88] "Soviet Union Update," *SES Exchange* 5 (Dec., 1988), 7-10.

[Shaw85] Shaw, S. J., "U.S. microelectronics top Soviet shopping lists," *Mini-Micro Systems* v. 18, 16 (December, 1985), 44.

[Shel73] Shelikov, A. A., *Computers: A Handbook*, Energiya, Moscow, 1973.

[Sher92] Shershul'skiy, V. I., "The Electronics Industry: The Cold Summer of 1992," *Computerworld Moscow* 28 (Aug. 25, 1992), 1,8-9.

[Sher92b] Shershul'skiy, V., "The Last Supercomputer of the Soviet Union," *Computerworld Moscow* 40-41 (Nov. 9, 1992), 1.

[Shka81] Shkabardnya, M. S., "Instrument Building for the 26th CPSU Congress," *Pribory i Sistemy Upravleniya* 2 (Feb., 1981), 2-5. (trans.: JPRS, CCAT/L/9703)

[Shka86] Shkabardnya, M., "Minpribor Streamlines its System and Creates Big R&D Center," *Sotsialisticheskaya Industriya* 45 (Feb. 22, 1986), 2. (trans.: SNAP, Mar. 18, 1986)

[Shka86b] Shkabardnya, M. S., "Speech by Comrade M. S. Shkabardnya, USSR Minister of Instruments, Automation Equipment and Control Systems (27th Party Congress).," *Pravda* (Feb. 27, 1986). (trans.: CDSP, Vol. XXXVIII, No. 9, Apr. 2, 1986, 9-10)

[Shka87] Shkabardnya, M., "Industry Strategy in Restructuring Process Discussed.," *Planovoye Khozyaystvo* (November 6, 1987), 9. (trans.: JPRS, UEQ-87-012, Nov. 6, 1987)

[Shka88] Shkabardnya, M., "The Lessons of Economic Accounting," in *The Rhythm of Perestroika. The New System of Management of the Economy. 1987.*, Rumyantsev, A. F., Ed., Ekonomika, Moscow, 1988, 78-81.

[Sini87] Sinitsyn, N. V., Petropavlovskiy, V. N., Nikitin, A. M., "Automated Systems of Scientific Research," *Radioelektronika i Svyaz'* 1 (1987), 1-64. (trans.: JPRS, UCC-87-017, Sep. 23, 1987, 32-81)

[Site93] Sites, R. L., "Alpha AXP Architecture," *Communications of the Association for Computing Machinery* v. 36, 2 (Feb., 1993), 33-44.

[Smit78] Smith, B., "A Pipelined Shared Resources MIMD Computer," in *IEEE Proceedings of the 1978 International Conference on Parallel Processing*, 1978, 6-8.

[Smit86] Smith, J. E., Weiss, S., Pang, N. Y., "A Simulation Study of Decoupled Architecture Computers," *IEEE Transactions on Computers* v. C-35, 8 (Aug., 1986), 692-702.

[Smit87] Smith, K. G., Grimm, C. M., "Environmental Variation, Strategic Change and Firm Performance: A Study of Railroad Deregulation," *Strategic Management Journal* v. 8 (1987), 363-376.

[Smit89] Smith, J. E., "Dynamic Instruction Scheduling and the Astronautics ZS-1," *Computer* (IEEE) (July, 1989), 21-35.

[Smit92] Smith, Michael J. S., "Synthesizing ASICs," *IEEE Spectrum* (29) n11, Nov. 1992, 44-48.

[Smol83] Smolov, V. B., Shumilov, L. A., Novoselov, V. V., "Address Handling and Architecture of Large-Scale Integrated Bit-Slice Microprocessors," *Elektronnoye Modelirovaniye* 1 (Jan.-Feb. 1983), 12-15. (trans.: JPRS, CCAT/L/11347, Jun. 1, 1983, 47-53)

[Smyk85] Smyk, V., "Don't Digress, Computer!," *Soviet Life* (Washington) 6 (June, 1985), 19-21.

[Soan82] *Siberian Division of the USSR Academy of Sciences*, Vneshtorizdat, Novosibirsk, 1982.

[Soan87] *Novosibirsk Science Center*, Vneshtorizdat, Novosibirsk, 1987.

[Soko85] Sokol, A., "Elektronika MK-72 Workplace and Teaching Microcomputer," *Pravda Ukrainy* (Kiev) 209 (Sept. 11, 1985), 4. (trans.: SNAP, Sep. 25, 1985)

[Sots790913] "New M-10 Computer Exceeds Five Million Ops./s," *Sotsialisticheskaya Industriya* 13 (Sept. 13, 1979), 6.

[Sots880515] "Computers of the Future are Being Built," *Sotsialisticheskaya Industriya* 113 (May 15, 1988), 1.

[Spal90] Spalvin', A. P., "Introduction," in *Computer Technology and Boundary Problems. Specialized Processors for Parallel Activities*, Kokareva, A., Kotovich, L. L., Maksimov, M. M., Eds., Riga Polytechnical Institute, Riga, 1990, 3.

[Stap85] Stapleton, R. A., Goodman, S. E., "Microcomputing in the Soviet Union and Eastern Europe," *Abacus* (NY) v. 3, 1 (Fall 1985), 6-22.

[Stap88] Stapleton, R. A., ''Personal Computing in the CEMA Community: A Study of International Technology Development and Management,'' Ph.D. Dissertation, The University of Arizona Management Information Systems Department, Summer 1988.

[Suga77] Sugarman, R., "Soviets, CDC Considering Making Computer Capable of Weapons Development," *Electrical Engineering Times* (Nov. 28, 1977), 4,30.

[Sukh87] Sukhodayev, A., "Branch Science," *Ekonomicheskaya Gazeta* 22 (May 1987), 9.

[Supe91] "Problems in the Development of High Performance Computing Systems," Nepetsino, Oct. 22, 1991.

[Supe91b] "Problems in the Development of High Performance Computing Systems," Nepetsino, Oct. 23, 1991.

[Supe91c] "Problems in the Development of High Performance Computing Systems," Nepetsino, Oct. 24, 1991.

[Tadz77] Tadzhiev, D., Musabayev, B., *History of the Creation and Development of Computing Technology in the USSR*, Tashkent, 1977.

[Tala90] Talapov, A. L., Andreychenko, V. B., Dotsenko, V. S., Shchur, L. N., "Dedicated Processor for Studying Ising Model on Random Lattice," *Pis'ma v Zh. Eksp. Teor. Fiz.* v. 51, 3 (Feb. 10, 1990), 161-163.

[Tala90b] Talapov, A. L., Andreychenko, V. B., Dotsenko, V. S., Shchur, L. N., *Ising Random Lattice Special Purpose Processor, Preprint CP/89-01*, Landau Institute of Theoretical Physics, Jan., 1990.

[Tamm88] Tamm, B. G., Tani, H. J., Tyugu, E. Kh, "Integration of Hardware and Software in New Generation Workstations," in *Man-Machine Interface*, Oulu, 1988, 457-462.

[Targ80] Targowski, A., *Informatics: Models of Systems and Development*, Panwstwowe Wydawnictwo Ekonomiczne, Warsaw, 1980.

[Tass78] ''Successes of Soviet Electronics'', TASS International Service Broadcast, July 21, 1978, 1433 GMT.

[Tass88] Moscow Tass. 2014 GMT, 25 April, 1988.

[Tcha92] Tchashchin, K. V., "High-Performance Computing in the Russian Petroleum Industry," *Supercomputing Review* v. 5, 2 (Feb., 1992), 32-33.

[Thac93] Thacker, C. P., Conroy, D. G., Stewart, L. C., "The Alpha Demonstration Unit: A High-Performance Multiprocessor," *Communications of the Association for Computing Machinery* v. 36, 2 (Feb., 1993), 55-67.

[Thom83] Thompson, J. C., Vidmer, R. F., *Administrative Science and Politics in the USSR and the United States: Soviet Responses to American Management Techniques, 1917-Present*, Praeger, New York, 1983.

[Tich83] Tichy, N. M., *Managing Strategic Change*, John Wiley & Sons, New York, 1983.

[Torg84] "Expandable Multicomputer System for Parallel Process Data Processing Organization," In *Design and Utilization of Hierarchical Multicomputer Systems*, Torgov, Yu I., Ed., Computer Center of the USSR Academy of Sciences, Moscow, 1984. (trans.: JPRS, UCC-85-001-L, Jan. 17, 1986, 110-113)

[Torg88] Torgashev, V. A., Plyusnin, V. U., Ponomarev, V. M., "Multiprocessors with Dynamic Architecture," in *EVT Electronic Computer Technology*, Przhiyalkovskiy, V. V., Ed., Radio i svyaz', Moscow, v. 2 (1988), 172-182.

[Torg89] Torgashev, V. A., Plyusin, V. U., "Processors with Dynamic Architecture," in *Electronization of the National Economy*, Gorshkov, N. V., Ed., State Committee on Computer Technology and Informatics, Moscow, 1-2 (1989), 27-36.

[Torn83] Tornatsky, L. G., Eveland, J. D., Boylan, M. G., Hetzner, W. A., Johnson, E. C., Roitman, D., Schneider, J., *The Process of Technological Innovation: Re-*

viewing the Literature, National Science Foundation, Washington, D.C., 1983.

[Tosi84] Tosi Jr., Henry L., Slocum.Jr, John W., "Contingency Theory: Some Suggested Directions," *Journal of Management* v. 10, 1 (1984), 9-26.

[Trap79] Trapeznikov, V. A., "Raising Efficiency of Automated Management Systems," *Avtomatika i Vychislitel'naya Tekhnika* (Riga) 6 (Nov.-Dec., 1979), 21-28. (trans.: JPRS, CCAT/L/8840, 59)

[Trap81] Trapeznikov, V. A., Prangishvili, I. V., Novokhatniy, A. A., Rezanov, V. V., "Dispatching Geophysical Computer Complexes Based on the PS-2000 Multiprocessor Computer," *Pribory i Sistemy Upravleniya* 2 (Feb., 1981), 29-31.

[Trel82] Treleaven, P. C., Brownbridge, D. R., Hopkins, R. P., "Data-Driven and Demand-Driven Computer Architecture," *Computing Surveys* (ACM) v. 14, 1 (Mar., 1982), 93-143.

[Tryb88] "Recognition Has to Be Earned," *Trybuna Ludu* 144 (June 22, 1988), 1,3.

[Tryb88b] *Trybuna Ludu* (Jun. 22, 1989), 1.

[Tsan79] Tsang, F. R., "Operating System of the Special Processor of the El'brus-1 Multiprocessor Computing System Compatible with the DISPAK Operating System.," *Referativnyy Zhurnal, Avtomatika, Telemekhanika i Vychislitel'naya Tekhnika* 1 (1979). (trans.: JPRS, CCAT/L/8876, Jan. 22, 1980, 88)

[Turc86] Turchin, V. F., Bibliography of Publications on the REFAL Programming Language, 1986. In Russian.

[Tyap82] Tyapkin, M. V., "Specialized Integrated BESM-6 Processor in El'brus Multiprocessor Computer," in *Raising the Effectiveness of the Use of High-Throughput Computers: Materials on Software*, Kopytov, M. A., Zel'dinova, S. A., Kobelev, V. V., Lipatova, Ye F., Mikhaylov, G. M., Samofalov, V. V., Eds., ANSSSR, Urals Science Center, Institute of Mathematics and Mechanics, Sverdlovsk, 1982, 4-9. (trans.: JPRS, UCC-84-007-L, May 30, 1984, 1-4)

[Tyug70] Tyugu, E. Kh, "Problem solving on computational models," *Journal of Applied Mathematics and Mathematical Physics* v. 10 (1970).

[Uprs81] "Put the Decisions of the 26th CPSU Congress into Effect!," *Upravlyayushchiye Sistemy i Mashiny* 2 (Mar.-Apr., 1981), 2. (trans.: JPRS, CCAT/78456)

[Usdi91] Usdin, S., Tchashchin, K., "Soviet Producing New Supercomputer, Plan to Make RISC Workstation," *Supercomputing Review* v. 4, 2 (Feb., 1991), 29-30.

[Vadi84] Vadimov, O., "Radio Institute's Microprocessors in All-Union Exhibition," *Sovetskaya Russiya* 57 (March 6, 1984), 1.

[Valk87] Val'kovskiy, V. A., Letichevskiy, A. A., Malyshkin, V. E., "Scientific Problems of the Creation of a New Generation of Computers," *Upravlyayushchiye Sistemy i Mashiny* 1 (Jan.-Feb., 1987), 124-126.

[Varc86] Varchenko, V. S., Natanson, L. G., "The Development and Realization of Vector Fortran," *Programmirovaniye* 4 (Jul.-Aug., 1986), 47-58.

[Vasi88] *The CPSU on Restructuring. A Collection of Documents.*, Vasil'evskaya, Ye V., Ed., Politizdat, 1988.

[Vdnk86] Printout of a computer search done at VDNKh, Mar. 9, 1986.

[Vdov81] Vdovin, A. A., "The Principles of Designing Collective-Use Functionally Distributed Systems," *Pribory i Sistemy Upravleniya* 2 (Feb., 1981), 16-20. (trans.: JPRS, CCAT/L/9703, Apr. 30, 1981, 151-162)

[Vegd84] Vegdahl, S. R., "A Survey of Proposed Architectures for the Execution of Functional Languages," *IEEE Transactions on Computers* v. C-33, 12 (Dec., 1984), 1050-1071.

[Veli85] Velikhov, Ye., "On the Way to New Computers," *Rabotnichesko Delo* (Sofia) (Jan. 11, 1985), 4. (trans.: JPRS, CCAT/L/85009L, Apr. 4, 1985, 100-102)

[Veli87] Velikhov, Ye. P., "Electronization of the National Economy: Tasks and Prospects," in *Informatics and Scientific-Technical Progress*, Yermolayev, N. A., Ed., Nauka, Moscow, 1987, 5-13.

[Veli89] Velikhov, Ye. P., "Talk at Annual General Meeting of the USSR Academy of Sciences," *Vestnik Akademii Nauk SSSR* 8 (1989), 21-28.

[Vino80] Vinogradov, V., "Smaller, Yes But Better (Penza 29)," *Pravda* 90 (Mar. 30, 1980), 1.

[Vish84] Vishnevsky, Yu. A., Kotov, V. Ye., Marchuk, A. G., "A Modular Asynchronous Extendable System," *Kibernetika* 3 (May-Jun., 1984), 22-29. (trans.: JPRS, UCC-84-018-L, Dec. 17, 1984, 8-24)

[Vish85] Vishnevskiy, Yu. L., "Principles of the MARS System in a High-Performance Processor Architecture," in *Cybernetics and Computing Technology*, v. 1 (1985), 79-86.

[Vnes89] "The Dedicated Processor ES-2701," brochure number 6439MB from Vneshtorgizdat, 1989.

[Vnes89b] "Multiprocessor Computer Systems ES-1766," brochure number 6434MB from Vneshtorgizdat, 1989.

[Vtp89] "Anniversary International Exhibition," *Vychislitel'naya tekhnika i yeye Primeneniye* 11 (Nov., 1989), 4-21. (trans.: JPRS, UCC-90-004, May 9, 1990)

[Vyrk78] Vyrk, A., "Estonian Academy Plans Time Sharing System Using El'brus-1 Computer," *Sovetskaya Estoniya* 142 (July 12, 1978), 2.

[Vyrk79] Vyrk, A., "Cybernetics Institute Heads Development of El'brus-1 Computer System," *Sovetskaya Estoniya* 54 (March 4, 1979), 2.

[Vyzh90] Vyzhikovski, R., Kanevskiy, Yu. S., "The Design of Systolic Processors for a Numerically Stable Solution for Systems of Linear Equations Using the Gaussian Method," *Elektronnoye Modelirovaniye* v. 12, 2 (1990), 29-36.

[Wirt84] Wirth, N., "History and Goals of Modula-2," *Byte* v. 9, 8 (Aug., 1984), 145-152.

[Wolc88] Wolcott, P., Goodman, S. E., "High-Speed Computers of the Soviet Union," *Computer* (IEEE) v. 21, 9 (Sept., 1988), 32-41.

[Wolc90] Wolcott, P., Goodman, S. E., "Soviet High-Speed Computers: The New Generation," in *Proceedings Supercomputing '90*, IEEE Computer Society Press, Los Alamitos, CA, 1990, 930-939.

[Wolc91] Wolcott, P., "Soviet and Eastern European Computer Peformance: Results of Benchmark Tests," *Mosaic Technical Report* 1991-003-I (Mar. 15, 1991).

[Wolf91] Wolfe, A., Shen, J. P., "A Variable Instruction Stream Extension to the VLIW Architecture" In *ASPLOS-IV Proceedings: Fourth International Conf. on Architectural Support for Programming Languages and Operating Systems* (Santa Clara, Apr. 8-11), pp. 2-4.

[Wood65] Wood, Joan, *Industrial Organization: Theory and Practice,* Oxford University Press, London, 1965.

[Wsj790328] "Tass Signs Contract with a French Firm For Computer System," *Wall Street Journal* (Mar. 28, 1979), 32.

[Yasm85] Yasmann, V., "Soviet Leaders Grapple With the Scientific and Technological Revolution," *Radio Liberty Research Bulletin* 192/85 (Jun. 14, 1985), 1-6.

[Yers80] Yershov, A. P., Shura-Bura, Mikhail R., "The Early Development of Programming in the USSR," in *A History of Computing in the Twentieth Century*, Metropolis, N., Howlett, J., Rota, Gian-Carlo, Eds., Academic Press, New York, 1980, 137-196.

[Yers80b] Ershov, A. P., *A. P. Ershov: The British Lectures*, Samet, P. A., Ed., Heyden & Son Ltd., London, 1980.

[Yers86] Yershov, A., "Put to Life Decisions of 27th CPSU Congress: Information Technology Time," *Znaniye-Sila* 3 (Mar. 1986), 4-5.

[Yevr62] Yevreinov, E. V., Kosarev, Yu. G., *On the Possibility of Constructing High Performance Computer Systems*, SO AN SSSR, Novosibirsk, 1962.

[Yevr74] Yevreinov, E. V., Prangishvili, I. V., *Digital Automatons with a Tunable Structure (Homogeneous Environment)*, Energiya, Moscow, 1974.

[Yevr81] Yevreinov, E. V., *Homogeneous Computer Systems, Structures and Environments*, Davydova, N. G., Kuznetsova, G. Z., Eds., Radio i svyaz', Moscow, 1981.

[Yin89] Yin, R. K., *Case Study Research: Design and Methods*, 2nd ed., Sage Publications, Newbury Park, 1989.

[Yush81] Yushchenko, Ye. L., Tseytlin, G. Ye., "Third All-Union Teaching Seminar ''Parallel Programming and High Performance Structures''," *Kibernetika* 3 (May-June 1981), 138-143. (trans.: JPRS, CCAT/L/11086, 117,126,131)

[Yush91] Yushchenko, Ye. L., Tseytlin, G. Ye., "Tenth All-Union Seminar on Parallel Programming and High-Performance Systems," *Kibernetika i sistemnyy analiz* 5 (Sep.-Oct., 1991), 3-5. (trans.: JPRS, UCC-92-005-L, Mar. 16, 1992, 17-19)

[Zabr91] Zabrodin, A. V., Myamlin, A. N., Kotov, Ye. I., Pazdnyakov, L. A., *Structure of a Matrix Computer System SMS-14 and a Concept of Developing of a Series of Computers on a Unified Basis*, Kel'dysh Institute of Applied Mathematics, Moscow, 1991.

[Zakh85] Zakharov, K., ""Mars" Supercomputer Being Developed by Interdisciplinary Group," *Sovetskaya Estoniya* 151 (July 2, 1985), 1. (trans.: SNAP, Jul. 17, 1985)

[Zamo85] Zamorin, A. P., Myachev, A. A., Selivanov, Yu. P., *Computers, Computer Systems, Computer Complexes*, Naumov, B. N., Przhiyalkovskiy, V. V., Eds., Energoatomizdat, Moscow, 1985.

[Zhoz85] Khokhikashvili, V. A., Vishnevskiy, V. M., "The Computer Network SIRENA: Design and Analysis," in *Tenth All-Union School-Seminar on Computer Networks: Abstracts of Reports, Part 3* AN SSSR Scientific Council on Cybernetics and AN GSSR IVM, Moscow-Tbilisi, 1985, 377-381.

[Zuyk82] Zuykova, S. A., Marushchak, A. P., Mirin, V. G., Tumanov, A. A., Khvostantsev, N. A., "Operating System for the PAROM Microcomputer Homogeneous Computer System," *Upravlyayushchiye Sistemy i Mashiny* 5 (Sept-Oct, 1982), 76-80. (trans.: JPRS, CCAT/83044, Mar. 10, 1983, 80-87)