JPRS L/9766

1 June 1981

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 14/81)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

APPROVED FOR RELEASE: 2007/02/09: CIA-RDP82-00850R000400020005-0

JPRS L/9766

1 June 1981

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 14/81)

CONTENTS

MICROCOMPUTERS

UDC 681.32-181.48(47+57)

'ELEKTRONIKA S5' MICROCOMPUTERS AND THEIR APPLICATION

Moscow MIKRO-EVM "ELEKTRONIKA S5" I IKH PRIMENENIYE in Russian 1980 (signed to press 3 Nov 80) pp 2-63, 73-84, 154-157

[Annotation, editor's foreword, introduction, chapters 1 and 2, sections 4.1 and 4.2, bibliography and table of contents from book "'Elektronika S5' Microcomputers and Their Application", by Mark Petrovich Gal'perin, Vladimir Yakovlevich Kuznetsov, Yuriy Aleksandrovich Masienikov, Vladimir Yefimovich Pankin, Viktor Panteleymonovich Tsvetov and Aleksandr Ivanovich Borovskoy, Izdatel'stvo "Sovetskoye radio", 35,000 copies, 160 pages]

[Text] The design principle and the hardware and software of a series of domestic microcomputers are discussed. An analysis is made of ways of solving questions arising in the application of microcomputers in instruments, equipment and control systems; practical recommendations are given, as well as examples of the use of microcomputers of the "Elektronika S5" series.

Intended for specialists involved in the development and introduction in various sectors of the national economy of control and data processing and transmission equipment based on computer technology facilities.

Editor's Foreword

A broad range of problems associated with the development, series production and application of microprocessors and microcomputers has evoked the steadily growing interest of engineers, scientists and technical managers at various levels working not only in the area of radio electronics, but also in other areas of the national economy. This interest has been occasioned by considerable success achieved in recent years in our country and abroad in using microprocessors and microcomputers as universal programmed logic devices as part of the structure of measuring instruments, machine tools, industrial and marine automation equipment, in equipment for light industry, the food, oil and gas industries, in medical equipment and in agricultural machinery and units.

The use of microprocessors and microcomputers has made possible the appearance of fundamentally new consumer properties for products; in many fields of engineering their use is a necessary and important condition for competition in the world market. At the same time, the use of microprocessors and microcomputers has made

the following possible: a drastic improvement of operating reliability and the possibility of self-diagnosis of malfunctions, a reduction in power consumption, a reduction in the time required for and the labor intensiveness of developing new kinds of equipment and modernizing it, and a reduction in weight and overall size.

At the present time domestic industry is producing an extensive list of microprocessors, memory and input/output large-scale integrated circuits (LSIC's), as well as completed models of microcomputers. As an example can be named such LSIC series as the following:

The K-580 series containing an eight-bit single-chip microprocessor, a RAM [random access memory] and ROM [read-only memory] LSIC, as well as a parallel and serial interface LSIC, fabricated according to the n-channel MOS [metal oxide semiconductor] technology.

The K-589 series, the basis of which is the so-called "microprocessor sections" executed with TTL [transistor-transistor logic] circuits and Schottky diodes.

The K-587 series constructed on the basis of the complementary MOS technology.

The K-584 series, executed with circuits with injection logic, etc.

Among microcomputers being produced can be named the "Elektronika-NTs," "Elektronika S5" and "Elektronika-60," etc., models. A number of computers are in the concluding stages of development and organization of series production. Software and debugging facilities, which are being improved constantly, have been created for all computer models. Sufficient know-how has been gained in using these computers in various fields of science, engineering and agriculture.

Computers of the "Elektronika S5" series occupy a merited position. The first model of this series--the "Elektronika S5-01" multiboard microcomputer--was created in 1975, and in 1976 began the production of the "Elektronika S5-11" single-board microcomputer. Several more models of microcomputers of this series are being produced at the present time.

Some distinctive features characteristic of the "Elektronika S5" microcomputer series can be mentioned:

The presence in the series' structure of single-board microcomputers and additional modules (memory and input/output), as well as multiboard microcomputers with unified software and debugging facilities, unified circuitry and design and technological solutions, and a unified production and metrological base.

The simultaneous and even advance creation of software, in relation to the development and start of the production of the microcomputers themselves.

Broadly comprehensive work on the creation of models of equipment, instruments and systems based on microcomputers concomitantly with the development of the computers themselves.

The design principles of the most popular models of the "Elektronika S5" series of microcomputers, their software and debugging facilities, as well as questions of procedure and experience gained in creating equipment based on microcomputers for various sectors of the national economy, are described in this book. Furthermore, all questions are discussed by the authors as inseparable parts of the overall problem.

Introduction

In recent years microprocessors and microcomputers have evoked great interest among a broad range of specialists. Conducive to a great extent to satisfaction of this interest have been a great number of publications in scientific and technical periodicals and a number of books by domestic and foreign authors discussing the fundamental principles of the design of foreign microprocessors and microcomputers and of their application. However, the work of specialists has been complicated by the lack of necessary information on domestic models of microcomputers, since journal publications as a rule have not made it possible to form a complete idea of a specific microcomputer or, even more so, of an entire series.

The authors of this book have endeavored to fill in this gap with regard to only one series of microcomputers produced by domestic industry, the "Elektronika S5."

The first chapter of the book is introductory and contains general ideas relating to the application of microcomputers and fundamental concepts without which it is difficult to read the material following it.

The second chapter is devoted to a description of the first generation of the "Elektronika S5" series of microcomputers, designed according to the p-channel MOS technology (p-MOS).

In the third chapter are discussed the key principles of the design of the "Elektronika S5-21" microcomputer, designed on the basis of a single-chip 16-bit microprocessor, according to the n-channel MOS technology (n-MOS). The brevity of the exposition was determined firstly by the succession of many fundamental solutions discussed in ch 2 and, secondly, by the fact that the development of the n-channel generation of the microcomputer series is still continuing.

The fourth chapter is devoted to questions relating to microcomputer software and contains a number of practical recommendations relating to the development of software for specific applications.

In ch 5 know-how gained in the application of microcomputers in various fields of engineering is generalized, a number of standard technical solutions are proposed and a description is given of know-how gained in organizing work relating to the application of microcomputers.

In this book the authors have endeavored to show that the simultaneous creation of microprocessors and microcomputers, their introduction in various instruments ans systems, and the organization of the series production of microcomputers and equipment based on them represent work management principles which make it possible

to solve highly important technical and economic problems and to achieve the enormous national economic saving which potentially lies in the very idea of microprocessor engineering.

The authors will be grateful for all comments which readers of this book deem necessary. It is requested that suggestions and comments be sent to the following address: Moscow, Glavpochtamt, a/ya 693, Izdatel'stvo "Sovetskoye radio".

Chapter 1. Microprocessor Technology and Its Application

1.1. Microprocessors--a Revolution in Engineering

The appearance of microprocessors, memory and input/output LSIC's, as well as microcomputers among electronic engineering products which have been developed and series produced has been evaluated by many experts in the field of radio electronics as a revolutionary phenomenon commensurate in significance with the appearance in the 50's of the first series-produced semiconductor devices.

These declarations were made five to seven years ago, and today the time has begun to determine the first results of work relating to the application of microprocessors and microcomputers and further ways of developing this new trend in engineering, as well as to evaluate the revolutionizing influence of microprocessor technology on various sectors of the national economy.

Among the majority of specialists in the field of classical computer technology the appearance of microprocessors and microcomputers, especially in the first few years, did not evoke notable enthusiasm. In fact, the simplicity of structural solutions, the not-too-developed instruction sets, the short word length and slow speed could be considered a betrayal of the main idea of computer technology--the creation of computers with the maximum possible computing resources. A certain amount of time was necessary to understand that microcomputers represent the most mass-consumption-oriented and most inexpensive class of computers and that their appearance has paved the way for computer technology to enter those areas of human activity for which two barriers--cost and ready availability--earlier seemed to be insurmountable.

In order to evaluate the value of the appearance of microprocessors for specialists, scientific teams and entire branches of industry involved in the development and series production of semiconductor technology products, it is necessary to consider two fundamental features of this new class of electronic engineering products.

The first feature is the universality of the application of a microprocessor and a microcomputer. With an increase in the degree of integration to a few thousand elements on a chip it would have been necessary to develop an integrated circuit (IC) nomenclature exceeding by hundreds of times the nomenclature produced today for circuits with a medium level of integration, in order to support the demand of all branches of industry. A microcomputer constructed with microprocessors and especially a microcomputer on a single chip, is the most universal of previously known electronic logical components, which can perform the functions of a specific piece of equipment depending on the information entered in its programmed memory.

Therefore, the appearance of microprocessors made it possible not only not to increase the IC nomenclature, but even to reduce it as compared with previous generations.

The second feature, closely allied to the first, is the exceptionally high system design properties of this class of electronic components. All previous generations of components designed for use in computers, digital automation equipment and control systems were tied to decisions made for the system as a whole with regard to such parameters as supply voltages and mechanical and climatic influences, since they represented too small a "building block" for constructing these products. A microprocessor and a microcomputer in many instances constitute the basic part of the system directly tied to controlled systems and information sources and carrying in their software an algorithm for operation of the entire system. This feature has forced developers of LSIC's for microprocessors to pose and solve problems relating to selection of the type of interface, to implement the possibility of linking with various information sources and receivers and to create software which is not inferior in complexity to the software of minicomputers. And, finally, in all its magnitude has appeared the problem of setting up a well reasoned procedure for interaction with microcomputer users making possible the required dynamics for the increase in demand for and the production of these products.

A detailed discussion of what a microprocessor and microcomputer is has been presented at the present time in a number of books and periodicals published here at home and abroad [1-5]. Having availed ourselves of these, let us give only some concise definitions.

A microprocessor (MP) is a functionally complete unit with a fixed interface, constructed with LSIC's and consisting of an arithmetic-logic unit (ALU), internal registers and a control unit and designed for implementation of a specified instruction set [1].

By an interface is meant a system of address, information and control lines which is designed for coupling units and for which a time chart has been specified for the flow of information.

A microprocessor set (MPK) is a combination of microprocessor and other IC's compatible with respect to structural design and fabrication technology and designed for use in microprocessors and microcomputers.

A microcomputer (microelectronic computer) is a computer consisting of a microprocessor, a semiconductor memory and means for coupling with peripherals.

1.2. Economic Efficiency of the Use of Microcomputers

A national economic saving from the use of microcomputers is achieved primarily on account of a drastic reduction in the labor intensiveness and cost of products constructed on their basis, improvement of the suitability of these products for quantity production and of their functioning reliability, as well as on account of the endowment of these products with fundamentally new consumer advantages. In addition to the mass consumption properties and low cost characteristic of

microcomputers, as well as the universality and the programmability of their functions, they also possess the property of incorporability. A microcomputer is designed primarily for use directly as part of different kinds of equipment--units, instruments, machine tools, data transmission equipment, test, medical and agricultural equipment, etc. In connection with this, it is feasible to divide all microcomputers into three groups.

Multiboard Microcomputer

A multiboard microcomputer is a computer in which the processor, storage devices and input/output units are implemented as independent boards joined in a single structure together with control and display units and the power supply and designed for use as an incorporable unit or as an independent instrument. It can be said that a multiboard microcomputer represents a readily available and inexpensive minicomputer constructed from microprocessor set LSIC's.

Single-Board Microcomputer

This is a computer in which the processor and storage and input/output units are executed on a single board, as a rule not having its own power supply and control and display units, and designed for use primarily as an incorporable piece of equipment.

Single-Chip Microcomputer

This is an LSIC containing a processor, a working storage, input/output channels and, as a rule, a permanent (semi-permanent) storage. In other words, it is a microcomputer executed as a single LSIC and used only as an incorporable unit.

The incorporation of single-board, single-chip and, in individual instances, multi-board multicomputers, too, in user's equipment is not only not primarily a problem of the structural arrangement of this equipment.

It is primarily the replacement of electronic, electromechanical and mechanical units ("hard logic") with a universal unit (a microcomputer) whose operating routine unambiguously determines the operating algorithm of the entire equipment as a whole. This solution of the problem posed makes it possible to achieve a quite high level of standardization and unification of engineering solutions and a drastic reduction in the labor intensiveness of and of the time required for developing and modernizing equipment based on microcomputers. It can be said that from the management viewpoint the use of microcomputers is one method of resolving the contradiction between the volume of project work and limited labor resources.

It is also the introduction of computing procedures and data processing into equipment, information sources and manipulators, endowing them with new consumer advantages. It is also simplification, reduction of the cost and improvement of the reliability of the hardware and software interface not only of computers, but also of the entire nomenclature of data reception and transmission equipment, measuring instruments, as well as equipment whose combination into a unified system was exceedingly difficult in connection with the lack of computing equipment

accessible in terms of cost and quantity production which could be incorporated into equipment.

The change from designing equipment based on integrated components ("hard logic") to designing even simple units based on microcomputers represents a change to designing a "system in miniature." In addition to improvement of an engineer's labor productivity, this change certainly requires an improvement of professional mastery and involves the appearance of new interesting aspects in everyday design work, which is the goal of each engineer.

A reduction in the cost and labor intensiveness of equipment is achieved not only on account of using LSIC's with a high level of integration and low cost. Single-board and multiboard microcomputers produced under conditions of large-lot production and designed for use in various engineering fields are economically more advantageous than special-purpose electronic units developed for each kind of equipment and produced in small lots. This is especially important for branches of industry which traditionally have not had sufficient knowhow in the development and production of radio electronic components and units but have been forced to create modern models of equipment utilizing electronics. An example of such enterprises is plants of the machine tool building industry and of the machine building industry for light industry and the food industry, agriculture and the like. It is extremely important that the conversion of a specific type of equipment to microcomputers make possible further improvement of the economic parameters of this equipment in proportion to improvement of production technology practically without additional expenditures of time and labor on the part of the manufacturers of the equipment. It can be said that the application of microcomputers represents a method of "transferring" the latest achievements in modern microelectronics characteristic of its technological and organizational approaches into other non-electronic branches of industry.

However, a great national economic savings, which is the basis of the very idea of microprocessors and microcomputers, remains only potentially achievable until equipment based on it passes through all stages of development, mastery in series production and introduction into everyday practice. All these problems are exceedingly complicated and they call for preparation not only of the physical base, but also the training of specialists in a specific field of knowledge, who completely have mastered their basic profession and simultaneously have a profound understanding of the possibilities of microprocessor equipment, their programming and the principles of designing systems based on microcomputers.

If attention is paid to the exceptionally rapid progress in semiconductor technology, which has made it possible to replace generations of microprocessors every two to three years, then it becomes understandable that a situation could be created whereby the cycle of the creation and series-production mastery of a new model of an instrument or other kind of equipment whose parameters are directly dependent on the technical and economic characteristics of the microcomputers included in it proves to be considerably longer than the period for the replacement of generations of technology, and, consequently, the equipment would turn out to be obsolete even before the start of series production. The resolution of this contradiction has proven possible only upon fulfillment of the following conditions.

Condition 1.

This is the comprehensive simultaneous solution of all problems associated, on the one hand, with the creation of the entire microprocessor set of LSIC's and of microcomputers based on them and with the preparation for and organization of their series production, and on the other, with the creation of a broad nomenclature of units, instruments and systems based on microcomputers in various fields of engineering and with the organization of their series production. Fulfillment of this condition is made possible on the basis of longterm interindustrial programs which regard as a unified whole the entire process from the development of microprocessors to the series production of equipment based on them.

Condition 2.

This is the creation at a new technological level of new models of microprocessors and microcomputers which are compatible to the maximum possible extent with computers developed earlier with regard to software, external interface and design, because of which the user's changeover to new models of microcomputers does not cause insurmountable difficulties and makes possible improvement of the characteristics of equipment.

A series of models of microcomputers united by common software, principles of interface organization and designs, we will call a microcomputer series. The concept of a series calls for, simultaneously with the ensurance of succession, also the definite improvement of all technical and economic characteristics when changing to new improved technological and circuitry solutions.

At the present time domestic industry is producing a sufficient number of models and series of microcomputers which vary in terms of characteristics [6] and which differ from one another in speed, word length, memory and interface capacity and principles of organization, utilization characteristics and areas of application. These include the "Elektronika-60" and "Elektronika K1-10" microcomputers and the "Elektronika NTs" and "Elektronika S5" microcomputer series [7-9].

1.3. Composition of the "Elektronika S5" Microcomputer Series

The following are the key components of this microcomputer series: a microprocessor set of LSIC's, single-board microcomputers, a set of microprocessor functional modules (MFM's), single-chip microcomputers, multi-board microcomputers, software and software debugging facilities. A methodology for the application of microcomputers in equipment and systems has been developed for this series.

By microprocessor functional modules of the "Elekronika S5" series is to be understood electronic units constructed with LSIC's of the microprocessor set and designed to broaden the functional capabilities of a single-board microcomputer with respect to memory capacity and composition and number of input/output channels.

Up to the present time the "Elektronika S5" microcomputer series has included three generations: single-board and multiboard microcomputers constructed on the basis of a microprocessor set of LSIC's employing the p-MOS technology; single-board

and multiboard microcomputers and MPM's constructed on the basis of a microprocessor set of LSIC's employing the n-MOS technology; and single-chip microcomputers employing the n-channel technology.

The following chapters of this book are devoted to a description of the first two generations of microcomputers and to experience gained in their application. The third generation represents the logical development of the main line of this series--the endeavor toward maximally small configurations; nevertheless it has a number of specific features as compared with the two previous generations of the series which are such that a description of the structure of single-chip microcomputers and their areas of application is worthy of becoming the subject of a separate book.

Chapter 2. Microcomputers Based on p-Channel MOS LSIC's

2.1. Basic Set of p-Channel MOS LSIC's

In the "Elektronika S5" microcomputers p-MOS LSIC's have the following key characteristics: level of integration--1000 to 2000 elements on a chip for irregular circuits and up to 8000 for regular; area of a chip does not exceed 20 mm^2; circuitry--dynamic, four-cycle; operating frequency of dynamic logic--100 kHz; supply voltage--+24 V and 1.5 V; magnitude of input and output signals--$U_{vkh"1"}$ [input] = = - (11 to 22) V , $U_{vkh"0"}$ = - (0 to 2) V , $U_{vykh"0"}$ [output] = = 0 to 2 V , output transistor open, $U_{vykh"1"}$ = - 22 V, output transistor closed; maximum value of LSIC's load current not greater than 2 mA; LSIC's power requirement not greater than 0.1 W.

MOS LSIC chips are mounted in 24-terminal glassy alloy packages and 48-terminal cermet packages. The external appearance of LSIC's is shown in fig 2.1 [photograph not reproduced].

The composition of a set of p-MOS LSIC's is presented in table 2.1. Several commonly used abbreviations are used in it, as well as below: ALU--arithmetic-logic unit; ROM--read-only memory; MROM--microprogrammed ROM; RAM--random-access memory; I/O unit--input/output unit; ADC--analog-digital converter; MPU--microprogrammed unit.

Table 2.1.

Type of LSIC	Purpose
K536IK9	Eight-bit ALU
K536IK8	Microprogram control
K535RYe2	ROM with 1024 X 8-bit structure
K535RY2	RAM with 1024 X 1-bit structure
K535RU3	RAM with 64 X 8-bit structure
K536IR1	Digital I/O units
K536IK3	I/O exchange control
K536IK4	Serial interface
K536IK5	Timer
K536IK6	ADC control
K536IV1	Keyboard coder

ALU LSIC (K536IK9)

This LSIC makes possible the performance of arithmetic and logic operations and copying operations on eight-bit numbers. A diagram of the ALU is given in fig 2.2.

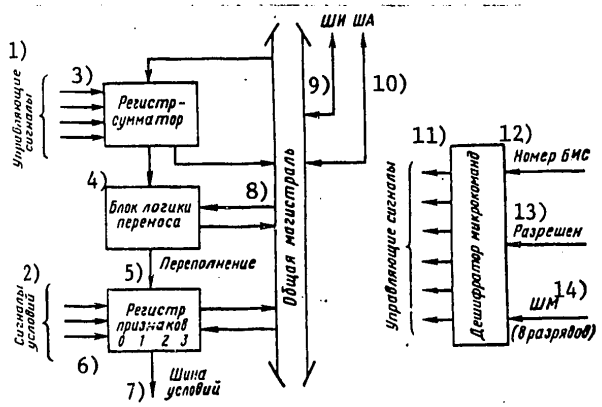


Fig 2.2. Diagram of ALU LSIC

Key:

1. Control signals
2. Condition signals
3. Register-adder
4. Logic carry unit
5. Overflow
6. Control character register
7. Condition line
8. Common line
9. Information line
10. Address line
11. Microinstruction decoder
12. Number of LSIC
13. Permitted
14. Microinstruction line (eight bits)

The operating section of the circuit consists of an eight-bit register-adder, carry logic, a control character register, and an eight-bit information line and address line bidirectionally coupled with the ALU's common line and making possible the exchange of address and numerical information between the microprocessor, memory and input/output units of the microcomputer. In addition to the results of an operation, which are also read out through these channels, the ALU generates four signals through the condition line (carry, overflow, equality to zero and contents of the zero bit of the register-adder).

The operating section is controlled by the microinstruction decoder, to which is supplied the number of the ALU's LSIC, a permit signal and an eight-bit microinstruction word through the microinstruction line (ShM).

MPU LSIC (536IK8)

This LSIC serves the purpose of controlling the operation of the ALU in the microprocessor's system and makes possible the formation of microinstruction addresses depending on conditions entering from the ALU, the microcomputer's console and from outside (cf. fig 2.3).

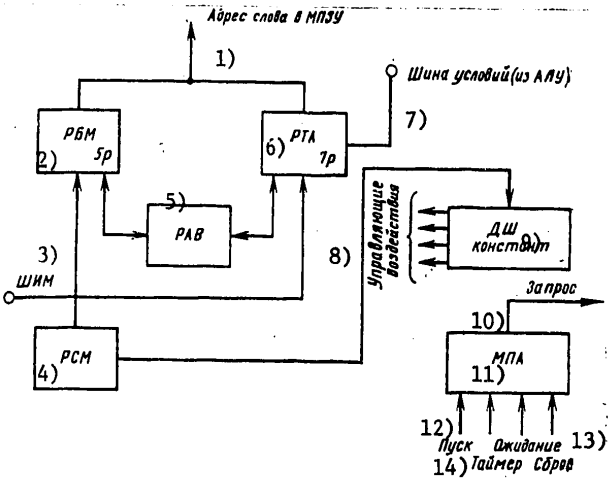


Figure 2.3. Diagram of MPU LSIC

Key:

1. Address of word in MROM
2. RBM [microinstruction base register]
3. ShIM [microinstruction information line]
4. RSM [next microinstruction register]
5. RAV [return address register]
6. RTA [current address register]
7. Condition line (from ALU)
8. Control actions
9. Constant decoder
10. Interrogation
11. MPA [microprogram automaton]
12. Start
13. Wait
14. Reset timer

The MPU LSIC consists of two sections: The first forms a microinstruction address and contains a microinstruction base register (RBM), a current address register (RTA), a next microinstruction register (RSM), a return address register (RAV) and a constant decoder (DSh), and the second contains a microprogram automaton (MPA) which determines the operating modes of the MPU and of the entire MP (start, stop, wait and forming a signal for interrogating the MROM).

The contents of the RBM and RTA are respectively the high-order and low-order bits of the microinstruction's address. Furthermore, six address bits are registered in the RTA in the execution of each microinstruction directly from the MROM, taking into account the information entering through the condition line. Changing the contents of the RBM requires the execution of special microinstructions and is performed with a pre-entry in the RSM. Thus, the address field of the MROM is divided into pages of 128 words each. The microinstruction set has six microinstructions with two of which ("Operation" and "Conditional Transfer") the address changes only within the limits of a page, and with four ("Unconditional Transfer (BP) and Storage," "Unconditional Transfer Without Storage," "Enter Constant" and "Execute Constant") the contents of both the RBM and RTA change. The microinstruction decoder controls all operations. The RAV serves the purpose of storing the return address when going to another page of the MROM. The microinstruction "Operation" is a microinstruction of the MPU with which the ALU performs the semantic processing of information.

All procedures for forming addresses are executed only when the MPA is in the active state and is executing the clocked MROM interrogation problem. The MPA enters this state after executing the "Reset" and "Start" procedures and remains in it until error or stop signals arrive or a signal for anticipation of a response signal from external sources.

Single-Bit RAM LSIC (K535RU2)

This has a capacity of 1024 X 1 bit, an internal regeneration circuit and a working frequency of up to 400 kHz (cf. fig 2.4).

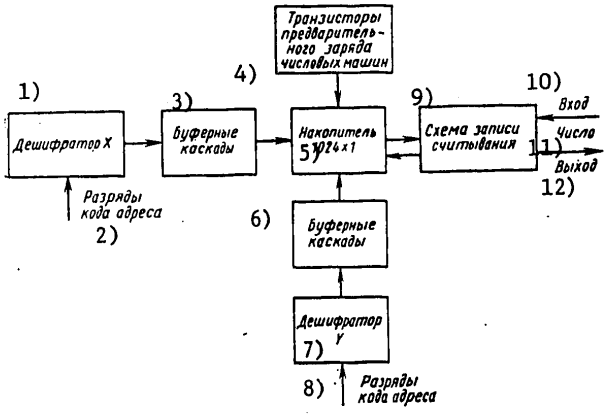


Figure 2.4. Circuit of RAM LSIC

[Key on following page]

Key:

1. X decoder
2. Address code bits
3. Buffer stages
4. Transistors for precharging numeric machines
5. 1024 X 1 storage
6. Buffer stages
7. Y decoder
8. Address code bits
9. Reading entry circuit
10. Input
11. Number
12. Output

Byte RAM LSIC (K535RU3)

This has a capacity of 64 X 8 bits, an internal regeneration circuit and an operating frequency of up to 400 kHz. The circuit is the same, but the storage has a 64 X 8 structure.

ROM LSIC (K535RYe2)

This has an information capacity of 1024 X 8 bits and an operating frequency of up to 400 kHz (cf. fig 2.5).

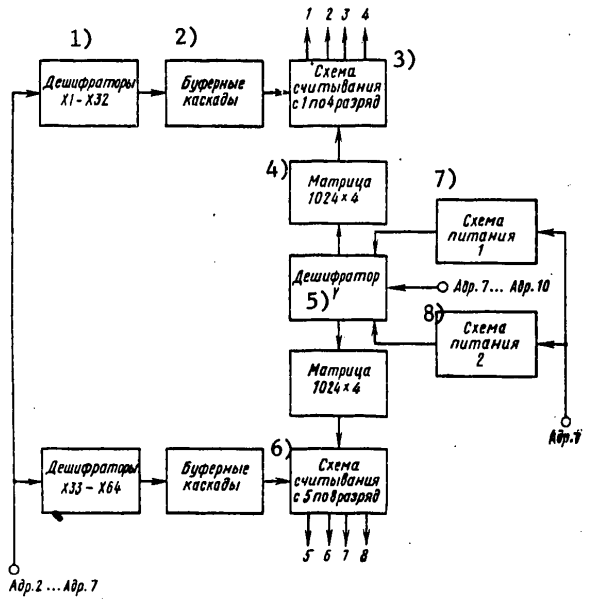


Figure 2.5. ROM LSIC Circuit

Key:

1. X1 to X32 decoders
2. Buffer stages
3. Circuit for reading out bits from 1 to 4
4. 1024 X 4 array

[Continued on following page]

5. Y Decoder
6. Circuit for reading out bits 5 through 8
7. Power circuit 1
8. Address 7 to address 10

Information is registered in the ROM LSIC in the process of fabricating the LSIC. The existence of a transistor at a specific place in the matrix corresponds to a "1." When an ROM LSIC with different information is fabricated, a single mask bearing this information is substituted. The remaining three masks are permanent and do not depend on the information being changed.

Information I/O LSIC's are represented rather fully in the p-channel MPK [microprocessor set] of the "Elektronika S5." Let us briefly discuss their functional design.

Input/Output Central Control (TsUVV) LSIC (K536IK3)

This LSIC contains the following circuits (fig 2.6): a state byte register; processor lines (ShVM's), input/output lines (ShVV's) and a 16-output address decoder; condition lines controlling the operating modes of the decoder; readin and readout control circuits which control the modes of information exchange between ShVM's and ShVV's; and buffer flip-flops for the flags of the LS [logical adder] and LU [logical multiplier], whose purpose will become clear in discussing the next LSIC.

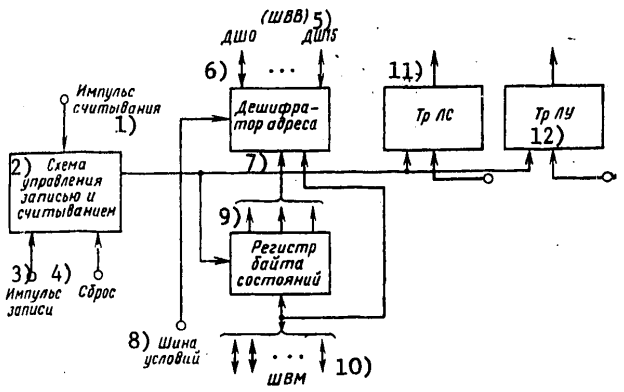


Figure 2.6. Input/Output Central Control LSIC Circuit

Key:
1. Readout pulse
2. Readin and readout control circuit
3. Readin pulse
4. Reset
5. ShVV
6. DSh0 ["0" decoder]
7. Address decoder
8. Condition line
9. State byte register

[Key continued on following page]

10. ShVM
11. LS flip-flop
12. LU flip-flop

The LSIC operates in the following three modes: numerical exchange between the ShVV and ShVM; ordinary decoding, when the address written in the state byte register is decoded; and rapid decoding, when the address in the ShVV is decoded, without storage, bypassing the state byte register. Selection of the decoding mode is controlled by the condition lines.

Digital Input/Output LSIC (K536IR1)

This is designed for the input/output of digital signals or for the reception and processing of interrupt signals. The circuit of the TsVV [digital input/output] LSIC is presented in fig 2.7.

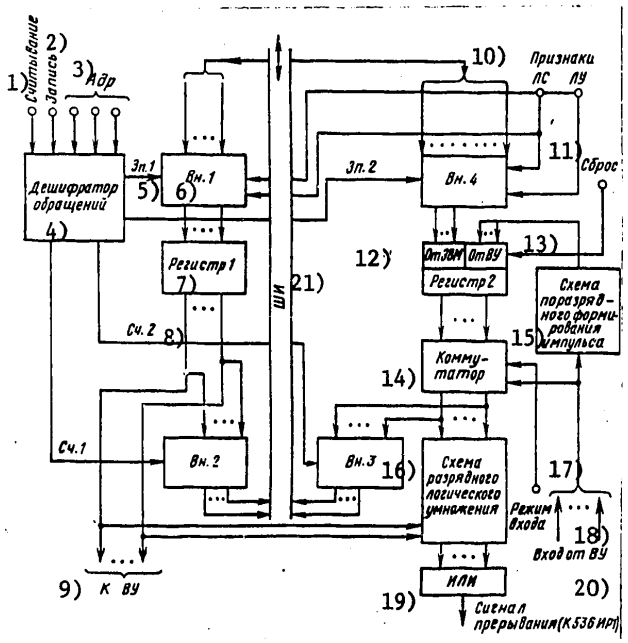


Figure 2.7. Circuit of Digital Input/Output LSIC

Key:
1. Readout
2. Readin
3. Address
4. Address decoder
5. Writein 1
6. Gate 1
7. Register 1
8. Readout 2

[Key continued on following page]

9. To VU [peripherals]
10. LS and LU flags
11. Reset
12. From computer
13. From VU
14. Commutator
15. Digital pulse shaping circuit
16. Digital logical multiplication circuit
17. Input mode
18. Input from VU
19. OR
20. Interrupt signal (K536IR1)
21. Information line

Basic communication between the LSIC and the processor is accomplished through a bidirectional information line. Information is entered either into register 1 through gates Vn. 1 or in register 2 through Vn. 4. Furthermore, depending on the combination of the values of the LS and LU flags (which arrive from the individual flip-flops of the TsUVV LSIC discussed above), is entered either new information, or the logical sum or logical product of the new code value and of that recorded earlier in the registers. This capability is quite convenient when designing various input/output sections. Entry into a specific register, just as the transfer of information to the processor (through gates Vn. 2 or Vn. 3), is determined by signals from the address decoder. Information to peripherals arrives from register 1 without additional gating; consequently, this register is a digital output with storage; the capability of taking back into the computer information previously read out makes it possible to economize on the storage register and, an important consideration, to conduct tests of the section.

Digital information is received in the information line through Vn. 3, depending on the "Input Mode" signal controlling the commutator (from pulse information sources--in register 2 via pulse shaping circuit, and from sources of the level type--directly through the commutator).

Before discussing the operation of the LSIC in the interrupt system, let us recall the basic steps of the interruption procedure regardless of whether they are implemented by means of hardware or software:

1) Recording the reasons for interruption; 2) masking interrupt signals; 3) with the existence of unmasked signals--the generation of a general interrupt signal; 4) analysis of the reason for permitted interruption; 5) selection of the interrupt with the highest priority; 6) development of the interrupt signal; 7) signal reset; 8) return to interrupted routine.

The TsVV LSIC enables the hardware implementation of steps 1, 2, 3 and 7 in the following manner. The recording of interrupt signals is made possible by the digital entry of interrupt signals in register 2, i.e., the inputs from the peripherals (VU) are used as the inputs of the interrupt system. Masking of signals is performed by the digital logical multiplication circuit, whereby the mask code is entered from the processor and is stored in register 1. The general interrupt signal is generated by the OR circuit.

After the generation of a specific interrupt signal it can be reset, for example, by entering in register 2 from the computer a code with "0" in the appropriate location with a logical multiplication flag.

Thus, depending on the connection arrangement, digital input/output LSIC's can be used either as digital channels or as interrupt circuits. Noteworthy also are the capabilities of the logical processing of output information, which are accomplished through the logical addition and logical multiplication flags. A disadvantage of the circuit is the fact that in it is not utilized the capability of redistributing the LSIC's contacts among the two directions of information flow--for reception and for readout.

Serial Interface LSIC (K536IK4)

This makes possible the following: the conversion of a serial code into a parallel eight-bit code and the conversion of a parallel eight-bit code into a serial. Accordingly, this LSIC can operate in the reception mode (PM) with the identification of the codes of the sequence of data received with one eight-bit code assigned in advance (through software), and in the transfer mode (PD). The rate of conversion in the reception-transfer modes is a maximum of 12K bits/s.

The circuit of the serial interface LSIC is presented in fig 2.8 and contains the following: RgBuf--an eight-bit buffer register; RgUst--an eight-bit control point register; RgRezh--a two-bit mode register; SkhSr--a comparison circuit; DSh--an address decoder for "Write" (Zp) and "Read" (Sch) instructions from the microcomputer; Upr--LSIC operation control; and Sch--a three-bit counter.

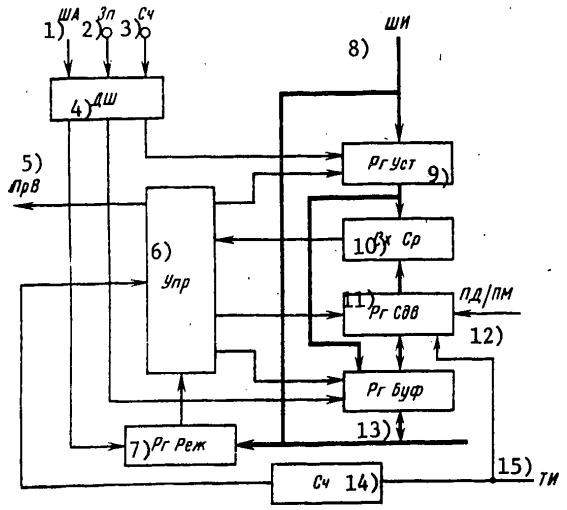


Figure 2.8. Circuit of Serial Interface LSIC

[Key on following page]

Key:

1. Address line
2. Write
3. Read
4. Decoder
5. Interrupt pulse
6. Control
7. Mode register
8. Information line
9. Control point register
10. Comparison circuit
11. Shift register
12. PD/PM
13. Buffer register
14. Counter
15. Clock pulses

The mode for the processing of data in the PM or PD modes is determined by a two-bit code arriving from the processor through the information line, according to table 2.2.

Table 2.2.

Code in mode register	LSIC operations
00	Initial state: All registers are in the reset state and the operation of the LSIC is inhibited.
01	Search in the sequence of PM signals received for a code equal to the code combination stored in the control point register; output of an interrupt signal upon identification of this code
10	All sequences of bytes received (with reference to the overflow signal of the three-bit counter, Sch) are converted into parallel bytes and transferred to the computer.
11	From the sequence of data received, code combinations not equal to the code of the byte stored in the control point register are converted into bytes and transferred to the computer.

In the PM mode the serial code entering the input of the shift register, RgSdv, fills it in relation to shift clock pulses, TI's, moving from the zeroth to the seventh bit. Each state of the shift register is compared with the control point code of the control point register and can be rewritten or not rewritten in the buffer register for subsequent transfer to the processor, depending on the code in the mode register.

In the PD mode the code enters the transfer output serially, beginning with the high-order bit, from the shift register in keeping with the clock pulses; upon completion of the eighth shift cycle, the control circuit through a signal from the output of the counter performs a rewrite from the buffer register into the shift register and generates an interrupt pulse. After this the new transfer cycle begins. If in the interval between interrupt pulses a new byte is not written

into the buffer register, then the code from the control point register will be rewritten in the buffer register.

Timer LSIC (K536IK5)

This LSIC serves the purpose of forming time intervals, delays, and sequences of pulses of various frequencies and has five operating modes: divider, delay, distributor, modulator and interpolator.

In the "Elektronika S5" microcomputer the timer LSIC is used only in the first two modes. The circuit consists of four channels, each of which contains a four-bit counter, a control point register and a comparison circuit. The first two identical channels have in addition two two-input decoders each, and the second two, circuits for reading the current code from the counter. The circuits of the first and third channels of the LSIC are presented in fig 2.9.

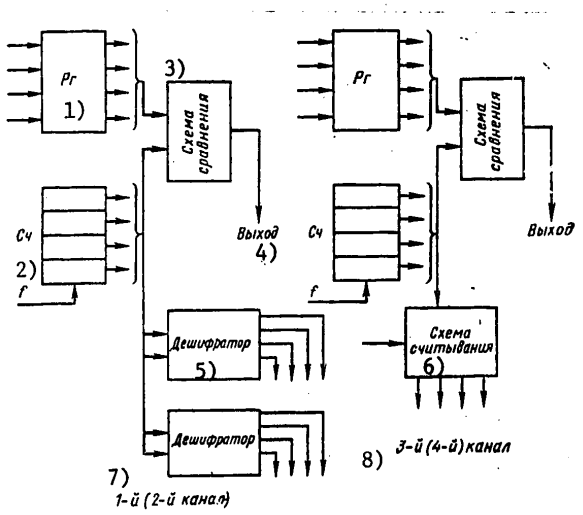


Figure 2.9. Circuits of Two Channels of Timer LSIC

Key:

1. Register
2. Counter
3. Comparison circuit
4. Output
5. Decoder
6. Readout circuit
7. First (second) channel
8. Third (fourth) channel

In the "divider" mode a division factor (from 1 to 16) for the input frequency can be registered in each register through programming. By external switching of the

LSIC's channels it is possible to obtain division factors equal to the product of the factors which are registered in the registers of each of the channels. The maximum value of the input frequency is $f = 20$ kHz .

In the "distributor" mode values of control points are not entered in the registers. The first and second channels operate in the mode of a distributor of pulses in the outputs of each decoder, and the third or fourth in the mode of dividing the input frequency by 10. In either mode in the third and fourth channels the current value of the counter can be read out by programming.

ADC Control LSIC (K536IK6)

This circuit is designed for constructing an analog-digital converter (ADC) which operates according to a two-step integration circuit. The ADC LSIC forms control signals for all ADC units, forms the value of the output code--the result of analog-digital conversion of the input signal--and makes possible interfacing of the ADC with a microcomputer. A circuit of the ADC LSIC and its operating principles are presented in the discussion of the operation of the ADC functional module (cf. "'Elektronika S5-121' Microprocessor Functional Module").

Keyboard Coder LSIC (K536IV1)

This LSIC makes possible the coding (in keeping with GOST [All-Union State Standard] 13052-74) and input of data from a keyboard containing up to 90 keys of the contact-switch type. The circuit of this LSIC and its operating principle are presented in discussing the operation of the system's operator's console.

2.2. Microprocessor

The microprocessor (MP) (fig 2.10) performs arithmetic and logic operations on 16-bit numbers with a fixed point expressed in complement form. The microprogram principle of control is implemented in the microprocessor; a 32-bit microinstruction makes possible the simultaneous control of all units of the MP in the synchronous mode. The time for executing a single microinstruction is 10 μs (one cycle of phased power). The same circuit for designing the MP is used in all p-channel microcomputers of the "Elektronika S5" series.

The 16-bit microprocessor consists of three basic sections: an arithmetic-logic unit, a microprogram control unit and a microprogram storage unit.

The ALU makes possible the storage and processing of information, the forming of addresses of instructions and numbers, and the forming, storage and output of conditions for transfer to the microprogram.

Six eight-bit microcircuits (K536IK9), grouped in twos, form three identical 16-bit ALU's. The first is designed for storing and processing instruction information (ALUK), the second for storing and processing numerical information (ALUCh) and the third--a supplementary (ALUD)--is used when working with bytes, as well as for speeding up the operation of the microprocessor.

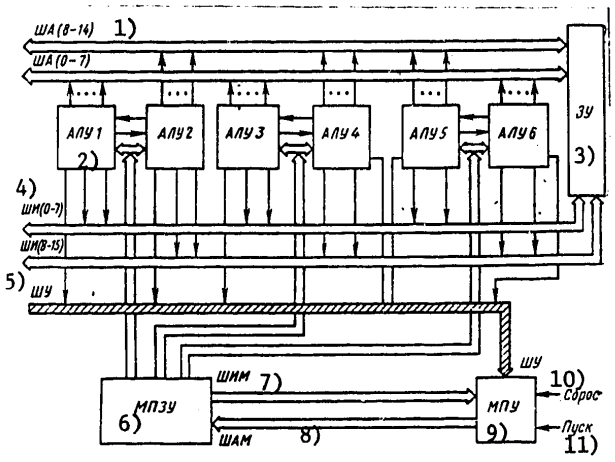


Figure 2.10. Diagram of Microprocessor Employing p-Channel LSIC's

Key:

1. Address line
2. ALU
3. Storage
4. Information line
5. Condition line
6. MROM
7. Microinstruction information line
8. Microinstruction address line
9. MPU
10. Reset
11. Start

The MPU makes possible the formation of addresses and accessing of microinstructions from the MROM, accomplishes branching in keeping with conditions arriving from the ALU and from outside, and enables an asynchronous mode of operation with I/O units.

The microprogram storage unit with a capacity of 1024 32-bit words is designed for storing microprograms of the instruction set, console modes and other functions such as microprograms for controlling peripherals. The ALU, MPU and storage unit (ZU) are interconnected by means of multibit lines: address lines, bidirectional information lines (ShI's), condition lines (ShU's), microinstruction address lines (ShAM's) and microinstruction information lines (ShIM's).

The information line unites the information inputs (outputs) of the ALU, RAM, ROM and of the input/output control unit.

A four-bit condition line serves the purpose of transmitting feedback signals from the ALU and MPU, which together with other control signals determine the sequence for accessing microinstructions from the MROM. The 10-bit ShAM makes possible the

transfer of an address from the MPU to the MROM, and the 32-bit ShIM transfers control information from the MROM to the ALU and MPU. The address and information lines represent the communications line between the microprocessor and memory and input/output unit.

2.3. "Elektronika S5-12" Microcomputer

The "Elektronika S5-12" (fig 2.11) [photograph not reproduced] is a single-board model of the "Elektronika S5" microcomputer series. The main purpose of creating this microcomputer model was the endeavor to achieve minimum cost for a universal programmable computing facility designed for mass application in local monitoring and control systems.

In connection with this, in developing the structure of the "Elektronika S5-12" single-board microcomputer, the following requirements were used as the raw data:*

The microcomputer must be software compatible with other models of the "Elektronika S5" series.

The number of LSIC's and external TTL circuits on the board must be minimal. For this purpose the size of the internal storage unit and input/output units was restricted and the basic set of LSIC's was supplemented with memory circuits with byte (numeric) structural organization and the static method of storing information in the RAM.

The number of interconnections on the board must be minimal on account of the line organization for linking the processor, memory and I/O units through information and address circuits.

The asynchronous principle for operation of the processor with the memory and I/O units was employed in order to speed up and simplify operation with I/O units, to make it possible to connect an auxiliary storage and I/O units with different speed parameters, and to make possible identity of the storage and I/O unit interface.

For the purpose of operating in real time, the possibility of receiving and processing interrupt signals must be provided and timers must be introduced into the computer's structure.

The single-board microcomputer must be used as a unit built into the system, whereby power supplies and controls represent equipment of the system itself.

For the purpose of broadening its range of application, the possibility is provided of increasing the auxiliary storage and I/O units by connecting storage and

*This model was preceded by the "Elektronika S5-11" microcomputer, which has twice as less ROM capacity and the ability to link only with ROM MFM's with electrical transcription of information.

input/output microprocessor functional modules indicated in table 2.4. A description of microprocessor functional modules is presented in sec 2.4.

These requirements, forming the design basis, are reflected in the structural diagram of the "Elektronika S5-12" microcomputer (fig 2.12).

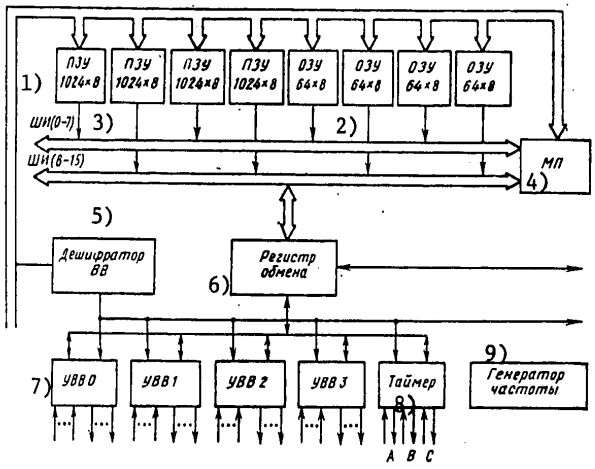


Figure 2.12. Diagram of "Elektronika S5-12" Microcomputer

Key:

1. ROM
2. RAM
3. Information line
4. MP
5. I/O decoder
6. Exchange register
7. I/O units
8. Timer
9. Oscillator

The main functional component of the microcomputer is the 16-bit multichip MP. The microprocessor, memory and the input/output control unit are coupled by a common line which includes address and information lines. Control of the transfer of information through the line and the ensurance of the required load parameters of lines are accomplished by means of information-storing amplifiers (a K535IK5 microcircuit).

The microcomputer's internal storage consists of an RAM with a capacity of 128 16-bit words and an ROM with a capacity of 2048 16-bit words. The RAM with the static method of storing information is executed with four type K535RU2 LSIC's, each of which contains 64 eight-bit registers. The ROM is executed with four type K535RYe2 LSIC's, each of which has a capacity of 1024 eight-bit registers. Recording of information in the ROM is performed in the process of fabricating the LSIC'S. The microcomputer is linked with peripherals through four digital I/O LSIC's

(K536IR1), each of which makes possible the reception and output of eight digital signals. One of the digital I/O LSIC's can be used for receiving interrupt signals and generating a routine-interrupt signal. Digital I/O LSIC's exchange information with peripherals through special buffer circuits whose inputs and outputs are matched with the levels of TTL circuits.

Three type K536IK3 LSIC's form the first input/output address decoding stage (the second I/O address decoding stage is implemented in the digital I/O LSIC's). The structure of the circuit for controlling the I/O unit also includes logic circuits which arrange for the control of the exchange of information between the I/O unit and external channels of the microcomputer, and the generation of an access signal (SOZh), a signal for response from the I/O unit, and a signal for division of the clock frequency. A timer executed with a type K536IK5 LSIC is used for the purpose of forming time intervals for three channels (channels A, B and C).

The microcomputer includes a clock pulse generator (a K535GG1 microcircuit) which forms phased supply voltage pulses for the logic circuits. Voltage pulses for supplying the memory LSIC's are formed in type K592GF1 microcircuits.

The "Elektronika S5-12" is software compatible with other models of "Elektronika S5" series microcomputers. In addition to certain advantages in utilizing software, this property of the microcomputer makes it possible to solve the problem of debugging specific problems by means of a single-board microcomputer of this series--the "Elektronika S5-02." This model has sufficient working storage capacity and information input and documentation facilities, as well as a programmer's console, which makes it possible to fulfill the required conditions for the debugging of programs and for display in the "microcomputer - controlled system" system.

Structurally the "Elektronika S5-12" microcomputer is in the form of a two-sided printed circuit board measuring 260 X 280 X 2 mm, on which on both sides are mounted microcircuits and discrete elements. Type GRPM-61 output connectors are mounted on the two opposite ends of the board measuring 280 mm. It has a metal frame for the purpose of making possible the required rigidity. The assembled board is covered on both sides with covers. On the side walls of the covers there are four threaded holes each, making it possible for the user to attach guides, taking into account the distinctive features of the specific equipment into which the microcomputer is to be built. The key technical data of the "Elektronika S5-12" microcomputer are presented in table 2.3.

Table 2.3.

Word length, bits	16
Operating principle	Parallel
Control principle	Microprogram
Speed	10,000 operations/s
Number of basic instructions	31
Capacity of memory, bits:	
RAM	128 X 16
ROM	2048 X 16

[Continued on following page]

Ability to increase addresses	Up to 32K
Number of single-bit digital input/ output channels	From 24 to 32
Number of interrupt signals	Up to 8
The microcomputer picks up an external signal with a frequency of not greater than 10 kHz and makes possible the following: division of frequencies from 1 to 16 through two channels and from 1 to 256 through one channel	
The microcomputer provides electrical compatibility with microprocessor functional modules	
Parameters of output signals:	
For digital outputs	Voltage of logical 0 not less than +4 V; voltage of logical 1 not greater than +0.4 V
For timer outputs	Voltage of logical 0 not greater than +0.4 V; voltage of logical 1 not less than +4 V
Parameters of input signals	Voltage of logical 0 from 2.4 to 5.0 V; voltage of logical 1 from 0 to +0.4 V
Maximum permissible load current	16 mA
Overall dimensions	284 X 298 X 30 mm
Weight	Not greater than 1.5 kg
External connectors	Type GRPM-61 (four units)
Operating conditions:	
Ambient air temperature, °C	From -10 to +50
Relative humidity of air	To 95 percent at +35 °C
Atmospheric pressure, mm Hg	From 630 to 800
Supply voltage	+ 5 V $\pm$ 5 percent, + 24 V $\pm$ 5 percent, + 1.5 V $\pm$ 10 percent
Power consumption	Not greater than 30 W

2.4. Microprocessor Functional Modules

The set of microprocessor functional modules (MFM's) presented in fig 2.13 [photograph not reproduced] [10.19] [as published] has been developed on the basis of p-MOS LSIC's for the purpose of constructing on the basis of the "Elektronika S5-12" microcomputer systems with broadened functional capabilities. The key characteristics of MFM's are presented in table 2.4.

Table 2.4.

Designation of MFM	Purpose and key characteristics
"Elektronika S5-121"	Fifteen-channel ADC for d.c. input voltages from -10 to +10 V; conversion error not greater than 0.4 percent; conversion time not longer than 0.1 s

[Continued on following page]

"Elektronika S5-122"	Digital input/outputs whose voltage levels match the output levels of TTL circuits: Number of inputs--4 bytes Number of outputs--4 bytes
"Elektronika S5-123"	Interfacing with punched tape I/O units, such as the FS-1501 photoelectric paper tape output unit and the PL-80 or PL-150 punches
"Elektronika S5-124"	Interfacing with type RTA-6 (RTA-7, RTA-60, T-63, STA-2M) teletype
"Elektronika S5-125A"	RAM with a capacity of 2048 16-bit words
"Elektronika S5-125B"	RAM with a capacity of 4096 16-bit words
"Elektronika S5-126"	Display adapter for output to a "Kvant-M" television-type video monitor of information from the microcomputer's storage
System operator's console	Enables interaction of the operator with a system controlled by the microcomputer in data card language
"Kvant-M" VKU [video monitor]	Television-type video monitor based on a 16LK2B CRT [cathode ray tube] and designed for compatible operation with the S5-126 module in displaying information from the microcomputer
"Elektronika P5-PPZU"	Reprogrammable ROM with electrical replacement of information and with an information capacity of 1024 16-bit words

Note: The operating conditions of MFM's are identical to the operating conditions of the "Elektronika S5-12" microcomputer. All modules are executed in the form of single-board constructions (the printed circuit boards have dimensions of 280 X 160 X X 2 mm or 140 X 160 X 2 mm) and have an external frame and protective covers.

"Elektronika S5-121" Functional Module

This makes possible the commutation, analog-digital conversion and entry into the microcomputer of direct-current analog signals. This module has the following key characteristics: number of input voltage channels--16; range of variation of input voltage from -10 to +10 V; conversion error at normal temperature not greater than 0.4 percent; conversion time not longer than 0.1 s; input impedance--100 kΩ.

The module is powered from +24 V $\pm$ 5 percent and -24 V $\pm$ 5 percent d.c. power supplies; the power required by the supplies is not greater than 2 W for -24 V and

not greater than 3 W for +24 V. The overall dimensions of the module are 295 X 165 X 25 mm and it weighs not more than 0.7 kg.

A diagram of the "Elektronika S5-121" module is presented in fig 2.14.

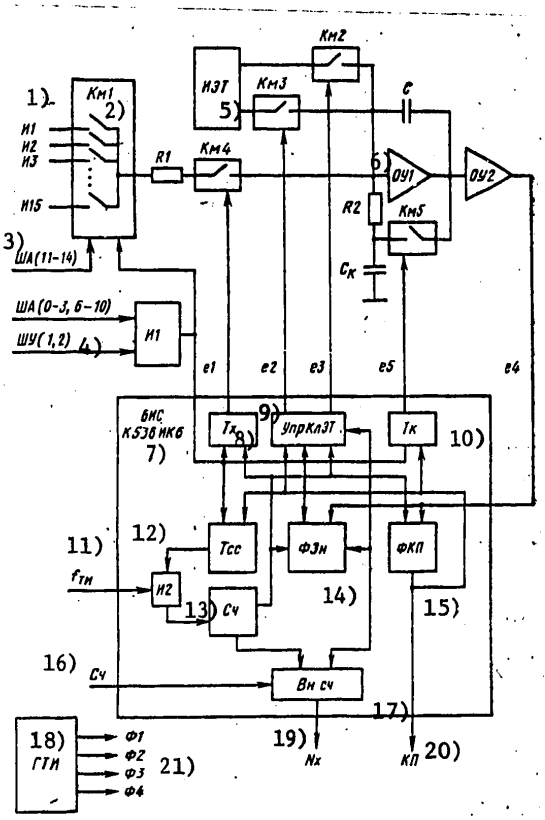


Figure 2.14. Diagram of "Elektronika S5-121" ADC Module

Key:

1. Comparison circuits
2. Commutator
3. Address line
4. Condition line
5. IET [reference current source]
6. OU1 [operational amplifier]
7. K536IK6 LSIC
8. Flip-flop
9. UprKlET [reference current switch control]
10. Commutator flip-flop
11. f_{TI} [clock pulses]
12. Tss [start-stop flip-flop]
13. Sch [counter]
14. FZn [circuit for forming sign bit of output code]

[Key continued on following page]

15. FKP [circuit for forming pulse for termination of conversion]
16. Counter
17. Counter gate
18. GTI [clock pulse generator]
19. N_x [output code sign bit]
20. KP [conversion termination pulse]
21. Shapers

Conversion of the input signal (I1 to I15) is accomplished by the two-step integration method. The key advantage of analog-digital conversion (ADC) of this type is its high noise rejection. Noise from alternating current sources whose period is equal to or a whole number of times less than the duration of the first integration step is suppressed completely, i.e., does not influence the result of analog-digital conversion. The switching of ADC into operation is accomplished with the simultaneous arrival of signals through the following lines: ShA (11 to 14)--a four-bit channel address code; ShA (0 to 3 and 6 to 10)--a nine-bit ADC address code; ShU (1, 2)--two control signals ("Interrogate Storage," "Store/Count").

A signal from the output of a comparison circuit, I1, turns on the input signal commutator (Kml) and the digital control automaton (K536IK6 LSIC). In the LSIC this signal through the start-stop flip-flop, Tss, and comparison circuit I2 enables the entry of pulses, f_{TI} (100 kHz) , in the input of the 12-bit counter (Sch) and through flip-flop Tk picks up signal e5 ("Correction"), turning off commutator Km5. In addition, this signal drives into the initial state circuit FZn--for forming the output code sign bit (in keeping with the polarity of the measured voltage); circuit Upr K1ET--for controlling reference current switches; and (through Tss) circuit FKP--for forming a pulse for the termination of conversion (KP). From the same signal flip-flop Tx sends out signal e1 for the purpose of connecting the channel selected through commutator Km4 to the input of the integrator, which contains an operational amplifier, OU1 , with a capacitor, C , in a negative feedback loop. As a result the first step in integration begins--integration of the input signal from the channel selected. The duration of the first integration step equals the time for the arrival of 2048 pulses from the clock pulse generator (GTI) (a K536GG1 LSIC). A signal from the output of the 11th bit of the counter returns flip-flop Tx to the initial state (at the same time turning off commutator Km4), forms (from signal e4 arriving from the output of null detector OU2) the output code sign bit, N_x , in circuit FZn, prepares circuit FKP and turns on circuit Upr K1ET, which sends out one of signals e2 or e3 (depending on the polarity of the voltage of the channel selected). As a result, in the second integration step to the input of the integrator through commutator Km3 or Km4 from the reference current source (IET) will be supplied a reference current whose polarity is opposite the polarity of the input signal and discharging of the capacitor will take place. At the instant when the capacitor in the second integration step is discharged to zero from signal e4 , circuit FKP forms signal KP, which through flip-flop Tk turns on commutator Km5 for correction of the integrator's zero drift, picks up signal e2 (or e3) through circuit Upr K1ET and through flip-flop Tss closes the input of the counter, Sch. The result of analog-digital conversion--the state of circuit FZn and the number of pulses arriving in Sch in the second integration step--is read with reference to the Sch signal from the outputs of gates Vn sch.

"Elektronika S5-122" Digital Input/Output Functional Module

This module makes possible the following: the reception, storage and transfer of four bytes of level or pulsed digital information in levels of TTL circuits from peripherals to the computer; the output and storage of four bytes of digital information from the computer to peripherals in TTL circuit levels; and the organization of a routine-interrupt system. A diagram of the "Elektronika S5-122" functional module is presented in fig 2.15.

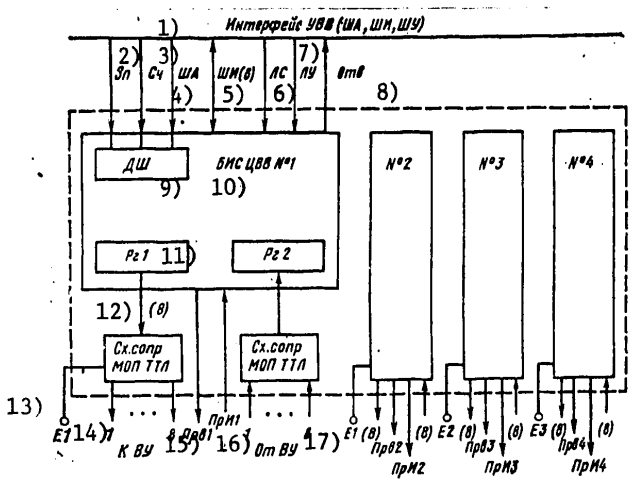


Figure 2.15. Diagram of "Elektronika S5-122" Digital Input/Output Module

Key:

1. Input/output unit interface (address line, information line, condition line)
2. Readin
3. Count
4. Address line
5. Information line
6. Logical adder
7. Logical multiplier
8. Response
9. Decoder
10. Digital I/O LSIC No 1
11. Register 1
12. MOS TTL interface
13. E_1 [voltage]
14. To peripherals
15. Output Prv1
16. Input PrI1
17. From peripherals

The module contains four channels, each of which includes a digital I/O LSIC (K536IR1) and circuits for matching MOS TTL levels for eight digital outputs and for eight digital inputs. Control characters of the LS and LU establish the type of readin from the computer into Rg1 (and Rg2) in keeping with table 2.5. These control characters make it possible to accomplish the selective readin of a "1" and "0" in the desired positions of Rg1 (and Rg2) and accordingly selective switching on and off of peripherals.

Table 2.5.

LS	LU	Type of entry
1	1	Ordinary entry
0	0	"
1	0	Logical summation
0	1	Logical multiplication

The collectors of the output transistors of the matching circuits in the outputs of each channel are connected via 10 kΩ load resistors to the voltage source (E_i not greater than +40 V). The maximum value of the current through the open output transistor (total current from E and peripheral connected to the collector of the output transistor) is 20 mA.

The signal in input PrI_i is established according to the kind of input signal: "1" (0 V) in input PrI_i for making possible the reception of pulsed input signals from the peripheral (with their storage in Rg2 of LSIC $TsVV_i$) and "0" (24 V) in input PrI_i with level input signals from the peripheral (entry is accomplished by bypassing Rg2 of LSIC $TsVV_i$).

When the channel operates in the routine-interrupt signal reception mode, for input PrI_i is established a "0" level, output register Rg1 of LSIC $TsVV_i$ is used for reception from the computer and for storing the interrupt mask, register Rg2 is used for receiving and storing interrupt signals, and through output Prv_i a signal regarding the presence of permitted interrupt signals enters the computer.

Register Rg2 is reset after readout when readout takes place from register Rg2 with the presence of a "1" LU control character. The module is powered from a -24 V $\pm$ 5 percent d.c. power supply, its power requirement is not greater than 4 W and its overall dimensions are 298 X 165 X 25 mm.

"Elektronika S5-123" Functional Module for Interfacing with Punched Tape I/O Units

This module is designed for implementing a hardware-software algorithm for controlling a photoelectric papertape output unit of one of the "Ridmom R40B" and "Ridmom ER40B" types, the FS-1501 when reading information into a microcomputer from punched tape, and for controlling one of the "Perfomom R35," "Perfomom YeR35," PL-150 or PL-80 punches when reading information out of a microcomputer onto punched tape.

The circuit of a functional module for the exchange of information with a tape punch (PL) and a photoelectric papertape output unit (FSU)--of a PL FSU--is presented in fig 2.16 and contains the following: a synchronizer which makes possible switching on and off of the electromagnets (EM's) of the punches upon an instruction from the microcomputer ("Line") and by means of synchronization pulses from the punch; digital input/output channels TsVV1 and TsVV2 (K536IR1 LSIC's), performing the functions of buffer registers for the input/output of digital information in the microcomputer via an I/O unit interface; an interrupt signal input channel--TsVV3 (a K536IR1 LSIC); a timer (a K536IK5 LSIC); amplifiers (U1 to U14);

commutators (K1 to K3); shapers F1 and F2; Coder Sh1; gates Vn1 to Vn8; and logical gathering circuits ILI1 and ILI2 [OR1 and OR2].

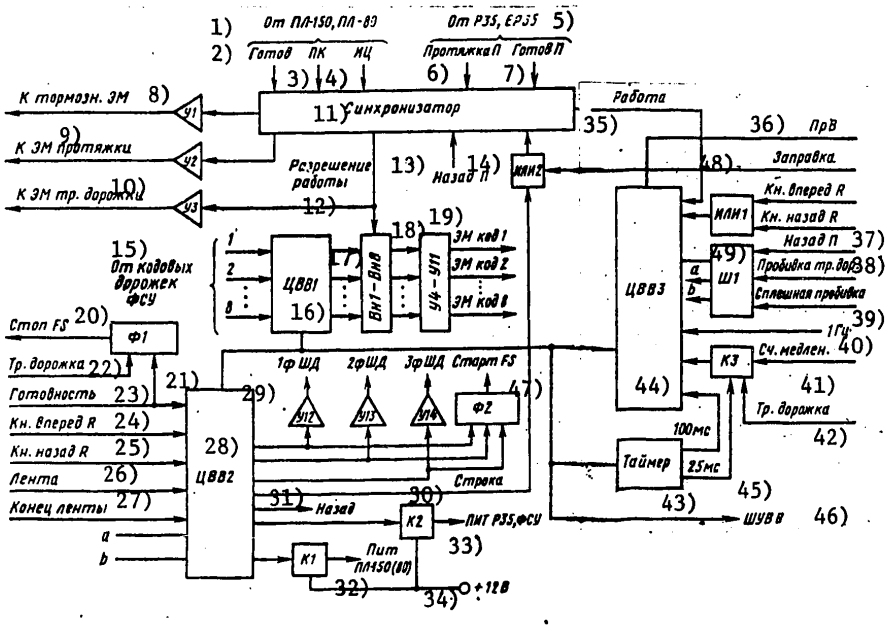


Figure 2.16. Diagram of "Elektronika S5-123" PL FSU Module

Key:

1. From PL-150, PL-80
2. Ready
3. Code reception
4. Beginning of cycle
5. From R35, YeR35
6. Punch advance
7. Punch ready
8. To brake electromagnet
9. To advance electromagnet
10. To transport track EM
11. Synchronizer
12. Permit operation
13. Punch reverse
14. OR2
15. From FSU code tracks
16. TsVV1
17. Vn1 to Vn8
18. U4 to U11
19. Electromagnet codes
20. FS stop
21. F1
22. Transport track
23. Readiness
24. R forward button
25. R reverse button
26. Tape
27. End of tape
28. TsVV2
29. Stepper motor phase 1
30. Line

[Key continued on following page]

31. Reverse
32. PL-150 (-80) power
33. R35, FSU power
34. + 12 V
35. Operation
36. Interrupt input
37. Punch reverse
38. Transport track punching
39. Continuous punching
40. 1 Hz
41. Slow readout
42. Transport track
43. Timer
44. TsVV3
45. 25 ms
46. ShUVV [coding I/O unit]
47. FS start
48. Servicing
49. Shl [coder]

From PL-150 and PL-80 punches synchronization signals enter the module's inputs--PL readiness, code reception (PK) and start of cycle (NTs), as well as state signals--servicing of tape, breaking of tape ("Tape"), and end of tape.

The following signals enter the tape punch from the functional module: turning on of code electromagnets for punching code holes ("EM Code 1" to "EM Code 8"); turning on the electromagnet for punching a transport hole ("EM Transport Track"); and turning on the advance electromagnet ("EM Advance").

The operation of the microcomputer's punch is software-controlled via an I/O unit interface. The tape punch's power is turned on through TsVV2 and commutator K1. For the purpose of turning on the microcomputer's tape punching mode it is sufficient in the absence of "Tape" and "End of Tape" signals to produce a "Line" signal and then with the absence of a "Operation" signal to output new values of the code for punching. With the "Servicing" signal (without the participation of the microcomputer) is accomplished initial servicing of the tape with the punching of only transport holes.

When working with the "Perfomom R35" and "Perfomom YeR35" punches, the following signals enter the module's inputs: synchronization signals--"Punch Ready" and "Advance Punch;" state signals--tape broken and end of tape; and the signals "Punch Track," "Continuous Punching," and "Punch Reverse."

The following signals enter punches from the module: for turning on code electromagnets for punching code holes ("EM Code 1" to "Em Code 8"); for turning on the electromagnet for punching a transport hole ("EM Transport Track") and the advance electromagnet ("EM Advance"); and for turning on the brake electromagnet ("Brake EM").

The punch's operation is software-controlled via an I/O unit interface. The tape punch's power is turned on through TsVV2 and commutator K2, and for turning on the microcomputer's tape punching mode it is sufficient in the absence of "Tape," "End of Tape," and "Reverse" signals to produce a "Line" signal and then with the absence of an "Operation" signal to output new values of the code for punching.

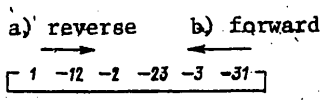
One of the following three signals enters the input of coder Shl: "Reverse Punch," "Punch Transport Track," and "Continuous Punching," which are coded in two-bit code which is transferred to the input of TsVV2; the gathering of these signals acts on input 6 of TsVV3, causing the software implementation of the required

function. Punching is stopped upon removal of voltage from the "Line" signal.

The following signals enter the module's inputs from FSU's of the "Ridmom R40B" and "Ridmom ER40B" type: from the transport track ("Transport Track"); for the presence of a serviced tape ("Readiness"); from FSU code tracks; and from the "R Forward" and "R Reverse" button.

The following signals enter an R40B and ER40B FSU from the module: from TsVV2 through K2 for powering the FSU (+12 V); and from TsVV2 through amplifiers U12, U13 and U14, a voltage of +12 V to the windings of the stepper drive motor.

Advancing of the tape is software-controlled by changing signals in the inputs of U12, U13 and U14 in the following sequence (simultaneous presence of "1" signals in the inputs of stepper motor amplifiers):



With the interlacing of sequences according to arrow a) the tape moves in reverse, and according to arrow b), forward. The "Transport Track" acts on the interrupt input of TsVV3 for the purpose of running the code readout routine and for forming the next step, which is delayed 25 ms (from a signal from the timer) in relation to the preceding.

The following signals enter the module's inputs from the FS-1501: from the transport track ("Transport Track"); the presence of a serviced tape ("Readiness"); and from FSU code tracks.

The following signals enter the FS-1501 from the module: "Start FS" and "Stop FS."

The "Start FS" signal is formed by circuit F2 with any change (increase or decrease) in the total number of "1's" (from one to three) entering its input from TsVV2.

The "Stop FS" signal is formed by circuit F1 in the absence of a "Readiness" signal or with the presence of a "Transport Track" signal (start-stop operating mode). Simultaneously the "Transport Track" signal acts on the interrupt input of TsVV3 for the purpose of running the code readout routine and for forming the next step.

The module is powered from a -24 V $\pm$ 5 percent power supply. The power requirement is not greater than 3 W. Peripherals are powered by the user's equipment. The module's dimensions are 298 X 165 X 25 mm.

"Elektronika S5-124" TA Module--Functional Module for Interfacing with Telegraph Equipment (TA)

This module makes possible the exchange of information between a microcomputer and a five-element-code telegraph set (RTA-6, RTA-60, STA-2M, etc.).

The nominal data rate is 50 bits/s and the code is five-element MTK-2 code; the line voltage is 110 ± 20 V and the electromagnet's current is 40 to 50 mA. A diagram of the TA functional module is presented in fig 2.17.

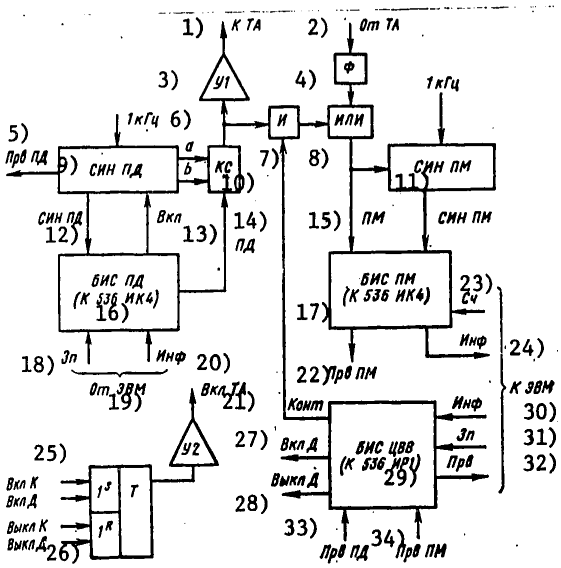


Figure 2.17. Diagram of "Elektronika S5-124" TA Module

Key:

1. To TA
2. From TA
3. U1
4. F [shaper]
5. Data transfer interrupt
6. 1 kHz
7. AND
8. OR
9. Data transfer synchronizer
10. KS
11. Data reception synchronizer
12. Data transfer synchronization
13. On
14. Transfer
15. Reception
16. Data transfer LSIC (K536IK4)
17. Data reception LSIC (K536IK4)
18. Readin
19. From computer
20. Information
21. TA on
22. Data transfer interrupt
23. Readout
24. Information
25. "On" button, motor on
26. "Off" button, motor off
27. Motor on
28. Motor off

[Key continued on following page]

29. TsVV LSIC (K536IR1)
30. Information
31. Readin
32. Interrupt
33. Data transfer interrupt
34. Data reception interrupt

The telegraph set's motor is turned on and off via flip-flop T and amplifier U2 either from buttons ("On Button," "Off Button"), or upon an instruction from the computer via the TsVV LSIC ("Motor On," "Motor Off" signals).

In the mode of the transfer of data (PD) from the computer to the TA, a byte of information from the computer enters the input of the PD LSIC (K536IK4 microcircuit) and at the "Readin" (Zp) signal is entered in the buffer register of the PD LISC, and at the same time the circuit for controlling the PD LSIC is turned on, which makes possible the copying of information from the buffer into the shift register of the PD LSIC and the output of an "On" signal to the input of the data transfer synchronization circuit (SIN PD).

With the "On" signal the distributor (from f_{TI} [clock pulse] = 1 kHz) is turned on, forming Prv PD [data transfer interrupt] signals a and b and SIN PD signals. The Prv PD signal, via the TsVV LSIC operating in the mode of the reception of pulsed interrupt signals, forms in its output a Prv [interrupt] signal, which through a routine-interrupt system enables transfer of the new byte of information.

Signals a and b , acting on the combination circuit (KS), make possible the formation of start and stop bits of the MTK-2 code. The SIN PD signals correspond to the beginning of each MTK-2 code and represent clock pulses for the PD LSIC shift register.

Upon the completion of the formation of all elements of a single MTK-2 pulse (a single character) the circuit for controlling the PD LSIC (from the results of counting SIN PD signals) accomplishes the copying of the new byte (actually a five-digit code) from the PD LSIC buffer register to its shift register and sends an "On" signal to the SIN PD circuit.

In the mode of receiving information from the TA, shaper F converts the signal from the line into a TTL circuit level signal. By means of the start pulse from the TA the data reception synchronizer (SIN PM) is turned on, from whose distributor is formed a SIN PM signal at instants corresponding to the middle of each MTK-2 code pulse. The PM signal enters the input of the PM LSIC's shift register, the clocking shift signal for which is the SIN PM signal. Upon completion of the reception of the last element of the MTK-2 code, it is automatically copied in the PM LSIC from the shift register into the buffer register and a Prv PM [data reception interrupt] signal is sent out, which via the TsVV LSIC acts on the routine-interrupt system, enabling the readout of information from the PM LSIC.

The module is powered from a -24 V $\pm$ 5 percent d.c. power supply and the power requirement is not greater than 3 W. The telegraph set is powered from the user's equipment. The overall dimensions of the module are 298 X 165 X 25 mm.

"Elektronika S5-125" RAM Functional Module

This module has a single board (measuring 140 X 160 mm) with an information capacity of 2048 16-bit words.

"Elektronika S5-125B" RAM Functional Module

This module has two boards of this kind. Each board contains 32 type K535RU2 microcircuits (RAM LSIC's). The information capacity of each microcircuit is 1025 X 1 bit. All LSIC's are divided into two zones having a common circuit--for enabling a specific zone. One zone is designed for entering, storing and reading out information for addresses 0 to 1023, and the other for addresses 1024 to 2047. Accessing of a specific location is accomplished by means of a 10-bit address code. The circuits for entering and reading out information of like bits of the two zones are united.

The board's structure includes a two-stage zone decoder executed with circuits of the KT-902 type. The first stage of the decoder makes it possible to access a specific RAM module from the entire memory field. The second stage is used for accessing one of the two RAM zones.

Four clock amplifiers are used for the purpose of converting and amplifying pulse signals of the master oscillator. Forming the basis of the amplifier's operation is the principle of charging a load capacitor through a low-resistance loop and rapidly discharging it through a high-current switch.

Eighteen readout amplifiers are employed for the purpose of amplifying readout signals and for making possible reliable operation of the microprocessor's input circuits.

The RAM board is powered by -24 V $\pm$ 5 percent, +3 V $\pm$ 5 percent and +5 V $\pm$ 5 percent d.c. power supplies with a total consumption of not more than 3.2 W.

"Elektronika S5-126" Display Adapter (DA) Functional Module

This module in combination with the "Kvant-M" video monitor (VKU) makes it possible to display, on the VKU's screen (a 16LK2B CRT) with a working field of 110 X 90 mm, a specific zone of the storage of a microcomputer with a capacity of 256 bytes (characters) in 14 lines. The height of a character is 4.4 mm and its width is 4 mm. On the VKU's screen are displayed 128 types of characters in keeping with KOI-7 code (GOST 13052-74) without lower-case characters (only capitals): 31 in Cyrillic script, 27 in Roman, 38 numbers and symbols and 32 auxiliary characters.

A diagram of the DA module is presented in fig 2.18, where BU_{zp} is a buffer register, BZU is the buffer RAM, PZU_{zg} is the character generator's ROM, Rg Sdv is a shift register, GTI is the clock pulse generator and BU_{Zpr} ZU is the unit for controlling access to the microcomputer's storage.

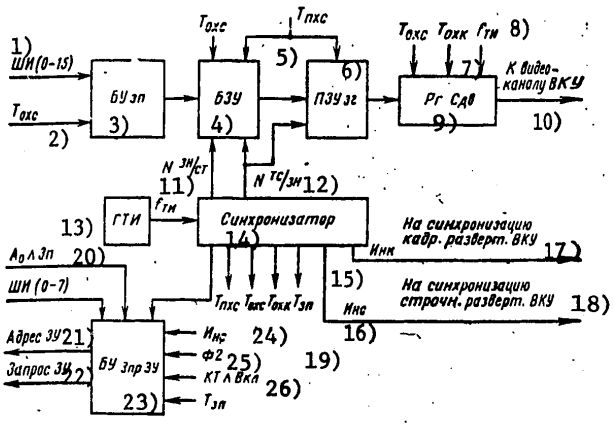


Figure 2.18. Diagram of "Elektronika S5-126" Display Adapter

Key:

1. Information lines
2. T_{okhs} [horizontal scanning fall time]
3. BU_{zp}
4. BZU
5. T_{pkhs} [horizontal scanning rise time]
6. PZU
7. T_{zg} [vertical scanning fall time]
8. f_{TI} [clock pulse]
9. Rg Sdv
10. To VKU video channel
11. N zn/st [number of character in line]
12. N ts/zn [number of television line of character]
13. GTI
14. Synchronizer
15. I_{nk} [vertical scanning pulses]
16. I_{ns} [horizontal scanning pulses]
17. For synchronization of VKU vertical scanning
18. For synchronization of VKU horizontal scanning
19. T_{zp} [readin time]
20. A_0 Λ Zp [readin]
21. Address storage
22. Enable storage
23. BU_{Zpr} ZU
24. I_{ns}
25. F2 [phase 2]
26. KT [end of text] Λ "On"

Time diagrams of the synchronizer's operation are presented in fig 2.19.

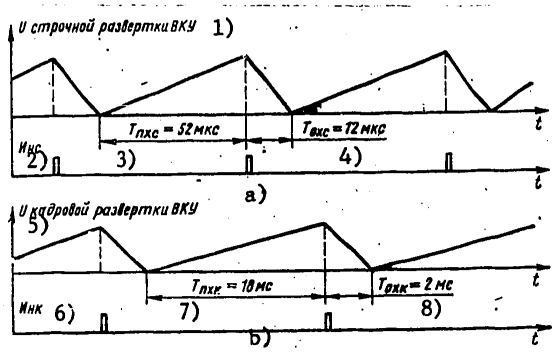


Figure 2.19. Time Diagrams of Synchronizer's Operation

Key:

1. U , VKU horizontal scan
2. I_{ns}
3. T_{pkhs} = 52 μs
4. T_{okhs} = 12 μs
5. U , VKU vertical scan
6. I_{nk}
7. T_{pkhk} = 18 ms [vertical scan rise time]
8. T_{okhk} = 2 ms

The DA module is switched into operation with the arrival from the microcomputer's digital outputs of an eight-bit code for the initial address (A_0) of the zone of the storage to be displayed and of a single-bit Zp [entry] signal. The display adapter makes it possible to form on the VKU's screen characters in the same sequence in which they are arranged in the storage. After readout and display of the last character in the last line, the cycles are automatically repeated with a frequency of 50 Hz. If in the same sequence of bytes (characters) to be read out from the microcomputer's storage a code appears corresponding to the symbol KT [end of text], then further information (to the last character in the last line) is not displayed on the VKU's screen from the 256-character zone indicated in A_0 .

When the signal "Inversion of Brightening" arrives from the microcomputer through the single-bit digital output, bytes read out from the computer's storage and having a "1" in the eighth position are converted into characters in keeping with KOI-7 code with the inversion of brightening.

The display adapter represents a selector channel with direct access to the computer's storage, making possible the conversion of parallel codes read out from

the computer's storage into a serial code for the purpose of modulating the brightness of the beam through the video channel and the output of pulses for synchronizing the horizontal (I_{ns}) and verical (I_{nk}) scanning of the VKU in accordance with the television standard.

Direct access to the computer's storage has been arranged for in the DA by taking into account the fact that access of the processor to the storage takes place always in the second half-cycle of the clock pulse for powering the microcomputer's dynamic logic, whereas the unit for forming interrogations of the storage (the BU_{Zpr} ZU) of the DA module accesses the microcomputer's memory only in the first half-cycle (in phase F2).

For the purpose of matching the slow rate of readout from the microcomputer's storage with the high rate of horizontal scanning, a buffer storage (BZU) has been added, executed on the basis of a K536RU3 LSIC and making it possible to store characters for two character lines of the VKU.

The circuit for synchronizing the operation of the DA contains a quartz oscillator for clock pulses with a frequency of f_{TI} = 2.7 MHz (a GTI) and a counter-divider with decoders ("synchronizer").

The formation of each character on the VKU's screen is accomplished by the method of a character matrix with a 5 X 7 arrangement, whereby two television lines are used for a single character string of the matrix and six television lines are used for forming a space between character lines (a total of 20 television lines for a single character line and the space).

With I_{ns} and F2 pulses and the presence of KT , Vkl ["On"] and T_{zp} [readin] signals the BU_{ZpZU} [as published] unit requests in each television line a byte of information from the computer's storage, whereby the high-order bits of the address code correspond to A_0 , entering from the microcomputer, and the low-order to the current number of the television line (No ts).

This byte of information from the computer's storage is stored in the buffer register of the readin control unit (BU_{Zp}) during the forward movement of the television line and is copied into the BZU with an address corresponding to the current number of the character's television line (N ts/zn) during the line's return movement. The higher bits of the PZU_{zg}'s address code are read out from the BZU according to the N zn/st [number of character in the line] address during the forward horizontal scanning stroke. The low-order bits of the PZU_{zg}'s address code correspond to the number of the character's television line (N ts/zn).

Information read out from the PZU_{zg} (a K536RYe2 LSIC) enters the shift register (Rg Sdv), from whose output with a frequency of f_{TI} (with the absence of T_{okhs} and T_{okhk} signals) is formed a signal for controlling brightening for the VKU's video channel.

The module is powered by -24 V $\pm$ 5 percent, +3 V $\pm$ 5 percent and +5 V $\pm$ 5 percent d.c. power supplies. The total consumption is not greater than 10 W. The overall dimensions of the DA module are 298 X 165 X 25 mm and it weighs no more than 0.7 kg.

"Kvant-M" Type Information Display Video Monitor (VKU)

This module makes possible in combined operation with the DA the output on the screen of a cathode ray tube (CRT) in alphanumeric form the contents of the memory of a microcomputer with a capacity of 256 characters. The "Kvant-M" type VKU is a unit for the passive display of information stored in a storage. A photograph of the "Kvant-M" VKU is presented in fig 2.20 [photograph not reproduced] and a diagram of it in fig 2.21. The VKU contains the following: A type 16LK2B CRT; a horizontal scanning driver (Fsr); a vertical scanning driver (Fkr); a video channel brightness intensifier (UV); and a high-voltage transformer (PV).

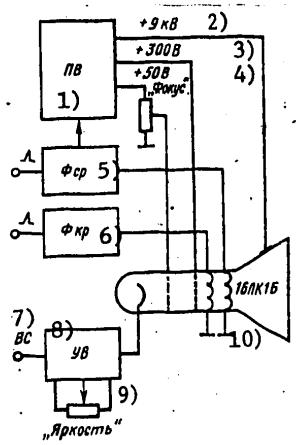


Figure 2.21. Diagram of "Kvant-M" VKU

Key:

1. Transformer
2. +9 kV
3. +300 V
4. "Focus"
5. Horizontal scanning driver
6. Vertical scanning driver
7. VS [video signal]
8. Brightness intensifier
9. "Brightness"
10. 16LK1B [as published]

The method of forming an image on the CRT's screen is the television method. The frequency of horizontal and vertical scanning is respectively 15,625 and 50 Hz.

Since in the reproduction of alphanumeric information on a screen nonlinearity of scanning is especially noticeable to the eye, for the purpose of reducing this distortion drivers Fsr and Fkr have been constructed on the basis of integrated operational amplifiers with feedback.

Since the information to be read out on the screen has only two gradations of brightness, the brightness intensifier has been executed in the form of gating stages. The volume of information read out can be varied over a wide range--from a single character to complete filling of the screen. Therefore in the brightness intensifier section have been eliminated bypass and interstage capacitors whose transient processes of charging and discharging would cause a change in brightness.

The high-voltage source has been executed by means of an electronically stabilized voltage multiplier circuit and is synchronized by means of horizontal scanning pulses for the purpose of reducing noise. Electronic stabilization makes it possible with a change in load current from 0 to 100 μA to change the 9 kV voltage over a wide range. The transformer (PV) also forms a +300 V acceleration voltage and a +50 V voltage for focusing.

The "Kvant-M" VKU is powered by a +12.6 V ± 10 percent d.c. power supply. The load current is not greater than 1.5 A. The overall dimensions of the VKU are 190 X 180 X 250 mm.

The structural design of the "Kvant-M" VKU makes it possible to use it both as a self-contained desktop instrument (in its plastic case) and as an instrument built into the system's console (in this case the plastic case is removed).

"Elektronika P5-PPZU" RROM Functional Module

This module is designed for storage, in the absence of a power supply, of periodically changed information.

Key technical data: information capacity--1024 16-bit words; information access time not greater than 5 μs; length of storage of information with power cut off--2000 h; number of rerecording cycles not less than 10^4 ; power supplies--+5 V ± ± 10 percent, +24 V ± 5 percent and -12 V ± 10 percent; total power requirement not greater than 5 W; overall dimensions 300 X 188 X 30 mm.

It is recommended that the working changing of information in the P5-PPZU module be performed with the "Elektronika P5-ZP PPZU" unit.

System Operator's Console

This unit makes it possible to control the order of executing the system's tasks, to enter new information into the microcomputer and to read out the results of computations to displays and audible signalling systems.

A diagram of the console is presented in fig 2.22 and contains the following: a keyboard unit (BK); a display unit (BI); a dynamic loudspeaker (ZD); a keyboard coder (ShK); a digital output channel (TsVV1); amplifiers (U1 to U9); and a phase voltage generator (F1 to F4). The exchange of information between the console and the microcomputer is carried out through software and hardware via an input/output interface.

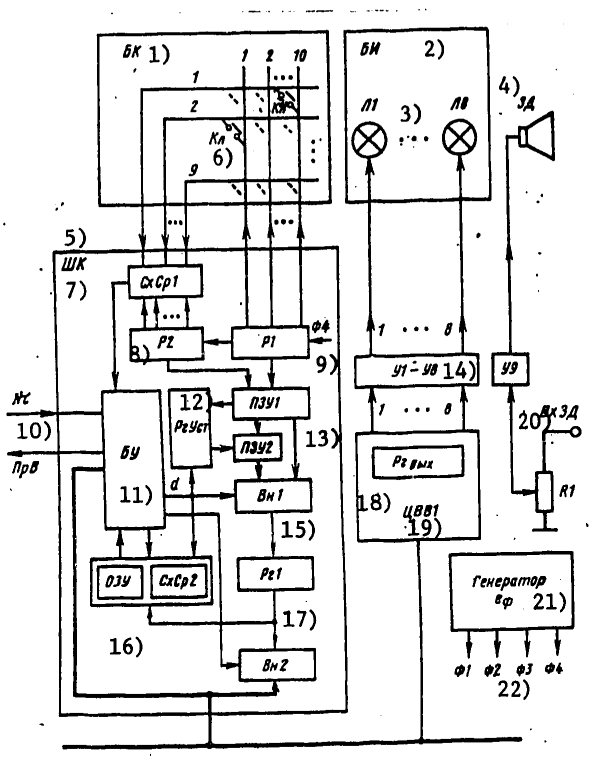


Figure 2.22. Diagram of Operator's Console

Key:

1. Keyboard unit
2. Display unit
3. Lamp 1
4. Dynamic Loudspeaker
5. Keyboard coder
6. Switch
7. Comparison circuit 1
8. Distributor 2
9. F4
10. Interrupt
11. Control
12. Control point register
13. ROM 1
14. U1 to U8
15. Gate 1
16. RAM
17. Register 1
18. Output register
19. TsVV1
20. Dynamic loudspeaker input
21. Phase voltage generator
22. Phases

The keyboard coder (K536IV1 LSIC) makes possible the entry of data from alphanumeric (maximum of 60 keys) and function (maximum of 30 keys) keyboards.

The keyboard unit of the system console contains the following: 4 X 4 alphanumeric keys and 4 X 2 function keys (VR [expansion unknown], NR [expansion unknown], Uprl [Control 1], Upr2 [Control 2], etc.).

The keyboard coder performs the continuous scanning of keys (Kl's) of the keyboard unit with a frequency of 200 Hz. Furthermore, through its 10 outputs distributor Rl sequentially excites the vertical lines of the keyboard matrix (only one vertical line is excited at each instant). When any key is pressed this excitation is transferred (in the appropriate operating cycle of Rl) to a certain horizontal line of the matrix. In each operating cycle of Rl is performed the comparison by means of SkhSrl of a nine-bit code from the horizontal lines of the matrix with the state code of the nine-bit distributor (shift register), R2. During one complete switching cycle of distributor Rl a pulse is formed for carry to the input of distributor R2. The codes for the states of distributors Rl and R2 represent the address code for ROM 1.

When any key is pressed, from the output of SkhSrl is formed a signal through which occurs the formation of a key code in the following order: If one of the function keys (VR, NR, Uprl, Upr2, etc.) was pressed, then the code for this key is copied from ROM 1 into register Rg Ust [control point register]; if a key from the 4 X 4 field is pressed, then its code is read out from ROM 1 and, taking into account the code in Rg Ust, recoding is performed by means of ROM 2 for the purpose of producing in register Rg 1 a code corresponding to the KOI-7 (KOI-8) code of the key pressed.

From a signal from the BU [control unit] is performed a comparison of the code entered in Rg 1 with the codes in the five locations of the RAM; if comparison circuit SkhSr2 then does not produce a signal for equality of the codes, then in unit BU a circuit is switched on for forming a delay for the duration of the "chatter" of the contacts (from 5 to 40 ms), depending on the three-bit control point code, N_T . The working cycle concludes with a repeated check of the repressed key and the formation of suppression signals in Rg 1 and information readiness signals (Prv) with which the microcomputer issues a request for readout and an address. When information is read out from Rg 1, at the same time it is entered in the RAM in the microcomputer.

After removal of the request for readout, suppression for entry into Rg 1 is switched off and the keyboard coder LSIC is ready for the reception and transfer of the next information. If in the first cycle after the removal of the suppression of Rg 1 a pressed key is detected, and if the code for this key is stored in the RAM, then its transfer to the microcomputer will be suppressed; and if this is a new code, then it will be transferred to the microcomputer and be entered in the RAM. As a result, in the RAM will be stored up to five "simultaneously" pressed keys, which makes it possible for the keyboard coder LSIC to enable entry from five "simultaneously" pressed keys.

If during a complete scanning cycle a pressed key is not detected, then the erasure of information takes place in the RAM.

Information enters indicators L1 to L8 from the microcomputer through output register TsVV1 and amplifiers U1 to U8. The input signal enters the dynamic loudspeaker (ZD) through volume control R1 and amplifier U9.

Interfacing the "Elektronika S5-12" Microcomputer with MFM's

Functional modules can be used together with a single-board microcomputer in any combination desired.

The nucleus of these systems is the "Elektronika S5-12" microcomputer, whose interface is in common with modules connected from the outside. The microcomputer forms all signals for accessing these units and does not require additional interfacing circuits when connecting any module. At the same time it is necessary to impose a number of restrictions in order that the capability of an arbitrary arrangement will not result in a number of cases in the creation of technically and economically unfeasible configurations and, in connection with this, in discrediting the idea of the use of single-board microcomputers and MFM's. Let us discuss these restrictions.

The first restriction relates to the capacity of the working storage. The address line system provides the capability of accessing a memory with a maximum capacity of 32K words. However, the construction of large capacities from modules with a capacity of 4K words is unfeasible if only because the chief advantage of the "Elektronika S5" microcomputer series is the implementation of advanced I/O functions in a single-board model. It is recommended that no more than two RAM modules be connected. It is obvious that when it is necessary to create large RAM storage capacities in a specific system another microcomputer model should be used, e.g., the "Elektronika 60" microcomputer.

The use of more than one ROM MFM is also not recommended.

Restrictions on the number of I/O modules are similar. The connection to the microcomputer of more than three parallel interface MFM's, more than one module for communicating with a teletype and more than one ADC module is not recommended. The connection of more than one DA MFM is not permitted. It must also be taken into account that its use is possible only together with a RAM module, since the output to a display of information from the internal storage is not provided for.

Let us discuss some configurations of computing facilities constructed with a single-board microcomputer and MFM's, and the key goals which are achieved by means of a specific configuration.

"Elektronika S5-12" Microcomputer + RAM

This most obvious configuration is intended primarily for those users who totally support the concepts of small configurations and single-board microcomputers, but for whom the capacities of the RAM's of today's microcomputer models are not sufficient for their tasks.

"Elektronika S5-12" Microcomputer + RROM

This configuration can be recommended for a number of cases. Firstly, for the development of experimental models of equipment prior to the fabrication of ROM LSIC's. Secondly, for putting together equipment to be produced in small quantities, for which the fabrication of ROM LSIC's is not justified economically. Thirdly, for the creation of equipment in which the main part of the control or data processing algorithm is unchanged and can be realized with ROM LSIC's installed directly in the microcomputer, and a certain part requires changing only when going from one modification of equipment to another, or in the process of use of each equipment unit.

"Elektronika S5-12" Microcomputer + ADC

This configuration can be used in quite varied fields of engineering. For example, on its basis can be constructed complete measuring instruments or sections for measuring technological parameters within the structure of an ASUTP [automated system for controlling technological processes], whereby the results of measurements can be displayed or documented by one of the methods discussed below. This configuration can also serve as a basic digital servo system in which control of the drive is accomplished by the pulse width modulation method by means of timer circuits included in the structure of the "Elektronika S5-12" microcomputers, or by means of digital-analog converter circuits, e.g., of the K572PA-1 type, which are to be interfaced with the microcomputer and built into user's equipment.

"Elektronika S5-12" Microcomputer + TsVV [Digital I/O Unit]

This configuration quantitatively broadens the capability of a microcomputer's parallel interface for the purpose of connecting information sources and receivers and control units and displays, as well as for the software formation of special interfaces.

"Elektronika S5-12" Microcomputer + RAM + Display + Video Monitor

The purpose of this configuration is the display of the results of computations made by the microcomputer, and more precisely of figures whose display is required by the operation of the equipment into which the microcomputer is built. A display section of this kind can be regarded in two ways. It can be regarded as a "poor display," making it possible to read out only alphanumeric information, and to an extent of less than 300 characters. However, it can be regarded as a unit capable of uniting information ordinarily read out on separate digital displays and capable of accompanying this information with the required text, facilitating its understanding by the operator, warning of the origin of a critical situation, etc. Furthermore, it must not be feared that the filling of even such a small display will be incomplete in a specific application: In the configuration discussed the DA MFM replaces several boards of the parallel interface even with 50-percent utilization of its capabilities with respect to the amount of information displayed.

"Elektronika S5-12" Microcomputer + Operator's Console

This configuration makes possible the manual entry of information into the microcomputer. In combination with the preceding configuration it represents the technical basis for arranging for a dialog between the operator and equipment constructed on the basis of a microcomputer.

"Elektronika S5-12" Microcomputer + Telegraph Set (TA)

The purpose of this configuration is to enable documentation, the working entry of data files from punched tape and individual numbers from a keyboard, and the readout of information onto punched tape.

In addition, this configuration can be a means of interfacing remote microcomputers with one another or with a computer of a higher level when working in a hierarchical system.

With the additional connection of a working storage ("Elektronika S5-12" microcomputer + TA + RAM configuration) a teleprinter version of a supervisory system (TVDS) can be employed.

Let us discuss a configuration which can be used as a facility for debugging programs for users for whom the purchase of a multiboard machine for debugging problems is unacceptable for one reason or another. A minimal configuration can serve as a program debugging facility: a microcomputer, one or two RAM modules and a TA module. The number of RAM modules is determined by the volume of problems to be debugged.

A teleprinter version of a supervisory system (a TVDS) with a capacity of 1024 words, which is located in the microcomputer's ROM, serves the purpose of controlling the operation of this system. The debugging system affords the user with facilities for working interaction with the supervisory system and the microcomputer's specific software. The debugging of user's problems is performed from the TA by entering information of any kind--new routines or numbers--into the microcomputer's working storage, by communicating to the supervisor the parameters of problems and the initiation of problems, and the output of information from the microcomputer's storage to the TA [11].

2.5. Multimachine Systems Based on the "Elektronika S5-12" Microcomputer

The small overall dimensions and low cost of "Elektronika S5-12" microcomputers have been responsible for their use as a multipurpose module to be built into various kinds of terminal equipment, such as regulators, measuring instruments, bench equipment, machine tool equipment, data processing equipment, consoles and the like.

Microcomputers built into terminal equipment contain a set of programs for controlling this equipment (for its various operating modes) and through their input/output channels have an interface both with this equipment's data transmitters and with its actuators. In the construction of systems using terminal equipment of

this type and a central computer which performs supervisory functions with regard to this terminal equipment on the basis of the results of solving a major problem of the system, the problem arises of arranging for the combined operation of several microcomputers.

In organizing the combined operation of microcomputers in a system of this sort the necessity arises of transmitting from the central microcomputer to a peripheral microcomputer located in the terminal equipment messages of the following types: switch on, switch off, transfer a new setting, switch to the execution of the task required, interrogate the state of equipment and the like.

In turn, the peripheral microcomputer must have the ability to transmit to the central microcomputer messages of the following types: the state of equipment, the results of data processing, completion of the execution of a task and the like, i.e., signals transmitted from one microcomputer to another represent (for a system of this type) short messages with a low repetition rate. It is advisable to interchange information with these characteristics through multiplex channels.

In fig 2.23 is presented a structural diagram for the connection of two microcomputers of the "Elektronika S5-12" type, making possible the interchange of information byte by byte, I1 and I2, through registers Rg1 and Rg2 of the TsVV1 channels of both microcomputers. The TsVV3 channel of microcomputer No 1 and the TsVV2 of microcomputer No 2 operate in the mode of the reception of interrupt signals: PD1 is the signal for advising of the transfer of byte I1; OTV2 is the signal for a response regarding the reception of byte I1; PD2 is the signal advising of the transfer of byte I2; and OTV1 is the signal for a response regarding the reception of byte I2.

The microcomputer's operating sequence in transferring a single byte of information from computer No 1 to computer No 2 must be as follows:

I. In computer No 1.
1. Information I1 is entered in TsVV1.
2. PD1 = 1 is entered in TsVV2.
3. PD1 = 0 is entered in TsVV2.
II. In computer No 2.
1. Interrupt signal PrV2 (PD1) is processed.
2. Information I1 is read out from TsVV1.
3. OTV2 = 1 is entered in TsVV3.
4. OTV2 = 0 is entered in TsVV3.
III. In computer No 1.
1. Interrupt signal PrV1 (OTV2) is processed.

In this algorithm the serial entry into the same output register location of first a "1" and then a "0" makes it possible to form a pulsed output signal.

With a similar algorithm it is possible to arrange for the combined operation of a single (central) microcomputer with several peripheral microcomputers, using for this purpose, as illustrated in fig 2.24, in addition a digital input/output module for every three peripheral microcomputers. The exchange of information byte

by byte (I12, I21, I13 and I31) is accomplished through two channels (TsVV1 and TsVV2) of the digital input/output module, and signals PD12, OTV12, PD13 and OTV13 enter the peripheral microcomputers (into their interrupt input/outputs) via channel TsVV3. Signals PD21, OTV21, PD31 and OTV31 enter from the peripheral microcomputers via channel TsVV4, operating in the interrupt signal reception mode.

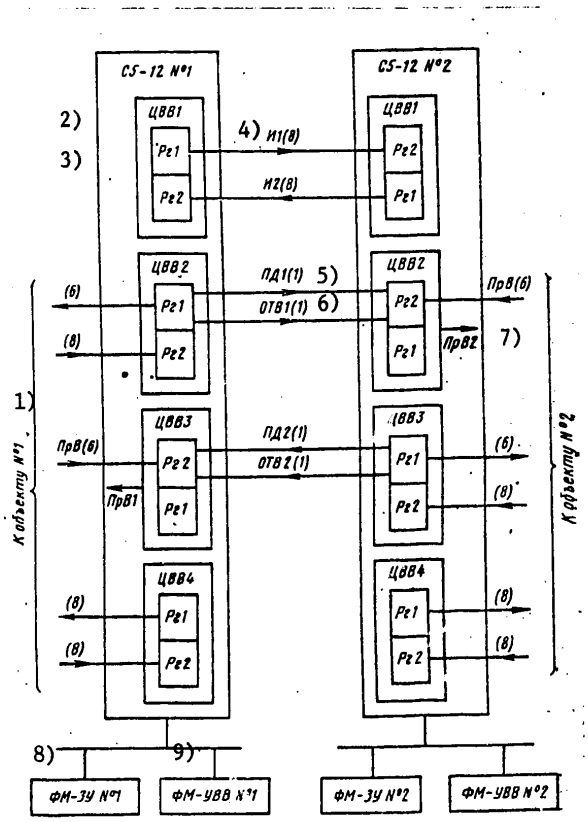


Figure 2.23. Diagram of System of Two Microcomputers Linked via TsVV's [Digital I/O Units]

Key:

1. To system No 1
2. TsVV1
3. Rg1
4. I1
5. PD1
6. OTV1
7. PrV [interrupt]
8. Storage functional module No 1
9. Input/output unit functional module No 1

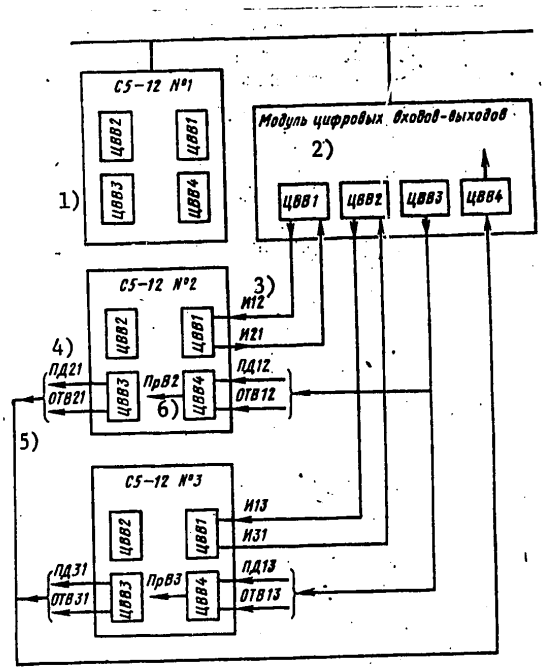


Figure 2.24. Diagram of System of Three Microcomputers Linked via TsVV's

Key:

1. TsVV's
2. Digital input/ouput module
3. I12
4. PD21
5. OTV21
6. PrV2

The permitted interrupt signal enters the interrupt input of the central microcomputer from the output of channel TsVV4 of the digital input/output module.

It must be mentioned that it is possible to connect two more peripheral microcomputers to the central microcomputer if its digital input/output channels are used for this purpose.

The structural diagrams of multimachine systems presented in figs 2.23 and 2.24 are realized on the basis of functional modules, but with an increase in the number of peripheral microcomputers is required a great number of central microcomputer digital input/output channels and communication links. In fig 2.25 is presented the structural diagram of a multimachine system of the trunk line type which minimizes the amount of equipment for arranging for interchange, but which requires the user to connect eight-digit gates (Vn10, Vn01, Vn20, Vn02, Vn30, Vn03, ...) to systems.

The following designations are used in fig 2.25: $TsVV_i$ represents an eight-bit information input/output channel; PrV_i is the interrupt signal of the i-th computer; OI_i represents eight-bit information entering from the common trunk line into the i-th microcomputer; IO_i represents eight-bit information entering from the i-th computer into the common trunk line; PD_{ij} represents a single-bit signal for advising the j-th computer of the need to receive information from i-th computer No 1; and OTV_{ij} represents a single-bit signal for the response of the i-th computer to the j-th.

The operating sequence of the microcomputers when transmitting a byte of information from microcomputer No 2 to microcomputer No 1 must be as follows:

I. In computer No 2.
1. Information I20 is entered into TsVV1.
2. PD21 = 1 is entered into TsVV3.
3. PD21 = 0 is entered into TsVV2.
II. In computer No 1.
1. Interrupt signal PrV1 (PD21) is processed.
2. Gates Vn01 are opened via TsVV3.
3. Signal OTV12 = 1 is entered into TsVV3.
4. Signal OTV12 = 0 is entered into TsVV3.
III. In computer No 2.
1. Interrupt signal PrV2 (OTV12) is processed.
2. Gates Vn20 are opened via TsVV3.
3. Signal OTV21 = 1 is sent out via TsVV3.
4. Signal OTV21 = 0 is sent out via TsVV3.
IV. In computer No 1.
1. Interrupt signal PrV1 (OTV21) is processed.
2. Information IO1 is read out from TsVV1.
3. Gates Vn01 are closed via TsVV3.
4. Signal OTV12 = 1 is sent out via TsVV3.
5. Signal OTV12 = 0 is sent out via TsVV3.
V. In computer No 2.
1. Interrupt signal PrV1 (OTV21) is processed.
2. Gates Vn20 are closed via TsVV3.
3. Signal OTV21 = 1 is sent out via TsVV3.
4. Signal OTV21 = 0 is sent out via TsVV3.
VI. In computer No 3.
1. Interrupt signal PrV1 (OTV21) is processed.

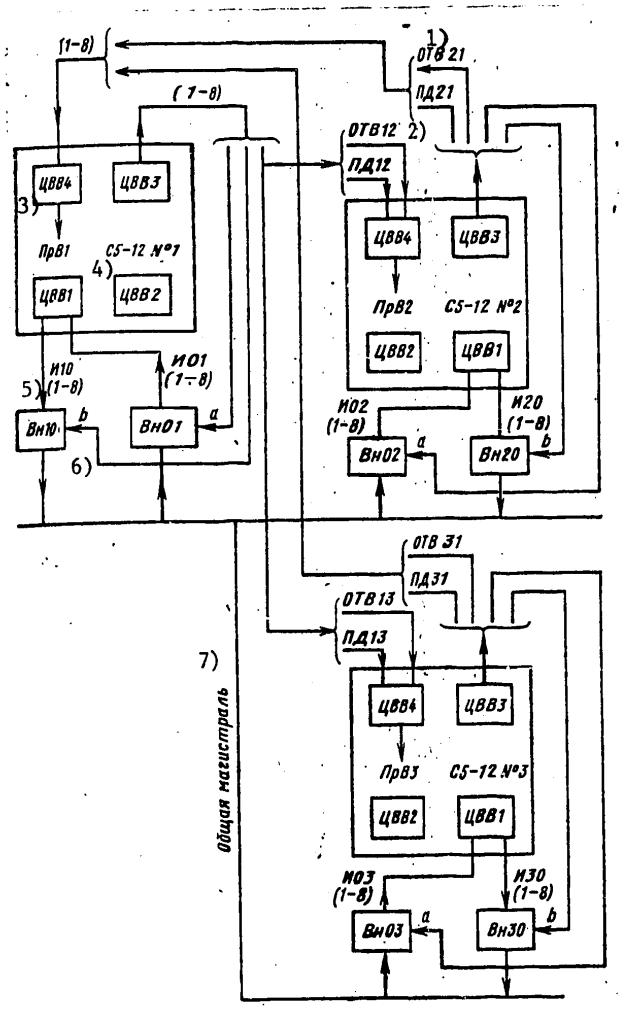


Figure 2.25. Diagram of a Multimachine System Linked via a Parallel Interface

[Key on following page]

Key:
1. OTV21
2. PD21
3. TsVV4
4. PrV1
5. I10
6. Vn10
7. Common trunk line

With great relative distances between microcomputers, for the purpose of lowering the cost of the communication channel it is possible to arrange for the joint operation of microcomputers by including a TA functional module in the structure of each microcomputer. This makes it possible to arrange for exchange by means of a serial code.

A diagram for the connection of two microcomputers of the "Elektronika S5-12" type by means of TA modules is presented in fig 2.26, where PD_{ij} represents a single-wire line for the transmission of data from microcomputer No i to microcomputer No j. With this type of connection, for each microcomputer the other microcomputer is regarded as a teleprinter and the operating algorithm for this system is similar to the operating algorithm for a microcomputer with a teleprinter employing a TA module (whereby exchange can be performed in the duplex mode).

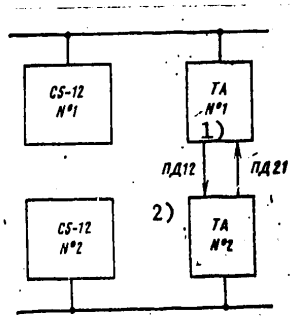


Figure 2.26. Connection Diagram for a System of Two Microcomputers Linked via a Telegraph Set

Key:
1. TA [telegraph set]
2. PD12

The connection of additional computers to one of the computers shown in fig 2.26 can be accomplished through one of the circuits presented in 2.23 to 2.25.

2.6. "Elektronika S5-02" Microcomputer

The "Elektronika S5-02" is a multiboard model of the "Elektronika S5" microcomputer series and represents a 16-bit computer with evolved storage and input/output

systems. This microcomputer's external appearance is shown in fig 2.27 [photograph not reproduced].

The following are the main features of the "Elektronika S5-02" microcomputer: large capacity of its internal storage (up to 20K 16-bit words); an extensive set of equipment for controlling external systems; the presence of a built-in control console designed for starting, stopping and indicating the state of the microcomputer and for executing program debugging routines; the presence of a built-in secondary power supply; and total structural completion (a unified chassis and case).

The "Elektronika S5-02" microcomputer's design includes the following: a microprocessor, a storage, an input/output unit, a control console, a clock pulse generator and a power supply.

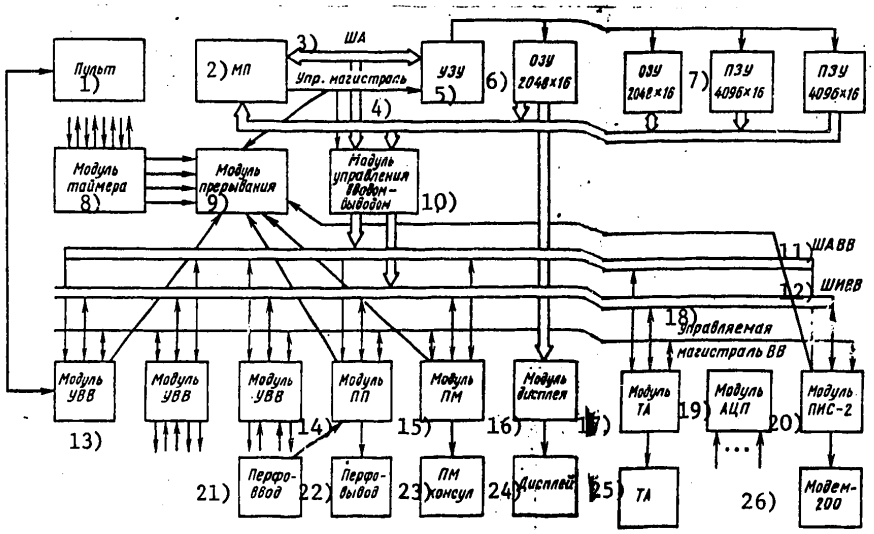


Figure 2.28. Diagram of "Elektronika S5-02" Microcomputer

Key:

1. Console
2. Microprocessor
3. Address line
4. Control line
5. Storage control (UZU)
6. RAM
7. ROM
8. Timer module
9. Interrupt module
10. I/O control module
11. I/O address line
12. I/O information line
13. I/O unit module
14. PP [punch converter] module

[Key continued on following page]

15. PM [loading machine] module
16. Display module
17. TA module
18. Controlled I/O line
19. ADC module
20. PIS-2 module
21. Punch input
22. Punch output
23. "Consul" loading machine
24. Display
25. TA [telegraph set]
26. "Modem-200"

Functionally the 16-bit microprocessor does not differ from processors of all models of the "Elektronika S5" series designed on the basis of a unified p-channel set of LSIC's.

The storage consists of a RAM and a ROM with a total capacity of 20K 16-bit words and a control unit which forms storage control signals (a UZU). The ROM includes individual functional ROM modules, each of which represents a storage with a capacity of 2048 16-bit storage locations. The information storage principle is the static one. A RAM module includes also an address decoder, RAM clock pulse amplifiers and readout amplifiers.

The ROM is made up of individual ROM modules, each of which represents a storage with a capacity of 4096 16-bit storage locations and includes an address decoder, clock pulse amplifiers and readout amplifiers. Blocking of the ROM with an identical RAM and ROM address is provided for. The structure and number of RAM and ROM modules in the microcomputer can be changed depending on the specific application.

The microcomputer's input/output unit makes possible the following: the operation of a set of input/output peripherals, a multilevel interrupt system including program-controlled timer channels, and the operation of a microcomputer control console.

The following are the key functional components of the "Elektronika S5-02" microcomputer input/output unit: a digital input/output unit; a routine-interrupt signal input unit; a punch control unit and an FSU; an analog-digital converter; a TA control unit; a display adapter; a "Consul-260.1" EPM [electrical loading machine] control unit; a unit for interfacing with the "Modem-200" and UPSTG [clock signal conversion unit]; and a program-controlled eight-channel timer; a unit for the reception and output of information to the microcomputer's control console; and an input/output control unit which includes an information exchange circuit and an address decoder.

The first six units have electrical circuits which are identical with the circuits of the corresponding functional modules.

In the "Elektronika S5-02" microcomputer has been implemented the program-controlled hardware method of controlling peripherals. Therefore, the peripheral control units included in the microcomputer's structure represent a set of input and output registers with circuits for matching levels making possible the operation of peripherals, and the algorithm for their operation is implemented through programs (microprograms). The addresses of I/O channels, the RAM and ROM are located in a unified memory field and are coded in specific bits of the 16-bit

address code, i.e., addressing of the I/O unit takes place just as the addressing of the storage does. Therefore, the input/output control unit generates a set of signals which form an input/output interface and which control the exchange of information between peripherals and the microprocessor in the asynchronous mode.

Three groups of lines form the input/output interface: 16 input/output information lines, 36 I/O unit address predecoder lines and eight control lines.

All input/output interface and power lines are led out to the microcomputer's external connectors for the purpose of making it possible to connect additional units for controlling peripherals.

A three-level interrupt system is used in the "Elektronika S5-02" microcomputer. The higher interrupt level represents a digital input of the level type. The higher-level interrupt signal enters the microprocessor. The higher-level inputs are represented by signals entering from the second-level interrupt register, which includes a routine-interrupt register, a computer console interrupt register and an input/output interrupt register. Routine-interrupt signals and signals for interruption from the console enter the digital pulse inputs and the interrupt signal from the input/ouput unit enters the digital level inputs.

The third level of the interrupt system is represented by the routine-interrupt preregister and input/output peripheral interrupt registers. The digital inputs, functioning as interrupt registers, depending on the type of signal entering, can operate either in the pulse or in the level mode. The possibility of further expanding the interrupt system is provided for.

The control console included in the structure of the "Elektronika S5-02" microcomputer is designed for starting and stopping the computer, for entering and reading out storage and input/output unit information and for displaying it, and for executing program debugging routines.

Console operations are implemented at the microprogram level. The control console is connected via digital input/output channels through which upon an interrupt signal from the console the microprocessor shifts to the microprogram implementation of one of the console operations.

The machine cycle power generator is included in the microcomputer's structure as an individual module and makes possible the formation of synchronization pulses for machine cycle powering of the microprocessor, storage and input/output units. The power supply in the microcomputer's structure makes it possible for it to operate from a single-phase 220 V 50 Hz industrial network.

The microcomputer's structure includes a chassis on which are mounted printed circuit boards with connectors of the GRPMI-61 type. The power supply is executed as an independent unit. The control console is in the form of an individual unit fastened to the front wall of the frame and is connected with other units of the microcomputer via a connector. Wiring between boards is accomplished by means of a printed distribution board. The external connectors of the "Elektronika S5-02" microcomputer are of the RP-15-50G type. The microcomputer can be executed in two

variants: desktop and built-in. The key technical characteristics of the microcomputer are presented in table 2.6.

Table 2.6.

Word length	16 bits
Operating principle	Parallel
Control principle	Microprogram
Speed	10,000 operations/s
Number of basic instructions	31
Maximum capacity of internal storage (RAM and ROM)	20K words
Address augmentation capacity	To 32K
The microcomputer makes possible the input and output of information to peripherals as follows for the following models:	
"Elektronika S5-02A"	9 single-byte digital inputs and 7 single-byte digital outputs; 4 two-byte digital inputs and 4 two-byte digital outputs; a PL-80 and PL-150 punch; an FSU of the FS-1501 type and a teleprinter, video monitor and ADC
"Elektronika S5-20B"	One single-byte digital input and one single-byte digital output; four two-byte digital inputs and four two-byte digital outputs; a PL-80 or PL-150 punch, an FSU of the FS-1501 type, and a teleprinter
Interrupt system	Three-level
The microcomputer generates signals	Through four 12-bit channels and four four-bit channels of a program-controlled timer
The microcomputer has a quartz oscillator by which are formed the following frequencies	100, 10 and 1 kHz and 100, 10 and 1 Hz
Levels of input and output signals for communication with peripherals	Correspond to standard levels of TTL circuits
Power required from 220 V 50 Hz industrial network	Not greater than 100 W
Overall dimensions	460 X 412 X 243 mm
Weight of microcomputer	Not greater than 23 kg
Operating conditions:	
Ambient temperature, °C	From -10 to +50
Relative humidity of air	To 95 percent at +35 °C
Atmospheric pressure, mm Hg	From 630 to 800

Chapter 4. Microcomputer Software

The quality of modern computers, including microcomputers, and the possibilities of using them for controlling, processing and transmitting data (UOPD) are determined to a great extent by their degree of equipment with facilities for automation of the development of programs, with operating systems, with libraries of standard programs, and with facilities for the program check and diagnosis of the state of the computing system. The reason for this is the steadily growing, and having become essential, amount of software design in the process of creating information controlling systems.

For such a mass-produced element base of equipment for controlling, processing and transmitting data as microcomputers are, the general software should be united from an organizational and technical viewpoint with the computer-aided design of LSIC's, since ROM LSIC's are the basic medium for specific routines in information controlling systems based on microcomputers.

A description of the various elements of the software (PO) of the "Elektronika S5" microcomputers, as well as certain questions relating to its design, represents the content of this chapter. Here the major emphasis is placed on what standard software gives the microcomputer user and what typical problems must be solved by the user in the process of developing specific routines.

4.1. Instruction Set

The structure of the "Elektronika S5" microcomputer series meets the requirements of universality for performing the most common tasks in the creation of different kinds of equipment, instruments and systems for controlling, processing and transmitting data. The presence in the instruction set's structure of the majority of arithmetic-logic operations, the evolved addressing system and the ability to work with words and files makes it possible to develop programs which are effective in terms of productivity criteria and the storage capacity occupied and which implement various algorithms for industrial automation and the processing and transmission of data and algorithms for controllers and terminals.

The presence in the instruction set of special operations making it possible to control the process of the transmission of routine and peripheral interrupts makes it possible to arrange for a multiprogramming mode and the performance of specific tasks in real time. All this, taking into account the available hardware, makes it possible to define microcomputers of the "Elektronika S5" series as broad-application multipurpose control computers.

The instruction set represents a microcomputer of the "Elektronika S5" series as a 16-bit computer with an address field with a maximum of 32K 16-bit words [13]. Included in this address field are input/output addresses with which interaction is carried out by the same operations as interaction with the storage. This solution, which basically does not complicate the structure, substantially extends the ability to work with peripherals.

The computer's structure includes the following (fig 4.1): a 16-bit ALU [arithmetic-logic unit] with a single-bit communication register (RS) and a two-bit result flag (P), a general address field with a capacity of 2^{16} bytes, an RgK [instruction register], and a four-bit problem number register (RNZ) which serves the purpose of indicating the location of general registers in the storage. Sixteen locations are set aside in the storage for each problem in correspondence with the RNZ and these are used as program-accessible general registers (OR's). Of these sixteen OR's the first three (the zeroth, first and second) are used for the microprocessor's operation and are engaged respectively by the instruction counter (SchK), the interrupted task register (RPZ), and the protection register. The remaining 13 OR's (the third to F) are considered program registers. The basic unit of information in the computer is an eight-bit byte.

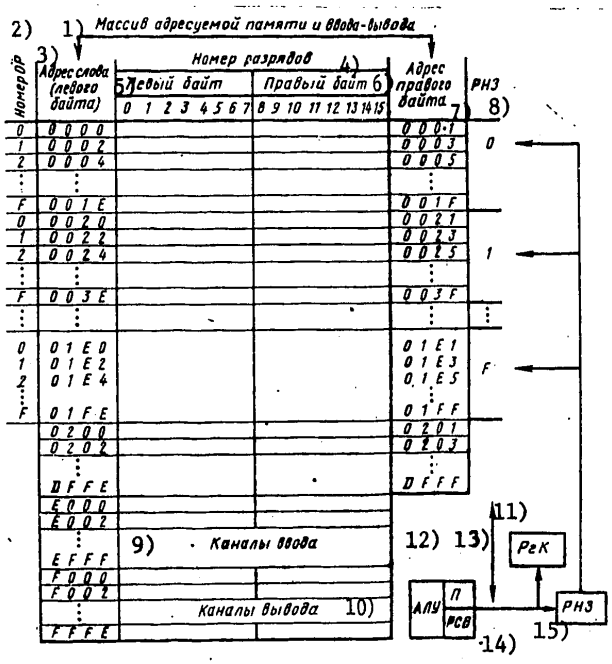


Figure 4.1. Structural Diagram of Microcomputer from the Programmer's Viewpoint

Key:

1. Addressable storage and I/O file
2. Number of general register
3. Address of word (left byte)
4. Number of bit
5. Left byte
6. Right byte
7. Address of right byte

[Key continued on following page]

8. Problem number register
9. Input channels
10. Output channels
11. Instruction register
12. ALU
13. Result flag
14. Communication register
15. Problem number register

The instruction set makes possible the execution of operations with two kinds of data formats: a byte and a word. A word consists of two bytes--a left and right. The numbering of bits in a word goes from left to right, beginning with zero. The zeroth bit of the number in the storage is used as the sign (fig 4.2a). Numbers in the computer's storage are represented in complement, and in computations, in modified complement code with a fixed point after the sign (zeroth) bit. The range of the representation of numbers in the computer is $-1 \leq x < 1$. Taking into account the representation of numbers in the computer in complement code, the maximum positive number ($1-2^{-15}$) is written as 7FFF, the maximum negative number (-2^{-15}) as FFFF, and the minimum positive number (0) is written as 0000 and the minimum negative (-1) as 8000.

Three instruction formats (fig 4.2b) are used in the instruction set. All formats have a half-byte field size.

Instructions of format I include the operations of addition, subtraction, nondestructive comparison, logical addition and multiplication, modulo-2 addition, search for a right-hand "1," accessing from the storage to general registers, loading the contents of the general registers into the storage, logical shifts to the right and left, jump to a subroutine and storage of the return address, and cycle organization. Mention should be made of the distinctive feature of employing the "accessing" and "loading" operations, the first of which makes it possible to directly assign a constant, and the second an unconditional jump to any address of the storage field.

Instructions of format II are used for making a jump with reference to the state of a flag and with reference to information specified in the "mask." Ones in individual bits indicate the condition for a jump in accordance with the appropriate flag. A forward jump takes place by no more than 128_{10} addresses with reference to the instruction counter, and a reverse jump by no more than 127_{10} addresses. When it is necessary to transfer control by a greater number of instructions it is necessary to employ in addition an appropriate modification of the "loading" operation.

Typical of format III are special instructions. They include working with the communication register and general registers, attendance to a new problem and return to an interrupted problem, accessing a problem number, substitution of a higher-level mask, working with the interrupt system and working with flags. Also belonging to this format are the operations of loading the OR [general register] file into the RAM and of copying the RAM's file into the OR, shifts with the direct indication of the number of shifts--arithmetic to the right and logical to the left and right, as well as jumps to a microprogram level and a halt. It should be mentioned that also included in the basic set of instructions supplied in all modifications of microcomputers are the multiplication and division of 16-bit numbers without rounding off (fig 4.3).

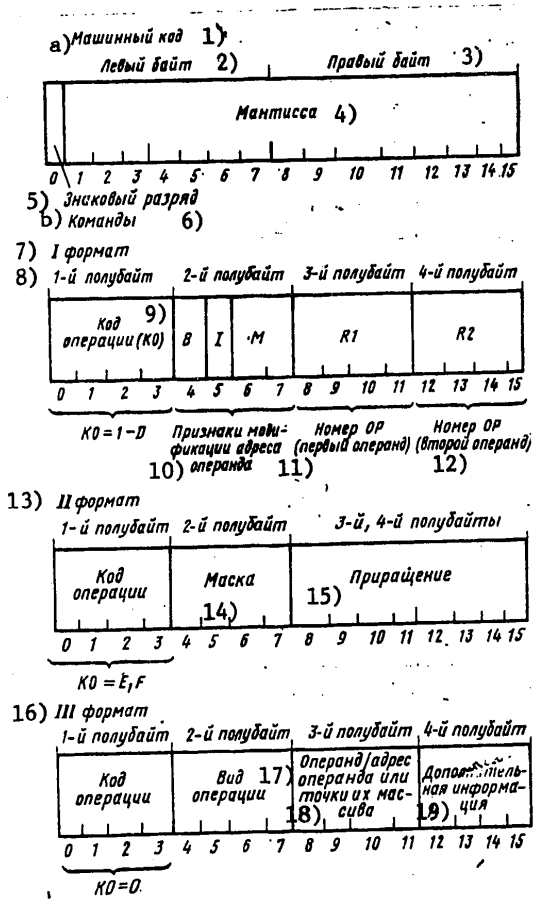


Figure 4.2. Representation of Numbers and Instructions in Microcomputers

Key:

1. Machine code
2. Left byte
3. Right byte
4. Mantissa
5. Sign bit
6. Instructions
7. Format I
8. First half-byte
9. Operation code
10. Operand address modification flag
11. Number of general register (first operand)
12. Second operand
13. Format II

[Key continued on following page]

14. Mask
15. Increment
16. Format III
17. Kind of operation
18. Operand/operand address or points of their file
19. Additional information

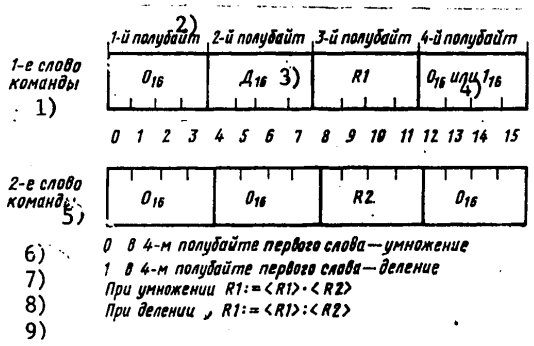


Figure 4.3. Operations of Multiplication and Division in Microcomputers

Key:
1. First word of instruction
2. First half-byte
3. D_{16}
4. 0_{16} or 1_{16}
5. Second word of instruction
6. "0" in fourth half-byte of first word--multiplication
7. "1" in fourth half-byte of first word--division
8. When multiplying
9. When dividing

Operation with input/output registers, which represent with regard to their addressing part of the general memory field, is performed in format I.

In the instruction set for the "Elektronika S5" series there is the concept of the instruction set core. This core includes the basic instructions of format I and II and certain instructions of format III. The introduction of this concept and its practical hardware implementation have made it possible in the "Elektronika S5-21" microcomputer to distribute optimally the computing requirements of specific programs between the storage and the speed, depending on the nature of specific algorithms and the logical-time requirements for programs implementing them.

4.2. Cross Facilities for the Automation of Programming

One of the main goals in the development of standard software for microcomputers is reducing the time required for designing and introducing instruments and systems based on them. An effective means of achieving this goal is the development of facilities for the automation of programming and the debugging of microcomputer programs employing domestic multipurpose computers. These facilities have become

known as cross facilities, and the multipurpose computer designed for this purpose, as a technological computer. For the "Elektronika S5" microcomputers both the automation of programming and work relating to debugging programs are accomplished by means of these facilities. Therefore, they can be called facilities for automating the development of programs (SAPR's). A distinctive feature of the creation of SAPR's for "Elektronika S5" microcomputers has also been the fact that they began to be developed as soon as a determination was made of the logical structure of these microcomputers and one of the major components of these facilities--the instruction set simulator (MPSK)--had been created and begun to be used before the appearance of the microcomputer itself, which made possible the parallel development of the software, the microcomputer and a number of systems based on it [7, 14, 15].

The capabilities of cross SAPR's are quire considerable. They include the utilization of the great resources of the input/output storage of technological computers, parallelism in the development of programs and the hardware portion of a system based on microcomputers and, finally, the ability to simulate external conditions by means of programs implemented on a technological computer and able to be joined to the MPSK. Therefore, now the delivery of microcomputers by major firms is unthought of without the appropriate software [16].

Cross facilities for the automation of the development of programs for "Elektronika S5" series microcomputers have been supplied for the BESM-6, YeS and M-220 multipurpose computers (fig 4.4). A structural diagram of the utilization of facilities which have been developed is presented in fig 4.5. The programming languages used in implementing cross facilities are indicated in table 4.1. The key components of SAPR's supplied for all technological computers include a translator routine with an assembler, a loading routine and a debugging routine, an MPSK and service routines.

Table 4.1.

Cross SAPR's	BESM-6	YeS	M-220
Translator routine from BASIC language	-	PL-1	-
Translator routine from assembler language	FORTRAN	"	Autocode
Loading routine and debugging routine	"	FORTRAN	"
MPSK	"	"	"
Documentation preparation system	"	-	"

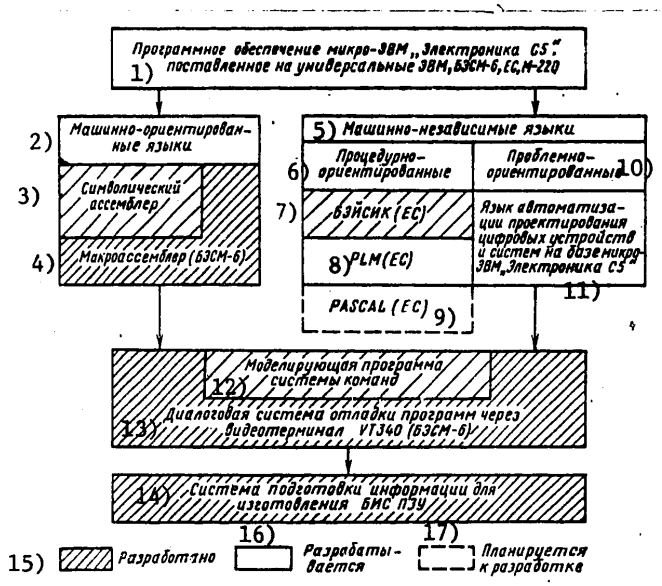


Figure 4.4. Cross SAPR's for Microcomputers

Key:

1. Software for "Elektronika S5" microcomputers supplied for BESM-6, YeS and M-220 multipurpose computers
2. Machine-oriented languages
3. Symbolic assembler
4. Macroassembler (BESM-6)
5. Machine-independent languages
6. Procedure-oriented
7. BASIC (YeS)
8. PLM (YeS)
9. PASCAL (YeS)
10. Problem-oriented
11. Language for automating the design of digital units and systems based on the "Elektronika S5" microcomputers
12. Instruction set simulator
13. Dialog system for debugging programs via a VT340 video terminal (BESM-6)
14. System for preparing information for the fabrication of ROM LSIC's
15. Developed
16. In development
17. Planned for development

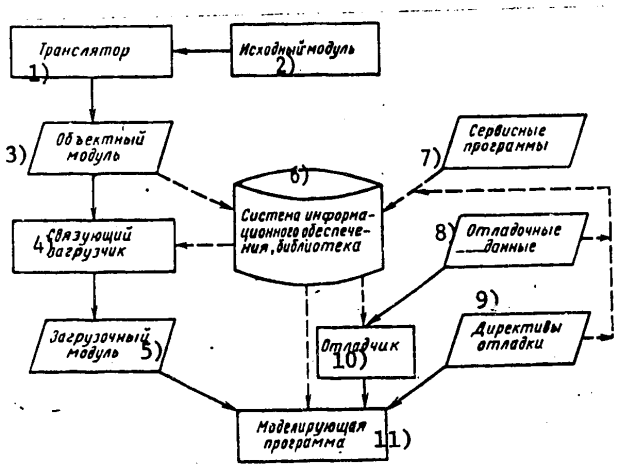


Figure 4.5. Interaction Structure of Cross SAPR's

Key:

1. Translator routine
2. Source module
3. Object module
4. Connecting loading routine
5. Loading module
6. Information support system, library
7. Service routines
8. Debugging data
9. Debugging instructions
10. Debugging routine
11. Simulator

The microcomputer's assembler, preserving the ability of utilizing all features of the instruction set, gives the programmer a good means of implementing an algorithm: mnemonic coding of operations and flags, symbolic addressing, automatic allocation of the storage, the linking of subroutines, and the diagnosis of syntactical errors in translation of the source text. All microcomputer instructions have equivalents in the assembler language. In order to simplify the entry of some frequently encountered instructions, special mnemonic codes have been provided which define not only the operation, but also the value of one of the operands. In the language are provided pseudo-instructions which define the names of routines and their termination, lists of input and external names, types of instruction and data storage, storage redundancy and the like. The assembler language makes it possible to use address constants, hexadecimal and binary sign constants of arbitrary length, integral constants and constants with a fixed point with a length of one byte or of a word. The results of the translator routine's work are printed out in the form of a text arranged in format No 11 with a margin in keeping with YeSKD [Unified System of Design Documentation] requirements. Presented in the text are tables of storage ranges and of input and external names, the source text and object module, and messages regarding errors of 23 kinds.

The uniting of individually translated routines (modules) and their adjustment for specific addresses is performed by the loading routine. The result of its work is also printed out. The loading routine can be performed in the same package with the translator routine or in a separate package. The machine routine produced after the loading routine is prepared completely for simulation or for entry into a system for preparing documentation for the fabrication of ROM LSIC's.

The MPSK, based on the data entered via the debugging routine, interprets the instructions of a specific microcomputer routine, executing in keeping with the debugging instructions the required kinds of printout: operation by operation, printout of the contents of files when executing a specific instruction, etc. (fig 4.6).

SAPR's supplied for BESM-6 and YeS computers make active use of the set of service routines servicing the SAPR library: creation of the library, printout of a table of contents, compression of the library, catalogueing, and removal, printout and punching of a module [17-19]. The use of the key components of SAPR's has demonstrated their effectiveness and convenience for programmers and developers of specific routines for instruments and systems based on the "Elektronika S5" series of microcomputers.

A natural development of cross facilities has been enhancement of the level of the input language: the development of translator routines for a macroassembler and the problem-oriented languages PL-1, BASIC, etc. This work is partly completed--the use of translator routines for the macroassembler for BESM-6 computers and of the BASIC language for YeS computers has begun and development of a translator routine from the PLM language for YeS computers has been completed. Next comes the extension of these facilities to remaining multipurpose computers and the creation of translator routines for other modern high-level languages.

Meanwhile it is necessary to recall the restrictions imposed by cross SAPR's: the relative labor intensiveness of implementing a program model of external conditions, the package mode of running, and check variants which are long in terms of time. While the first limitation is a natural disadvantage of simulation, the second has been reduced substantially by the introduction of a dialogue mode of debugging by means of terminal equipment of the "Videoton-240" type [20]. This mode, implemented in BESM-6 computers, is made possible by a dialogue debugging system (SIDOP), which has made it possible to carry out debugging procedures in an interaction mode with the time sharing of resources between terminals. Included among these procedures are the translation of microcomputer routines from autocode, their unification and loading into a simulated storage, execution by means of an MPSK, the formation (editing) of the required text files and servicing (fig 4.7).

The SIDOP includes the following: a monitor, accomplishing centralized control of the execution of procedures in the time sharing mode; a dialogue translator routine making possible a syntactical and semantic check and the translation into an internal representation of the system of requests (responses) of SIDOP users; a file combination of routines implementing direct access to the BESM-6's external storage; a service combination of routines making dialogue editing possible. In addition, the SIDOP includes modified SAPR-MPSK units, a translator routine and a loading routine.

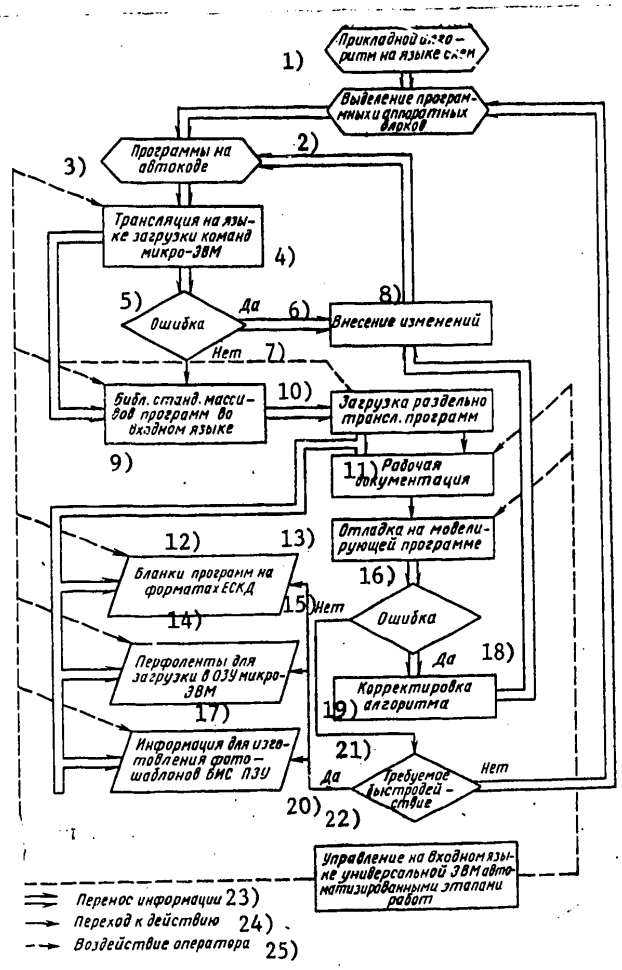


Figure 4.6. Development of Specific Routines for Microcomputers, Employing a Multipurpose Computer

Key:

1. Applied algorithm in hardware language
2. Separation of software and hardware units
3. Routines in autocode
4. Translation in loading routine language of microcomputer instructions
5. Error

[Key continued on following page]

6. Yes
7. No
8. Introduction of changes
9. Library of standard files of routines in input language
10. Loading of translator routines individually
11. Working documentation
12. Forms for routines in YeSKD format
13. Debugging in simulator
14. Punched tape for loading into the microcomputer's RAM
15. No
16. Error
17. Information for making masks for ROM LSIC's
18. Yes
19. Correction of algorithm
20. Yes
21. Required speed
22. Control of automated working steps in input language of multipurpose computer
23. Information transfer
24. Jump to operation
25. Action of operator

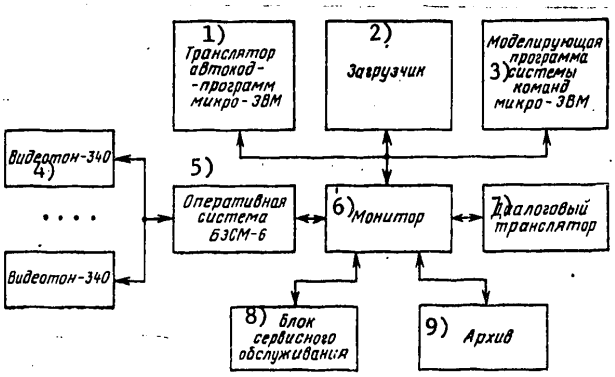


Figure 4.7. Structural Diagram of System for Dialogue Debugging of Microcomputer Routines

Key:

1. Autocode - microcomputer routine translator routine
2. Loading routine
3. Microcomputer instruction set simulator
4. Videton-340
5. BESM-6 operating system
6. Monitor
7. Dialogue translator routine
8. Servicing unit
9. File

The SIDOP's work is organized on the basis of the "Dubna" operating system, and exchange with terminals is performed by means of the "Multitype" subsystem.

The user's interaction language enables all capabilities of the interaction operating mode, including the dynamic control of the processes of translating, loading and executing microcomputer routines. Also present are the required procedures for interaction with the program to be debugged, used by a programmer in working at the console of a real microcomputer. The interaction language used realizes both classical modes of dialogue between the user and computer--at the user's initiative and at the computer's initiative--which provides maximum convenience in interacting with the system and is the most suitable for debugging programs in the interpretation mode. At each step in the dialogue in the SIDOP a syntactical and semantic check is made, along with a diagnosis of users' requests (responses) in keeping with the vocabulary of the interaction language.

The results of using the dialogue system of debugging have produced important quantitative and qualitative results. The time required for the comparable debugging of microcomputer programs at the stage of working with BESM-6 cross SAPR's has been reduced three- to fivefold on average. This takes place even under the condition that the use of the dialogue mode of interaction with SAPR's is especially effective only at the stages of editing a translated text of routines and with the operation-by-operation passage or running of short (in terms of duration of the solution) sections of routines piece by piece, since the execution rate for routines in the simulation mode is not greater than 200 to 1000 instructions per second, depending on the kind of MPSK implementation and the speed of the technological computer. The "degree" of the generation of routines at this stage has also increased considerably on account of a drastic reduction in the amount of work passing through computer operators and on account of an improvement in the comfort of debugging conditions associated with the presence in the SIDOP of all operating facilities for access to routines and data present in a real microcomputer and of facilities for editing, storing and rapid request and on account of the employment of the considerable service set of routines available in cross SAPR's and the SIDOP.

The next stage, representing a qualtitatively new approach to the design of instruments and systems based on microcomputers, is the changeover from automating programming to automating the design of instruments and systems whose basic component is a microcomputer. For this purpose the macroprocedural problem-oriented ZENIT language has been created and put into experimental use, along with a corresponding translator routine for several classes of microcomputers. The evolution of this trend will make it possible in part to level out the contradiction between the high demand for programmers and developers of specific routines for instruments and systems based on microcomputers and the extensive introduction of the microcomputer in product development practice [21].

Bibliography

1. Alekseyevskiy, M.A. and Yevzovich, I.S. "Development of Microprocessors and Microprocessor Systems. Part I," OBZORY PO ELEKTRONNOY TEKHNIKE. SERIYA 3. MIKROELEKTRONIKA, No 5, 1975, 63 pages.

2. Val'kov, V.M. and Ilyushenko, Yu.M. "Tsifrovyye integral'nyye skhemy, mikroprotsessory i mikro-EVM" [Digital Integrated Circuits, Microprocessors and Microcomputers], edited by V.M. Proleyko, Moscow, Sovetskoye Radio, 1977, 104 pages.

3. Prangishvili, I.V. "Mikroprotsessory i mikro-EVM" [Microprocessors and Microcomputers], Moscow, Energiya, 1979, 232 pages.

4. Khilburi, Dzh. and Dzhulich, P. "Mikro-EVM i mikroprotsessory: Tekhnicheskiye sredstva, programmnoye obespecheniye, primeneniye" [Microcomputers and Microprocessors: Hardware, Software, Application], translated from English, edited by S.D. Pashkeyev, Moscow, Mir, 1979, 263 pages.

5. Makglin, D. "Mikroprotsessory: Tekhnologiya, arkhitektura i primeneniye" [Microprocessors: Technology, Architecture and Application], translated from English by M.A. Kasper, edited by I.V. Prangishvili, Moscow, Energiya, 1979, 224 pages.

6. Proleyko, V.M. "Microprocessors, Microcomputers and Their Development," ELEKTRONNAYA PROMYSHLENNOST', Nos 11-12, 1979, pp 3-6.

7. Gal'perin, M.P., Zhukov, Ye.I., Kuznetsov, V.Ya., Malinovskiy, B.N., Maslenikov, Yu.A., Pankin, V.Ye. and Palagin, A.V. "Series of Multipurpose Microcomputers," UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, 1976, pp 27-29.

8. Borovskoy, A.I., Kuznetsov, V.Ya. and Tsvetov, V.P. "'Elektronika S5' Multipurpose Control Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 30-32.

9. Kuznetsov, V.Ya., Maslenikov, Yu.A., Nikitin, E.A. and Tsvetov, V.P. "Development of 'Elektronika S5' Microcomputers and Systems Based on Them," ELEKTRONNAYA PROMYSHLNENNOST', Nos 11-12, 1979, pp 9-12.

10. Gal'perin, M.P., Zhukov, Ye.I., Pankin, V.Ye. and Selivanov, V.I. "Principles of Organization of the Interface of 'Elektronika S5' Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 33-34.

11. Bessrebrennikov, V.V., Gal'perin, M.P., Gorodetskiy, V.V. and Gilyarovskaya, Ye.B. "Set of Functional Modules for 'Elektronika S5' Single-Board Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', Nos 11-12, 1979, pp 43-44.

12. Gal'perin, M.P., Zhukov, Ye.I., Pankin, V.Ye. and Selivanov, V.I. "Principles of the Design of Microelectronic Interfacing Units for 'Elektronika S5' Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', Nos 11-12, 1979, pp 87-88.

13. Istomin, G.S., Kuznetsov, V.Ya., Maslenikov, Yu.A. and Melamed, O.N. "Investigation of the Operations Structure and Loading of Microcomputers for the Implementation of Subscriber Station Algorithms," TEKHNIKA SREDSTV SVYAZI, SERIYA TPS, No 2, 1977, pp 94-98.

14. Istomin, G.S., Kuznetsov, V.Ya., Maslenikov, Yu.A. and Pankin, V.Ye. "Use of Microcomputers for Solving Functional Problems of Data Transmission Network Subscriber Stations," TEKHNIKA SREDSTV SVYAZI, SER. TPS, No 6, 1976, pp 45-51.

15. Gal'perin, M.P., Maslenikov, Yu.A., Nikitin, E.A. and Shebarshin, A.V. "Principles of the Creation of Software for the 'Elektronika S5' Series of Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 35-36.

16. Vinogradov, V.V., Khavkin, V.Ye. and Shebarshin, A.V. "Development of Microprocessors and Microprocessor Systems. Part II" in "Obzory po elektronnoy tekhnike. Ser. 3. Mikroelekronika" [Reviews of Electronic Equipment, Series 3, Microelectronics], No 3, 1976, 56 pages.

17. Ivanov, Yu.V., Kolosovskiy, L.I., Chebykin, N.Ye. and Shturts, I.V. "Cross Facilities for the Development and Debugging of Programs for Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 37-38.

18. Ivanov, Yu.V., Maslenikov, Yu.A., Chebykin, N.Ye., Shebarshin, A.V. and Shturts, I.V. "Questions Relating to Designing Systems for Automating the Development of Programs for Microcomputers," UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 5, 1978, pp 39-42.

19. Ryzhikova, A.I., Savitskaya, G.S. and Shebarshin, A.V. "System for Automating the Programming of Microcomputers Based on a Syntactical Translator Routine," ELEKTRONNAYA TEKHNIKA. SERIYA 3. MIKROELEKTRONIKA, No 3, 1978, pp 38-42.

20. Gendler, I.A., Ryzhikova, A.I. and Shebarshin, A.V. "System for Dialogue Debugging of Microcomputer Programs, Employing a Multipurpose Computer," ELEKTRONNAYA PROMYSHLENNOST', Nos 11-12, 1979, pp 114-116.

21. Maslenikov, Yu.A., Usik, V.G., Chebykin, N.Ye. and Shturts, I.V. "Automation of the Design of Digital Microprocessor Systems," ELEKTRONNAYA PROMYSHLENNOST', No 6, 1979, pp 24-29.

22. Butomo, I.D. and Kotlyarov, V.P. "Structure of the Programming Language for the 'Elektronika S5-01' Microcomputer," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 38-39.

23. Butomo, I.D., Kotlyarov, V.P., Morozov, N.B., Reznik, A.E. and Samochadin, A.V. "Editing-Debugging Complex for 'Elektronika S5' Series Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 6, 1979, pp 29-33.

24. Gal'perin, M.P., Maslenikov, Yu.A., Shebarshin, A.V. and Shenderovich, Yu.I. "Integration of Systems for the Computer-Aided Design and Automation of the

Programming of Microprocessors," ELEKTRONNAYA PROMYSHLENNOST', No 10, 1978, pp 5-9.

25. Finikov, V.A. "Organization of Data Support in the Overall Designing of LSIC's," ELEKTRONNAYA TEKHNIKA. SERIYA 3. MIKROELEKTRONIKA, No 4, 1975, pp 35-39.

26. Lyubich, L.G., Skvortsov, A.Ye. and Shebarshin, A.V. "Automated System for Documenting the Development and Ordering of Microcomputer ROM LSIC's," ELEKTRONNAYA TEKHNIKA. SERIYA 3. MIKROELEKTRONIKA, No 3, 1978, pp 43-47.

27. "Mini-EVM" [Minicomputers], translated from English, edited by E. Opperl', Moscow, Mir, 1975, 102 pages.

28. Lipayev, V.V. "Proyektirovaniye matematicheskogo obespecheniya ASU (skhemotekhnika, arkhitektura, tekhnologiya)" [Designing Software for Automated Control Systems (Circuitry, Architecture, Technology)], Moscow, Sovetskoye Radio, 1977, 400 pages.

29. Gal'perin, M.P. and Maslenikov, Yu.A. "Utilization of Facilities for Debugging Specific Routines for a Series of Multipurpose Microcomputers," UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, No 6, 1979, pp 51-56.

30. Sebern, M. "DEC LSJ-11--a Microcomputer-Based System Compatible with Minicomputers," TIIER, No 6, 1976, pp 58-68.

31. Whitbred, M. "Standards in Microcomputer Systems Design," MICROPROCESSORS AND MICROSYSTEMS, Vol 2, No 6, 1978, pp 329-332.

32. Antoshevskiy, V.S., Istomin, G.S. and Shpilevskiy, E.P. "Data Transmission System Subscriber Stations," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 82-84.

33. Knyazev, K.K., Levina, I.A., Lipinskiy, G.V., Lyashenko, V.S. and Syromyatnikova, I.L. "Code Sequence Generator Employing the 'Elektronika S5-11' Microcomputer," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 51-52.

34. Allen, M. and Lafri, D. "Synergism of a Microprocessor and IEEE Standard 488-1975," TIIER, No 2, 1978, pp 63-77.

35. Zhamov, V.A., Mindlin, N.Ya., Samburskiy, V.V., Tarasov, A.P. and Trilesnik, I.I. "Application of the 'Elektronika S5-11' Microcomputer to the Processing of Spectrophotometric Data," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 44-46.

36. Sigalov, V.I. and Yusifov, S.I. "Application of Microprocessors in Spectrum Analyzers," ELEKTRONNAYA PROMYSHLENNOST', No 10, 1978, pp 15-18.

37. Al'tshul', S.D., Voronkov, L.A., Gil'man, G.I., Tikhonov, V.S. and Khryapina, I.G. "Unit for Processing Chromatographic Information," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 46-47.

38. Tikhonov, V.S., Gil'man, G.I. and Al'tshul', S.D. "Application of Microcomputers in Automatic Testing and Control Systems," PRIBORY I SISTEMY UPRAVLENIYA, No 5, 1979, pp 9-10.

39. Zav'yalov, E.N., Ponamareva, N.I., Khil'ko, V.V. and Shamenkov, V.I. "Application of Microcomputers in Equipment for Laboratory Investigations," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, p 79.

40. Baranovskiy, A.L., Kalinichenko, A.N., Manilo, L.A., Nemirko, A.P. and Predtechenskiy, A.G. "Automated System for Monitoring the Condition of Cardiology Patients," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 80-81.

41. Dernov, D.S., Neroslavskiy, I.A. and Syroyegin, N.K. "Bioengineering Complex for Professional Screening," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 81-82.

42. Yeremeyev, Yu.M., Orekhov, S.A. and Podboronov, B.P. "Multipurpose Digital Regulator Based on a Microcomputer," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 48-51.

43. Anisimov, E.I., Korshunov, O.N., Krivoruchenko, V.S. and Martynov, V.A. "Control of a Ventilation Unit," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 69-72.

44. Gal'perin, M.P. and Maslenikov, Yu.A. "Questions Relating to Organizing the Application of Microcomputers," PRIBORY I SISTEMY UPRAVLENIYA, No 10, 1978, pp 7-9.

CONTENTS

8831
CSO: 1863/136

ROBOTS

PROGRAMMABLE ROBOTS

Leningrad UPRAVLENIYE ROBOTAMI OT EVM in Russian 1980 (signed to press 17 Jul 80) pp 2, 4, 260-261

[Annotation, excerpt from foreword and table of contents from book "Programmable Robots", by Yevgeniy Ivanovich Yurevich, Sergey Ivanovich Novachenko, Vladimir Anatol'yevich Pavlov, Nikolay Sergeyevich Teleshev and Mikhail Vladimirovich Ivanov, edited by Yevgeniy Ivanovich Yurevich, Izdatel'stvo "Energiya", 9,600 copies, 264 pages]

[Text] Recent progress in robot technology, methods and systems of controlling robots are described in the book. A comprehensive approach to the analysis and synthesis of the final control systems of robots is given. The basic design principles of a robot control algorithm system and stationary and mobile robot control algorithms are examined.

The book is intended for engineering technicians and scientists, engaged in the design and application of robots and may also be useful to students and graduate students of the corresponding specialties.

The book is written on the basis of the materials of works published in recent years in this field by the Special Design Office of Engineering Cybernetics (OKB TK) of the Leningrad Order of Lenin Polytechnical Institute imeni M. I. Kalinin.

Contents

7872
CSO: 1863/113

CLASSIFICATION OF ROBOT CONTROL METHODS AND SYSTEMS

Leningrad UPRAVLENIYE ROBOTAMI OT EVM in Russian 1980 (signed to press 17 Jul 80) pp 15-17

[Excerpt from book "Programmable Robots", by Yevgeniy Ivanovich Yurevich, Sergey Ivanovich Novachenko, Vladimir Anatol'yevich Pavlov, Nikolay Sergeyevich Teleshev and Mikhail Vladimirovich Ivanov, edited by Yevgeniy Ivanovich Yurevich, Izdatel'stvo "Energiya", 9,600 copies, 264 pages]

[Text] The applications of robots are expanding rapidly, embracing all new fields of human endeavor. Today there are the following basic applications of robots, which determine the basic types of robots from the standpoint of purpose: industry, agriculture, transportation, service, ocean and space exploration, scientific research, health and military.

The basic and most general characteristics of any robot, irrespective of the control system and of the area of application, are the following: the number of manipulators, the existence of transport systems, the number of degrees of freedom of individual manipulators and transport systems, type of working zone of manipulators, lifting capacity, types of instruments and type of execution.

We will discuss these robot characteristics briefly. Robots can have one, two, three and four manipulators. For instance, the "Tsiklon-3b" robot has two manipulators (Figure 1.3), the LPI-2 robot also has two (Figure 1.4), while the MP-2 robot has four manipulators. The classification of robots as stationary and mobile is related to the existence of transport systems. As was mentioned in §1.1, transport systems can be based on different types of propulsion, the choice of which is determined by the requirements and conditions of robot movement (speed, mobility, lifting capacity, etc.).

The number of degrees of mobility of a robot determines the variety of motions both of manipulators and of the entire robot as a whole. The number of degrees of mobility of an individual manipulator ranges from 3 to 10. The degrees of mobility of the robot itself are added to the degrees of mobility of a manipulator of a mobile robot. The number of degrees of mobility varies from 1 to 3 for the center of mass and up to 3 for corner movements. An example of a mobile robot with one degree of mobility of the body is a robot that travels on a rail -- robots of the "Sport-1" type (Figure 1.5), "Sprut-1" type (Figure 1.6) and MP-1 type. An example of a robot with six degrees of mobility of the body is a mobile robot (Figure 1.7).

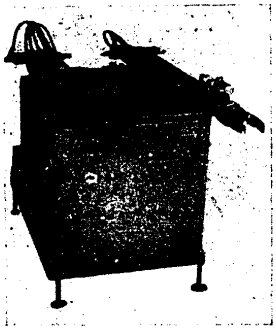


Figure 1.3. General view of "Tsiklon-3b" robot.

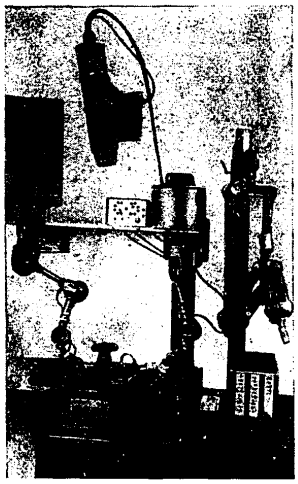


Figure 1.4. General view of LPI-2 robot.

Figure 1.5. General view of "Sport-1" robot.

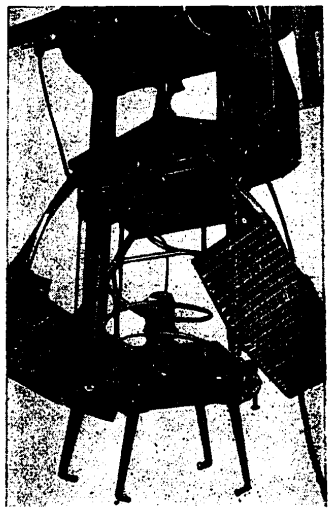


Figure 1.6. General view of "Sprut-1" robot.

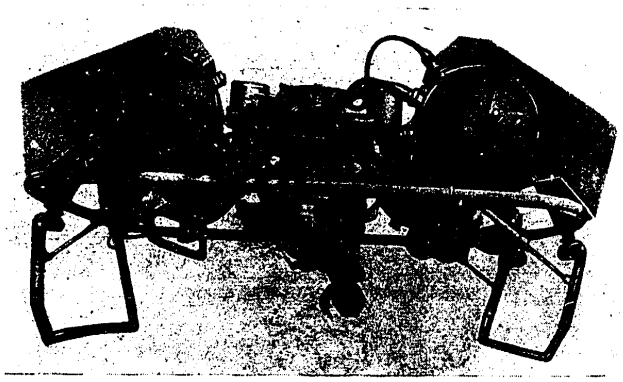


Figure 1.7. General view of mobile robot.

One of the important characteristics of a robot is the type of working zone (service zone). The working zone of a stationary robot is determined by the number of degrees of mobility of the manipulator and of its kinematic system, and by the relative sizes of the links. The type of working zone may be determined qualitatively by the following criteria: working zone on a plane, on a surface, cylindrical, parallelepiped, spherical or combined. To each type of working zone corresponds a coordinate system, in which are planned the grasping movement of a robot, and accordingly algorithms for calculating the spatial positions of the links of the manipulator for achieving a given trajectory of grasping movement.

The lifting capacity of a robot is determined by the weight of the objects to be lifted and of the working organs and may vary in a wide range -- from several tons to fractions of a gram.

Robot drives, as was already mentioned, can be electric, hydraulic and pneumatic. The choice of type of drive is determined not only by the lifting capacity, precision and dynamic characteristics of a robot, but also by its range of application. For instance, pneumatic and hydraulic drives are not recommended for use in space robots.

Robots are manufactured in different executions, depending on operating conditions: normal, hermetically sealed, heat-insulated, explosion-safe, etc.

The basic characteristics of robots from the standpoint of control are: the principles of movement control, the nature of human operator participation in control, the number of simultaneously controlled robots or the number of manipulators of one robot, movement control quality, principles of control system organization, and method of hardware-software execution of the control system.

7872
CSO: 1863/113

ROBOT SENSOR SYSTEMS

Leningrad UPRAVLENIYE ROBOTAMI OT EVM in Russian 1980 (signed to press 17 Jul 80) pp 54-58

[Excerpt from book "Programmable Robots", by Yevgeniy Ivanovich Yurevich, Sergey Ivanovich Novachenko, Vladimir Anatol'yevich Pavlov, Nikolay Sergeyevich Teleshev and Mikhail Vladimirovich Ivanov, edited by Yevgeniy Ivanovich Yurevich, Izdatel'stvo "Energiya", 9,600 copies, 264 pages]

[Text] We proceed to an examination of the designs of certain robot sensor systems, developed by OKB TK [Special Design Office of Engineering Cybernetics].

Tactile Sensors. These primary converters belong to the simplest class of robot sensor systems. They are used for signaling grasping contact of a manipulator or of the body of a robot with objects of the environment and can be used for solving certain recognition problems and for determining the dimensions of objects [74]. The basic deficieny of tactile sensors, as was already mentioned, is the fact that they usually place certain limits on the speed of a robot. A diagram of a tactile sensor for sensitizing the grasp of a manipulator, which is partly devoid of this deficiency, is presented in Figure 2.3. The body of the sensor, consisting of membrane 3 and feelers 2, is made of an elastic material (for instance, vulcanized rubber). Deformation of any of the feelers of the sensor when an obstacle is touched is transmitted to the membrane. Permanent magnet 1, fastened to the inside of the membrane, moves and actuates sealed contact 4. The sensor actuates upon a slight deformation of the feelers. Because of their elasticity movement of the gripping device toward the activated sensor in the manipulator braking does not disturb or damage objects. The force sensitivity of the sensor is 0.15-0.20 N. A general view of the gripping device of the manipulator of a mobile robot, sensitized by elastic sensors, is shown in Figure 2.4.

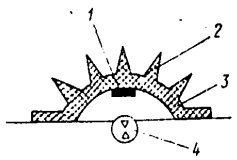


Figure 2.3. Diagram of elastic tactile sensor.

Contactless Sensors for Sensitizing Gripping Devices of Manipulators. A robot gripping device often is equipped with a contactless receptor field in order to improve the reliability and speed of working operations. This problem can be solved with the aid of lidar or ultrasonic sensors. A functional diagram of a lidar sensor [74] is shown in Figure 2.5.

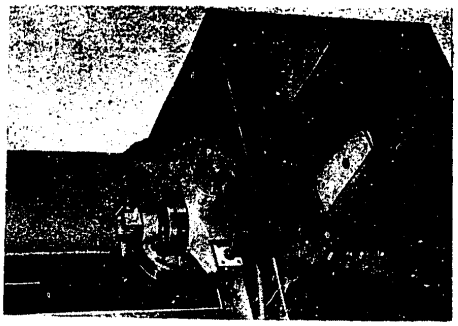


Figure 2.6. General view of sensitized gripping device of industrial robot.

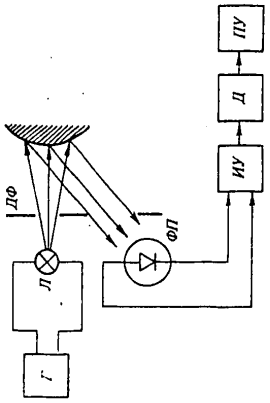


Figure 2.5. Functional diagram of lidar sensor. [Г=G; Л=L; ДФ=DP; ФП=PR; ИУ=SA; Д=D; ПУ=TS]

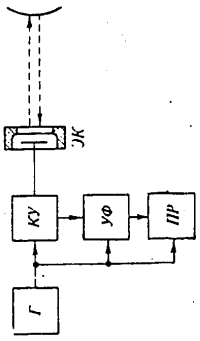


Figure 2.7. Functional diagram of ultrasonic sensor. [Г=G; КУ=SS; ЭН=EC; ПР=CON]

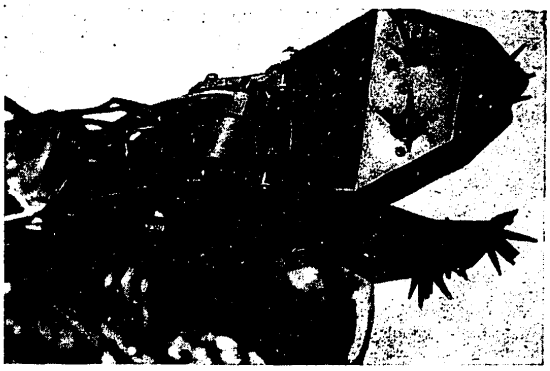


Figure 2.4. General view of sensitized gripping device of mobile robot.

The light beam produced by an incandescent lamp is modulated by changing the power voltage from generator G. Passing through diaphragm DP, the light beam is radiated into space. When any obstacle appears in the field of that radiation the light beam is reflected. Then some of the reflected light strikes photoreceiver PR and is then amplified by selective amplifier SA, tuned to the modulation frequency of the radiation, and after being detected it goes to threshold system TS. The latter generates a signal when the received signal crosses a given threshold, proportional to the distance being measured. The sensor has the following characteristics: range 0.03-0.07 m; activation time 30 ms; maximum error 30%.

A general view of the gripping device of an industrial robot, sensitized with lidar sensors, is shown in Figure 2.6. By virtue of their simplicity and compactness these systems were the basic type of short-distance sensors of the first experimental sensitized robots. However, their deficiencies, such as sensitivity to fluctuations of the external light level, the transparency of the atmosphere, reflectivity of objects and poor precision, prompted the development of different types of sensor systems with high flexibility in relation to the properties of the environment and objects and with high distance measurement precision. Ultrasonic sensors [74] produce good results in this area.

A functional diagram of an ultrasonic sensor for sensitizing the gripping device of a manipulator is shown in Figure 2.7. Its operating principle consists in the acoustic sounding of the space near the gripping device. Generator G produces individual short electronic high-voltage pulses. These pulses pass through switching system SS to the electrode of electrostatic capsule EC. Under the influence of the electrostatic field the membrane of the capsule is deformed, radiating an ultrasonic pulse into the air, which after being reflected from an object is perceived by the same capsule. The received pulse goes to an amplifier-shaper, which is locked during the time of radiation to prevent false activation. The shaped pulse then goes to converter CON, which produces a pulse equal in duration to the time interval between the sounding and received pulses and proportional to the distance to the point of reflection.

A general view of the gripping device of the manipulator of the LPI-2 robot, sensitized with 12 such ultrasonic sensors, is shown in Figure 2.8. The sensors have the following characteristics: distance measurement error not more than 2%; effective range 0.01-0.7 m; activation time not longer than 10 ms; area of membrane 80 mm^2; width of radiation pattern of individual sensor about 60°.

Ultrasonic Scanning Range Finder. Ultrasonic range finders may be used for measuring distances in the 0.1-5.0 m range in air or water with moderate precision. These range finders have known advantages under these conditions over optical or radar range finders [14]. A functional diagram of a range finder, based on the principle of stereophonic ultrasonic sounding [74], is shown in Figure 2.9. The range finder has two identical ranging channels, each of which is analogous to the above-described ultrasonic sensor. However, the greater effective range of the instrument is the result of some of its distinctions from the sensors of a gripping device. The atmosphere is sounded with parcels of sinusoidal ultrasonic pulses, generated by pulse generator PG. Ultrasonic wave radiator R is structurally detached from receivers Rec1, Rec2. It represents the above-described electrostatic capsule, positioned between analogous capsule-receivers.

Figure 2.8. General view of sensitive gripping device of LPI-2 robot.

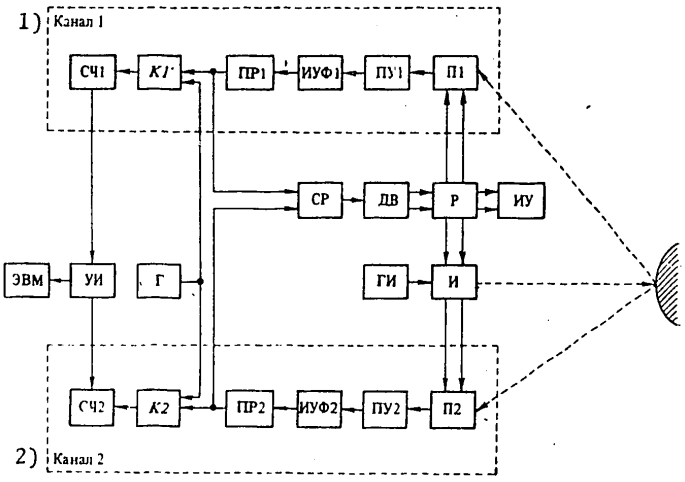


Figure 2.9. Functional diagram of ultrasonic range finder.
Key: 1. Channel 1 2. Channel 2
[СЧ=CT; ПР=CON; ИУФ=SAS; ПУ=PA; П=Rec; СР=CS; ДВ=MOT; Р=Red; ИУ=RG; ЭВМ=Computer; УИ=IS; Г=G; ГИ=PG; И=R]

The receivers are placed deep inside wave guide channels to increase the directivity of the range finder and to reduce the influence of interference. Signal preamplifiers PA1, PA2 are installed in the same case with each capsule-receiver. Additional amplification of a received signal takes place in selective amplifier-shapers SAS1, SAS2, located in the information processing unit. They are tuned to the frequencies of the sounding pulses, which greatly improves the noise immunity of the system. The pulses go from the amplifier-shaper of each channel to a converter (CON1 or CON2), which produces a width-modulated pulse, the duration of which is proportional to the distance to the reflecting surface. The shaped pulses are filled in with pulses of a reference frequency, which pass through keys K1 and K2 from the pulse generator. The distance to an object is determined on the basis of the number of the pulses, recorded by counters CT1 or CT2. This distance is displayed on digital indicator system IS and is fed into a computer.

Comparison system CS compares the duration of the width-modulated pulses of both channels and, depending on the sign of the mismatch, sends one control signal or the other to electric motor MOT. The latter, through reducer Red, rotates the sonar system until the object is aligned with the axis of the sonars, i.e., until the distances between the object and Rec1 and Rec2 are equal. In this position a signal is taken from rotation goniometer RG, which determines the azimuthal coordinate of the object in the plane of the working zone. When three channels are used it becomes possible to determine spherical coordinates of objects in space (azimuth, elevation and range). The described range finder has the following characteristics: measureable range 0.3-3.3 m; range measurement error ±1%; resolving power at 1 m a 160 mm diameter ball; input energy (including information processing unit) not more than 30 W; weight with drive 0.4 kg; weight of information processing unit 1.6 kg. A general view of the range finder and of the information processing unit is presented in Figure 2.10.

Figure 2.10. General view of ultrasonic range finder.

Parallax Near-Range Laser Range Finder. Ultrasonic ranging methods are not suitable for measuring geometric parameters of objects in the environment of a robot (1-3 m)

with high resolving power. Laser phase range finders are also difficult to use because of their low speed [15]. This problem can be solved successfully with a parallax laser range finder, developed jointly by OKB TK and SZPI [Northwestern Correspondence Polytechnic Institute (Figure 2.11). The range finder is intended for measuring topographic elevation in the zone around a mobile robot.

7872
CSO: 1863/113

HUMAN OPERATOR-ROBOT COMMUNICATIONS SYSTEMS

Leningrad UPRAVLENIYE ROBOTAMI OT EVM in Russian 1980 (signed to press 17 Jul 80) pp 64-76

[Excerpt from book "Programmable Robots", by Yevgeniy Ivanovich Yurevich, Sergey Ivanovich Novachenko, Vladimir Anatol'yevich Pavlov, Nikolay Sergeyevich Teleshev and Mikhail Vladimirovich Ivanov, edited by Yevgeniy Ivanovich Yurevich, Izdatel'stvo "Energiya", 9,600 copies, 264 pages]

[Text] The organization of the interaction between the human operator and robot is very important for the effective participation of man in the control of a robot, and for the joint completion of tasks. At the different levels of robot control examined in §1.2, this interaction takes place through various languages and hardware. Communications systems are divided into two groups (Figure 2.18): master systems of the operator and data display systems for displaying information that comes from the robot.

The master systems include command consoles, selector panels, mechanical analogs of final control organs and their parts, target indication systems, standard computer terminals and voice command systems.

Command consoles and selector panels are used for sending digital and analog commands for controlling individual robot subsystems by manual control (for instance manipulator drives), for switching robot routine programs or for sending operator instructions during supervisory control and during programming of robots. A control console of the "Retab" industrial robot and the selector field of the "Matbac" robot are shown in Figure 2.19.

Mechanical analogs of final control systems and their parts are used as master controls in the copying control mode and in velocity vector control. Master manipulators and arms are used extensively here. With the latter it is possible to assign, for instance, the movement velocity vector of the tool of a manipulator. A general view of the master manipulator of the LPI-2 experimental robot is shown in Figure 2.20.

Target indication systems are used for selecting target objects in the working zone of a robot and for measuring their spatial coordinates. Different kinds of range finders (ultrasonic, laser, etc.) may be used for this purpose, but they must be equipped with systems that enable the operator to visually monitor the aiming of the range finder at a selected target.

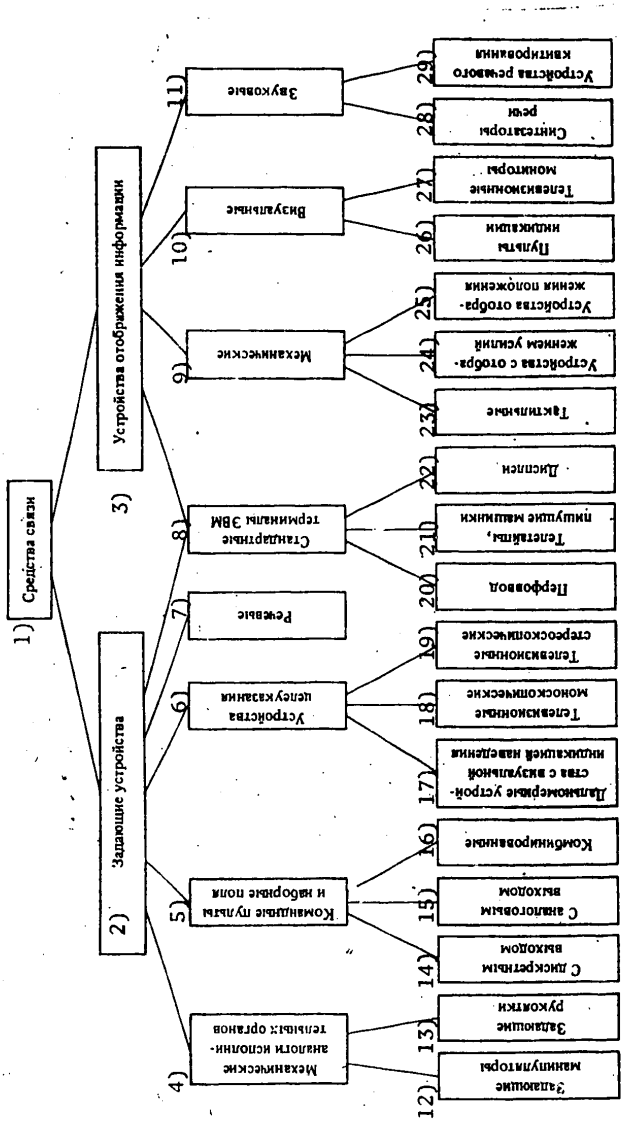


Figure 2.18. Classification of operator-robot communications systems.

1. Communications systems
2. Master systems
3. Information display systems
4. Mechanical analogs of final controls
5. Command consoles and selector fields
6. Target indication systems
7. Voice
8. Standard computer terminals
9. Mechanical
10. Visual
11. Acoustical
12. Master manipulators
13. Master arms
14. With digital output
15. With analog output
16. Combined
17. Range finder systems with visual guidance indication
18. TV monoscopic
19. TV stereoscopic
20. Perforated tape input
21. Teletypes, typewriters
22. Displays
23. Tactile
24. Force display systems
25. Position display systems
26. Indication consoles
27. TV monitors
28. Speech synthesizers
29. Voice acknowledgement systems

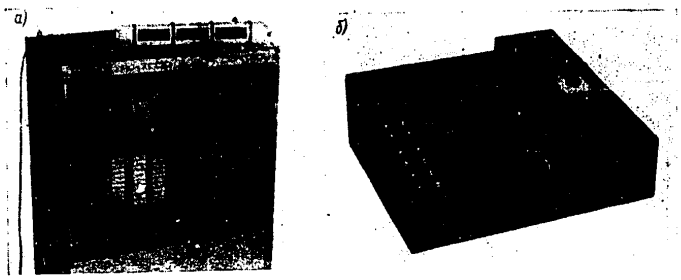


Figure 2.19. Control console of "Retab" industrial robot (a) and selector field of "Matbac" robot (b).

Figure 2.20. General view of master manipulator of LPI-2 robot.

The target control systems for remote-controlled robots should be developed on the basis of scanning TV robot systems. A TV target indication system, developed by OKB TK [Special Design Office of Engineering Cybernetics], and a general view of which is shown in Figure 2.21, will be described below.

Voice command systems enable the operator to give a robot instructions in the most natural way. Here the operator can use standard oral voice communications systems for remote transmission of instructions to the robot, which is particularly important for the joint completion of tasks by people and robots. Voice control of robots still does not enjoy extensive application in connection with the considerable complexity of speech pattern recognition problems. There are only a few of such systems with a dictionary, limited to two-three hundreds of words, and operated by one-two speakers. A general view of a 200-command voice command system, developed at OKB TK for robot control, is shown in Figure 2.22.

In addition to the master systems described above, standard computer terminals -- teletypes, console typewriters, displays, etc., may be used for human operator-robot communications.

Information display systems are divided into mechanical, visual and acoustic. Mechanical display systems are used for the most complete and natural involvement of the neuromotor system of a human operator in the robot control process by means

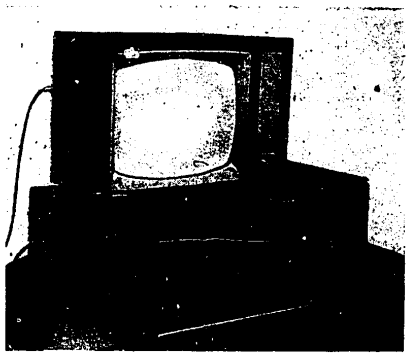


Figure 2.21. General view of target indication system.

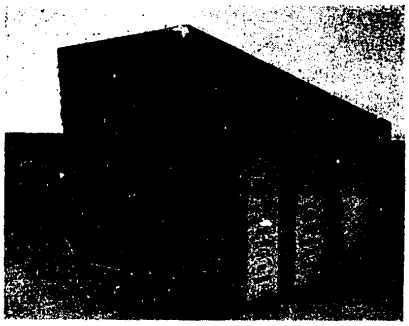


Figure 2.22. General view of voice command system.

of active dynamic display of information about the interaction of a robot and surrounding objects, and about its spatial orientation. Examples of the use of such systems are tactile displays [75], manipulators with force display [36], a mobile operator console that displays the spatial orientation of underwater work [79].

Visual information display systems comprise the largest group, including information consoles, plotters, scanning robot TV systems and other sources of information of visual feedback, and also standard computer terminals.

Acoustic display systems act upon the human hearing organs. In the simplest case acoustic feedback (in the form of individual signals) can be used for attracting the operator's attention or for signaling a danger. In more complex cases a sound signal can carry information about the interaction of a robot with the environment or about the condition of its subsystems; for instance, the frequency of an acoustic signal can carry information about the force developed by the tool of the manipulator. The best form of acoustic communications is oral voice communications between a robot and human operator. Voice communications can be achieved through speech synthesizers or with voice acknowledgement systems. One variety of such a system will be described below.

The above classification by type and physical attributes does not reflect the classification of these systems by utilization on different levels of the control system. Such an hierarchical structure of man-robot communications systems is given in Table 2.3.

As can be seen in the table, on low robot control levels (I-III) man-robot and robot-man communications systems exhibit a considerable difference. On control levels IV and V the communications systems tend to be more alike and the languages approach natural human language. At the same time the master systems and information display systems, which on level V blend into a single control system, show the same tendency.

We will proceed to a more detailed examination of some standard man-robot communications, developed by OKB TK.

Table 2.3. Structure of Man-Robot Communications Systems

Control level	Master systems	Display systems	Language
I	Command buttons; potentiometers	TV; telemetry indication systems	--
II	Master manipulators; master arms	TV; telemetry indication systems and force display systems	--
III	Target indication and instruction systems	TV; generalized information display systems	
IV	Teletypes; typewriters; displays; voice command systems	Teletypes; typewriters; displays; speech synthesizers; speech reply systems	Task-oriented
V	Same	Same	Limited natural

TV Target Indication System. Shown in Figure 2.23 is a structural diagram of a TV target indication system, used in underwater work with supervisory control [69].

Let us examine the operation of the system. The operator, by looking at an image of the robot's working zone on the TV screen, selects the object-target in which he is interested and puts an electronic mark on it. The position of the mark is controlled with a two-stage master lever, connected to two orthogonal potentiometers. The line and frame voltages U_{lin} and U_{fr}, taken from the potentiometer, go to coincidence circuits. The other inputs of these circuits receive the appropriate line and frame scanning voltages. As soon as these systems are actuated the logic system that generates a marker video pulse is started, and this pulse is sent to the input of the video channel of the TV receiver, where it is mixed with the video signal of the image. The position of the mark relative to the beginning of a frame and line is determined uniquely by voltages U_{fr} and U_{lin}, selected by the operator. The image brightness of the mark is controlled by the amplitude of the video pulse. By changing the polarity of the video pulse it is possible to make a bright or dark image of the mark.

Voltages U_{lin} and U_{fr} are fed into an angular coordinate computer (an electronic digital computer), which computes angular coordinates α_t and β_t. These angles are computed by formulas (2.80). As was already mentioned in §2.2, additional range finding systems or automatic optical focusing can be used for determining the third target coordinate -- range.

The described target indication system can also be used in a stereoscopic TV system. In this case the operator must indicate the object on the screen two times for images, translated alternately by each TV camera. A general view of a supervisory control console for controlling the manipulator of a mobile robot, and containing the examined target indication system, is shown in Figure 2.24.

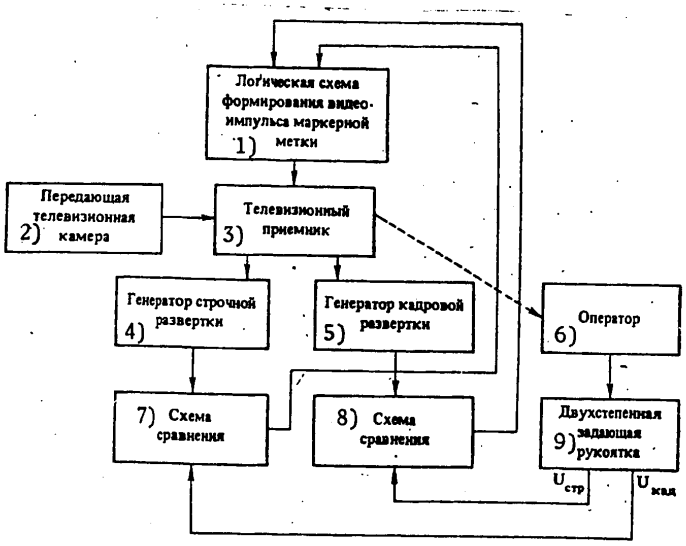


Figure 2.23. Structural diagram of TV target indication system.

1. Logic system generates mark video pulse
2. Transmitting TV camera
3. TV receiver
4. Line scanning generator
5. Frame scanning generator
6. Operator
7. Comparator
8. Comparator
9. Two-stage master arm

[стр=lin; над=fr]

Voice Command System (VCS). Work is done basically in two directions by automatic speech recognition [25]. One of these directions -- phoneme recognition, is aimed at the global solution of the problem of automatic speech recognition. The second direction is the search for methods of automatic recognition of a limited set of instructions. The solution of the first problem naturally will be the solution of the second. However, automatic phoneme recognition encounters considerable difficulties and necessitates the construction of extremely complicated systems. In addition it is necessary in many cases to use systems that react to a comparatively small set of commands. It would be uneconomical in this case to use methods based on phoneme recognition. A more intelligent approach is to find a simple and inexpensive way of solving a given specific problem.

Many systems have already been developed that solve the problem of the recognition of a limited vocabulary. However, it is important to point out that most works on voice command recognition are more theoretical than applied in character. The use of large computers, which model certain recognition algorithms, can hardly be assumed acceptable in practice. Recognition algorithms of this kind are based on spectral band, formant, correlation and certain other methods.

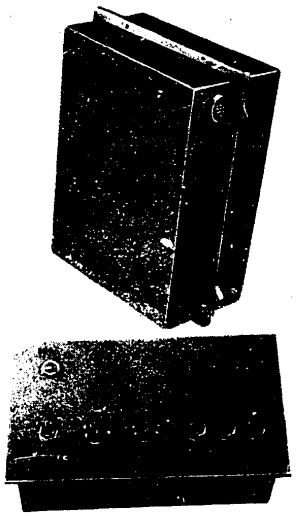


Figure 2.24. General view of supervisory control console.

A structural diagram of a voice command system for controlling a robot, developed at OKB TK by V. A. Novikov, et al, [67], is shown in Figure 2.25.

A speech signal (command) goes into a compressor system, which is used for compressing the dynamic range of the speech signal by 30 dB. The compressed signal goes into a filter unit, which consists of 11 identical narrow-band envelope shapers (Figure 2.26). Each envelope shaper contains a bandpass filter F, detector D, low-pass filter LPF and zero level hold ZLH. The bandpass filters are active filters with a steepness of not less than 40 dB/dec. The passband of the first filter on the 0.7 level is 100 Hz and increases by the 11th to 600 Hz. The signals from the filters go into adders for making the envelopes of the first half of the first formant, the second half of the first formant, etc. The signals from the adders go to detectors, which are HF limiters with a 40 dB linearity range. The detected signals pass through low-pass filters with an 80 dB/dec falloff steepness and a cutoff frequency (at the 0.7 level) of 10 Hz. To eliminate the DC voltage from the LPF output due to the instability of previous elements and the existence of noise at the microphone output (motor, fan and other like noises), the zero level holds are used. The tolerable noise in terms of voltage is 0.3 of the maximum useful signal. The speech signal, after undergoing preprocessing in the compressor and filter unit, goes into the attributes unit (Figure 2.27), where characteristic features (attributes) of the received speech envelopes are identified. Attributes are sorted out in accordance with the following criteria: a) invariance to rate; b) invariance to speaker; c) invariance to loudness; d) simplicity.

The following attributes were selected as the result of an analysis: 1) number of segments of an envelope with larger than a given area, lying above a certain threshold; 2) number of the segment with the largest area; 3) amplitude ratio of the first three signals; 4) amplitude of the initial segment.

Five channels must be used in VCS for the described attributes, and to the input of each of them may be fed any envelope from the filter unit.

The most informative envelopes in the frequency bands given below were selected on the basis of the results of experimental investigations: 1) the envelope in the entire frequency range; 2) the envelope corresponding to the second half of the first formant; 3) the first and second halves of the second formant; 4) the second formant.

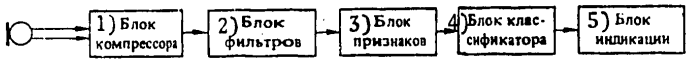


Figure 2.25. Structural diagram of voice command system.
Key: 1. Compressor unit
2. Filter unit
3. Attributes unit
4. Classifier unit
5. Indication unit

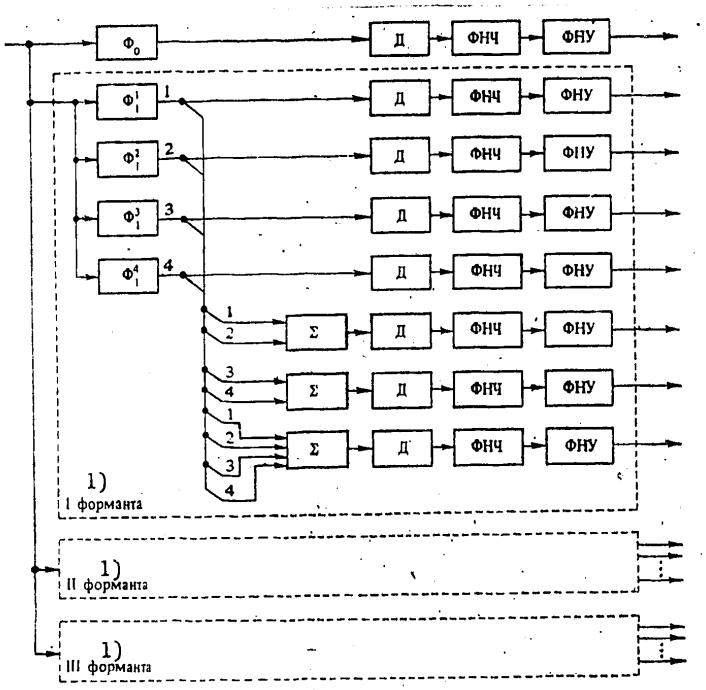


Figure 2.26. Structural diagram of filter unit.
Key: 1. Formant
[Ф=F; Д=D; ФНЧ=LPF; ФНУ=ZLH]

The selected attributes of the speech signal go into a classifier (Figure 2.28), whose functions include logic processing of information about the speech signal for the purpose of identifying a speech signal with a command stored in the robot's memory. The signals are classified in accordance with an attribute processing algorithm, based on the processing of statistical material. Hyperplanes, separating commands in the space of attributes, were plotted as the result of this processing. The "delete" method, the essence of which is explained below, was developed on the basis of the processing algorithm used in the VCS classifier.

We will examine an m-dimensional space of attributes, characterizing N commands. Each command is represented in the space of attributes by an m-dimensional

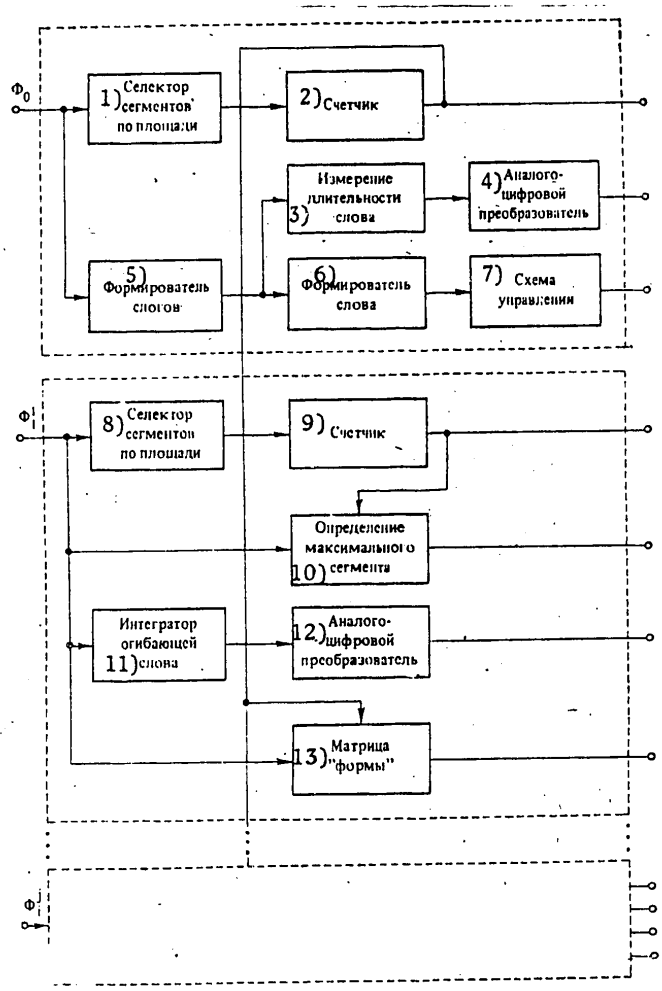


Figure 2.27. Structural diagram of attributes unit.

Key: 1. Selection of segments by area
2. Counter
3. Measurement of word length
4. Analog-digital converter
5. Syllable generator
6. Word generator
7. Control system
8. Selection of segments by area
9. Counter
10. Identification of maximum segment
11. Word envelope integrator
12. Analog-digital converter
13. "Form" matrix

hyperparallelepiped, the sides of which are parallel to the coordinate axes. Edge length is $l_{ij} = 2k\sigma_{ij}$, where i is the number of command, j is the number of attribute, σ_{ij} is the mean square deviation of the values of an attribute, k is a constant.

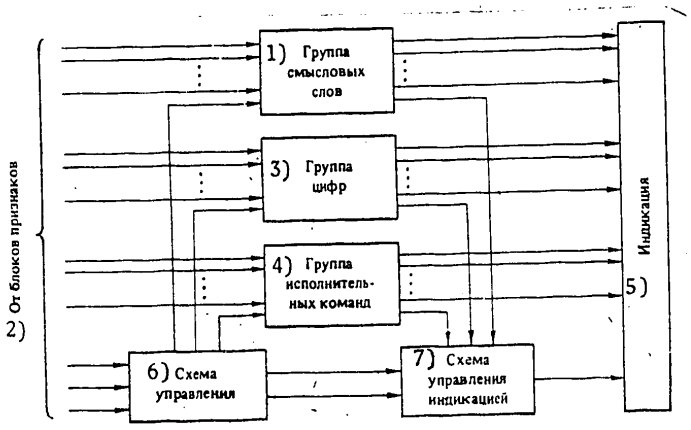


Figure 2.28. Structural diagram of classifier.

Key: 1. Semantic vocabulary
2. From attributes unit
3. Group of numbers
4. Execution command group
5. Indicator
6. Control system
7. Indicator control system

For recognition probability P = 0.95 and average number of attributes n = 4 we have k ≈ 2 (the hypothesis that the probabilities of the attributes have a normal distribution is accepted).

In order to solve the problem of classification at the lowest equipment cost it is necessary to draw dividing hyperplanes that minimize the number of attributes necessary for the recognition of a given command with the fewest total attributes utilized for recognizing all commands of the vocabulary. We introduce quality functions F_j, characterizing the average number of commands "deleted" by the j-th attribute. These functions are

$$F_j = \frac{1}{N}\sum_{i=1}^{N} F_{ij}; \quad F_{ij} = \frac{1}{N-1}\sum_{k=1}^{N} \rho_{ijk}, \tag{2.85}$$

where ρ_{ijk} expresses the number of attributes, on the basis of which a given command k is separated from investigated command i:

$$\rho_{ijk} = \begin{cases} 1, & \text{if the j-th attribute separates command i from k;} \\ 0 — & \text{otherwise.} \end{cases}$$

The use of functions F_j is based on the assumption that coefficients F_{ij} are sufficiently close for different commands. To find the minimum number of attributes necessary for recognizing each i-th command it is necessary to solve the linear programming problem

$$\min CX, \quad AX \geqslant B,$$

where X is an m × 1 attribute matrix, and

$$X_i = \begin{cases} 1, & \text{if the j-th attribute is used for recognizing given i-th command;} \\ 0 & \text{otherwise;} \end{cases}$$

C is a 1 × m matrix with element $c_j = 1$; A is an N × m matrix with elements $a_{kj} = \rho_{ijk}$; B is an N × 1 matrix with elements $b_{k=j} = 1$; $b_{k\neq j} = 0$.

By using $c_j = F_j^{-1}$ instead of $c_j = 1$, it is possible to reduce the total number of utilized attributes. The attributes needed for recognizing each command are selected in accordance with a recurrent procedure, which in each step provides the maximum value of function F_j. Here max F_j is sought in the set of attributes that separate the i-th command from a command that differs from it in the fewest attributes. The described algorithm makes it possible in each step to select the attribute that on the average provides the greatest separation of commands. Since as the result of this procedure uninformative attributes are excluded, the total number of selected attributes approaches the minimum number of attributes necessary for solving the problem as stated.

The VCS classifier is adjusted on the basis of the results of the processing of statistical information, obtained from an analysis of the voices of a group of speakers, consisting of four to eight people. This distinguishes the described VCS from the existing analogs in the USSR and abroad, which are tuned to the voice of just one speaker. Test results of VCS indicated the following. The probability that the system will recognize the commands of the speakers to whose voices the classifier is tuned is 0.98, and for speakers whose voices were not analyzed it is 0.94. In the latter case tests were conducted with speakers, both men and women (without retuning).

Voice Acknowledgement System. As is known, the voice spectrum can be confined in the 300-3,000 Hz band, and the dynamic range to 30 dB without significantly impairing the quality and intelligibility of speech. In this case the amount of information in a 1 s long segment of a speech signal is about $3 \cdot 10^4$ bits [67]. Therefore, to design a voice reply system it is desirable to use analog speech signal storage, in which moving information carriers -- magnetic tape, disks, etc., are used.

A structural diagram of such a system is presented in Figure 2.29. The robot's vocabulary (256 words) is stored in a storage system with 1 m long 25 mm wide magnetic tape loops. The information on a tape is recorded on 12 tracks; each track is divided into zones, or segments of tape, in each of which is recorded one

word. The zones are divided by between-zone spaces, necessary for preventing the loss of information during starting, stopping and switching of tape speeds. The address method is used for retrieving words. The address of any word is assigned by the four-digit code of the number of a track and five-digit code of the number of zone, and a 13th track is used for recording the codes. The retrieval time does not exceed 0.5 s per word. Words are played back at a speed of 1 cm/s.

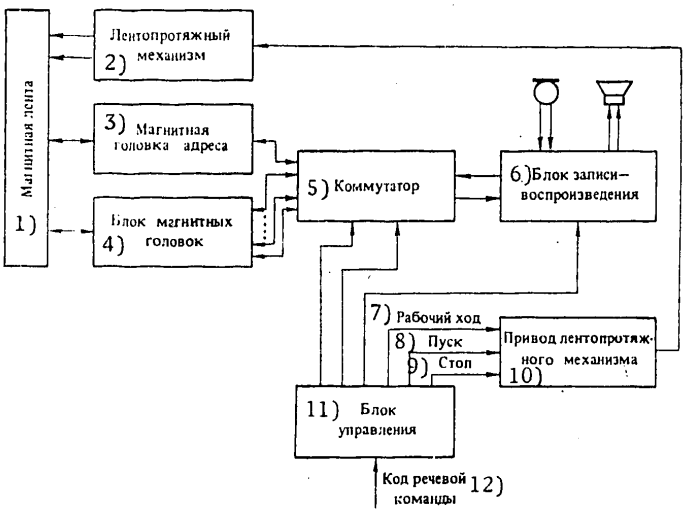


Figure 2.29. Structural diagram of voice acknowledgement system.

Key:
1. Magnetic tape
2. Tape transport mechanism
3. Address magnetic head
4. Magnetic head unit
5. Switch
6. Record-playback unit
7. Run
8. Start
9. Stop
10. Tape transport mechanism
11. Control unit
12. Voice command code

Let us examine in greater detail the operation of the speech acknowledgement system. The address code of a word of a voice command goes from the computer to the control unit, which switches the tape transport drive to the retrieval mode. As the tape runs the codes of the zones are played from the 13th head and are fed into the control unit, where they are compared with the zone address code. When the addresses match the control unit switches the tape transport drive to the playback mode and simultaneously connects to the record-playback unit the magnetic head that corresponds to the address track. Then it asks the computer for the next address. After the words are played the control unit, having received a new address, repeats the described operating cycle; otherwise it sends to the computer a signal to finish the job.

The robot's vocabulary is revised or changed by the learning method. The operator assigns the address of a word, on the basis of which the control unit automatically finds the necessary segment of tape (zone). Here the unit does not switch the tape transport drive to the playback mode, but sends the appropriate signal to the operator's console. Before recording a command the operator turns on the tape transport mechanism from the console.

The examined voice acknowledgement system has the following specifications: vocabulary 256 commands; dimensions 0.5 × 0.3 × 0.2 m; weight 3.5 kg; input power 30 W.

7872
CSO: 1863/113

'BARS' ALGORITHM SYSTEM

Leningrad UPRAVLENIYE ROBOTAMI OT EVM in Russian 1980 (signed to press 17 Jul 80) p 133

[Excerpt from book "Programmable Robots", by Yevgeniy Ivanovich Yurevich, Sergey Ivanovich Novachenko, Vladimir Anatol'yevich Pavlov, Nikolay Sergeyevich Teleshev and Mikhail Vladimirovich Ivanov, edited by Yevgeniy Ivanovich Yurevich, Izdatel'stvo "Energiya", 9,600 copies, 264 pages]

[Text] The "Bars" robot control algorithm system was developed by OKB TK [Special Design Office of Engineering Cybernetics] [42, 51, 56]. The system is built on the modular principle, which makes it easy to modify it. The "Bars" system was debugged on the LPI-2 experimental sensitized robot and on several Soviet industrial robots. The system controls a group of robots, both nonsensitized and sensitized, in the supervisory and automatic modes in real time. Its speed is such that one computer can control simultaneously up to 40 unsensitized, or up to 15 sensitized robots. The robots are programmed in the dialogue mode by the operator in the special ROCOL language (see Chapter 5). The programming process can be accomplished concurrently with the control of a preprogrammed robot. The structure of the "Bars" system and the working algorithms of the individual program modules are examined in greater detail in the next sections.

By way of concluding this section we should like to mention that in addition to the above-mentioned developments, specialized robot control systems with computer control are being developed at the University of California at Berkeley (United States) [87, 94], Massachusetts Institute of Technology (United States) [71], Edinburgh University (Great Britain) [7], Leningrad Aviation Instrumentation Institute [26, 27], Cybernetics Institute of the UkSSR Academy of Sciences [2, 50], Applied Mathematics Institute of the USSR Academy of Sciences [54, 72], MVTU [Moscow Higher Technical School imeni N. E. Bauman] [60] and a number of other organizations.

7872
CSO: 1863/113

SOFTWARE

ABSTRACTS FROM THE JOURNAL 'PROGRAMMING'

Moscow PROGRAMMIROVANIYE in Russian No 1, Jan-Feb 81 (signed to press 15 Jan 81) pp 95-96

UDC 681.3.06

SEMANTIC STRUCTURES OF PROGRAMS

[Abstract of article by Red'ko, V. N.; received by editors on 22 Sep 80]

[Text] Complete (general-purpose) program logics are constructed, the conclusions in which adequately reflect the design of the programs. Based on them, concepts of semantic structure and semantic design are refined. All basic concepts and results are illustrated by substantive examples. There are 3 figures, 1 table and a bibliography with 2 titles.

UDC 681.3.323

ALGORITHM SCHEMATA CORRECTNESS ANALYSIS METHOD

[Abstract of article by Anishev, P. A.; received by editors on 24 Aug 79]

[Text] Conditions of correctness of parallel schemata of algorithms (PGSA) are derived, based on properties of liveliness and safety of Petri nets. Defined are rules for transfer from PGSA to Petri nets that permits reducing checking it for correctness to analysis of the corresponding Petri net. There are 4 figures, 2 tables and a bibliography with 8 titles.

UDC 51:621.391

SCHEMATA OF PROGRAMS WITH CYCLICAL INTERPRETATIONS

[Abstract of article by Lisovik, L. P.; received by editors on 7 Jun 79]

[Text] Concept of cyclical interpretation of schemes with one unary base functional symbol is defined. It is proven that the problem of equivalence with cyclical interpretations is insolvable for unary recursive schemes. It is shown that this problem is solvable for Yanov schemes with finite-reversible counters. Bibliography with 4 titles.

UDC 681:519

PROBLEM OF REALIZATION OF COMPOUND CYCLES

[Abstract of article by Lyubimskiy, E. Z., and Mitashyunas, A. Yu.; received by the editors on 10 July 1980]

[Text] A formal apparatus is proposed to describe the realization of compound cycles. Within the bounds of this apparatus, the problem of optimal realization of compound cycles is discussed. There are 2 figures and a bibliography with 3 titles.

UDC 681.3

A PRACTICAL METHOD OF ANALYSIS AND CONVERSION OF SEQUENTIAL PROGRAMS TO PARALLEL FORM

[Abstract of article by Khoroshavina, G. F.; received by the editors on 14 Dec 1979]

[Text] A method is given in the article to automate conversion of sequential programs to parallel form. There are 2 tables and a bibliography with 5 tables.

UDC 681.3.06

DECISION TABLE TRANSLATION

[Abstract of article by Yeremeyev, A. P.; received by the editors 1 Jul 1979]

[Text] Problems of translating decision tables (TP) into programs are discussed, comparative features of a number of algorithms for translating decision tables are given and some recommendations are made for selecting a translation algorithm. There is 1 table, 4 figures and a bibliography with 24 titles.

UDC 681.3.06

PROBLEM OF CERTIFICATION OF COMPUTER PROGRAMS

[Abstract of article by Arushanyan, O. B., and Borisov, V. M.; received by the editors 18 Jan 1980]

[Text] Basic aspects of certification of programs for solving problems of numerical analysis are discussed. The concept of certification of programs is introduced. The most widespread methods of conducting computer experiments are discussed. Bibliography with 8 titles.

UDC 681.142.2

INFORMATION SUPPORT FOR AN ALGORITHM SYNTHESIS SYSTEM WITH GUARANTEED ACCURACY OF RESULTS

[Abstract of article by Razumovskiy, S. N.; received by editors 2 July 1979]

[Text] Fundamentals of constructing information support for a system of synthesis of algorithms for problems of computational class with guaranteed accuracy of results are given. Features of value formation processes are defined, and the system data base structure is outlined. Bibliography with 9 titles.

UDC 681.3.06:51

SOME MODIFICATIONS TO THE KNUTH ALGORITHM FOR CHECKING CYCLICITY OF ATTRIBUTE GRAMMARS

[Abstract of article by Chebotar', K. S.; received by the editors on 23 Oct 1979]

[Text] Some modifications are suggested for the Knuth algorithm for checking the cyclicity of attribute grammars that make it possible to save storage needed for algorithm operation and to speed up its convergence. There are 2 figures, 4 tables and a bibliography with 3 titles.

UDC 518.5

ANALYSIS OF SOME METHODS OF COMPARING TRANSLATOR QUALITY

[Abstract of article by Gorelik, A. M.; received by the editors on 21 Sep 1979]

[Text] Criteria by which various translators may be compared are discussed, and various methods of comparing the efficiency of generated programs are analyzed. Bibliography with 8 titles.

UDC 681.3

ARCHITECTURE OF A METABASE OF DATA FOR A RELATIONAL DBMS

[Abstract of article by Kilov, Kh. I., and Popova, I. A.; received by the editors on 25 Jul 1979]

[Text] An architecture is suggested for a metabase of data for a relational SUBD [DBMS=data base management system] realized in the form of a system of text tables constructed uniformly and ensuring semantic control of text values. A schematic example of organization of the data metabase tables is shown. There is 1 figure and a bibliography with 10 titles.

UDC 681.3.06:514.14

PACKAGE OF PROGRAMS FOR DESIGN OF ONE CLASS OF COMPLEX OBJECTS

[Abstract of article by Trumen', A. Ye.; received by the editors on 21 Jan 1980]

[Text] Discussed in the article is a package of programs for constructing three-dimensional objects in the interactive mode with display of results of the design on graphic devices (displays, plotters). One figure and a bibliography of 8 titles.

8545
CSO: 1863/148

UDC 681.3.06.

LANGUAGES FOR COMPUTER GRAPHICS

Kishinev YAZYKI MASHINOY GRAFIKI in Russian 1980 (signed to press 4 Mar 80) pp 2-8, 75-77, 106-107, 122-123, 152, 176-177, 190, 213-217

[Annotation, introduction and chapter excerpts from book "Languages for Computer Graphics", by Dmitriy Nikolayevich Todoroy, Lyudmila Ivanovna Romanchuk and Sergey Makarovich Peretyatkov, Izdatel'stvo "Kartya Moldovenyaske", 1200 copies, 252 pages]

[Text] Computer graphics is a relatively new, but promising field of application for electronic computers and mathematical languages in the country's national economy.

In this handbook, computer graphic languages are described and presented by their pragmatics, syntax and semantics; problems of representation, input, storage, processing and output of geometric data are examined; the field of application of computer graphics and ways of using languages in solving its various problems are pointed out.

This handbook is intended for computer users, students and post-graduates of the applied specialties of VUZ's and VTUZ's, and specialists in scientific research institutes.

In the process of representation, conversion, input, output and analysis of geometric data, in addition to traditional computer devices, computer graphic hardware (terminals) are used: input and output devices and on-line input-output graphic terminals. Transmission of geometric data between various users and representation, conversion, analysis and documentation of this data is done by computer software and graphic terminals. Such systems are based on a common language base--languages to describe, represent, analyze and convert geometric data.

Problems associated with representation of a studied or derived object in graphic form, in the form of graphs, drawings, schematics, etc., form a most interesting sphere of computer applications. Solving problems of this type (they make up more than 85 percent of all the problems solved by computers) has brought about the development of a special class of geometric programming languages called languages for computer graphics and has led to the development of computer software systems (computer graphic systems) based on these languages.

The first computer graphic systems permitting programming processing and output of results in languages of graphic terminals of the ATsPU [alphanumeric printer] type,

teletype and plotter by developing the appropriate programs and using them in systems for automation of programming and designing were the result of the efforts by collectives of associates led by A. P. Yershov and S. S. Lavrov, corresponding members of the USSR Aacademy of Sciences, and V. L. Rvachev and Ye. L. Yushchenko, corresponding members of the UkSSR Academy of Sciences. Subsequently, the emergence of languages for input, representation, analysis, conversion and output of geometric data led to the development on their basis of powerful computer graphic software systems.

In the process of automated projecting and designing, as well as in solving other problems of science and technology, the need arises for automatic representation, processing and reproduction of graphic data. In the process, two independent problems have to be solved: development of devices for display and representation of geometric data and development of software for these devices.

Display device software includes the aggregate of methods, algorithms and programs that provide automatic conversion of the description of a geometric object into its geometric image.

The problem of converting a three-dimensional image to a machine drawing includes development of complex algorithms by which the optimal number of planar images is selected and the detail three-dimensional image is broken down into projections and auxiliary forms.

The problems of machine coding of an object drawing and converting the drawing data into a program for drawing have now been worked out. Methods and algorithms have been investigated and program systems to solve the problems of forming and outputting graphic data on display devices (input, output and input-output) have been developed.

Geometric data input devices are scanners that examine a geometric image and input, in the majority of cases, a "matrix of intersectability" into computer storage. Geometric output devices are graph plotters, photo plotters and others that make it possible to put graphs, drawings, schematics and others on paper with very high accuracy (up to 0.05 mm). These output devices are used to output multicolor diagrams, networks, graphs of functional dependencies, seismograms, machine-building and construction drawings, synoptic and meteorological charts, functional and circuit diagrams, various types of charts, and alphanumeric, drawing and graphic data.

Subsequently, the possibility emerged of efficiently solving problems by using on-line input-output devices for geometric data (teletypes, displays, etc.).

These computer graphic devices, languages and systems have led to the capability of solving the problems of introduction of standardization of geometric data tasks with conversion from one coordinate system to another, and transmission and input of geometric data to computer storage. This has made it possible to efficiently perform debugging operations while designing and projecting and to generate geometric images during experimental and theoretical investigations. The capability has emerged for language and image intercourse between designers and developers, and for establishment and utlization of libraries of programs for input, processing and output of geometric data.

The theoretical bases for building computer software systems oriented to solving geomwtric problems have been established by the works of Soviet scientists. One of the most complicated problems that occur in solving geometric problems by computer is the development of methods to record geometric data in digital form. In connection with this, the need arose to create a language to describe this data. A number of requirements are imposed on such languages:

--no ambiguity (each record must allow unequivocal interpretation);
--convenience of formalization of the various actions associated with data storage and conversion;
--universality in describing data;
--flexibility so changes and additions can be made without detriment to the general technique; and
--singleness of purpose, i.e. no redundancy, etc.

A large number of languages have been developed in recent years that to a certain extent meet the requirements of convenience of description, analysis, conversion, input, processing and output of geometric data using computers.

The following computer graphic languages are the most widespread and used in the USSR: GRAFOR (developed by Yu. M. Bayakovskiy, candidate of physicomathematical science), APPARAT (developed by a group of associates headed by S. S. Lavrov, corresponding member of the USSR Academy of Sciences), GEOMAL (authored by I. Ye. Pedanov, candidate of physicomathematical science), GEOMETR (created by a group of authors led by G. K. Goranskiy, doctor of engineering science), GRAFIK (developed by a group of associates led by D. N. Todoroy, candidate of physicomathematical science), SIRIUS (authored by V. F. Sokolov, candidate of engineering science), OGRA (developed by D. M. Zazulevich, candidate of engineering science), and IRIS (created by a group of developers led by V. L. Katkov, candidate of physicomathematical science). This handbook deals with just these languages.

The authors express their thanks to the associates at the laboratory for SMO VK and ASU [software systems for computer complexes and automated management systems] at the department of algorithmic languages and programming at Kishinev State University for their help and support, and to M. G. Gonets, V. V. Goncharenko, A. N. Popeska and V. K. Stroya for their valuable comments and suggestions during the work on this handbook.

Chapter 1. The GRAFOR Language

1. Pragmatics

GRAFOR is an expandable program system providing a simple and convenient means for data output in graphic form. The system is machine independent and may be used on the various machines that support a FORTRAN compiler.

The aggregate of calls to these programs makes up the graphic language enabling description of an image that is ultimately recorded on paper. In translating programs written in languages such as ALGOL and FORTRAN, the compiler detects all syntactic and many other semantic errors and many provide a listing of diagnostic messages. GRAFOR lacks this capability since the language statements are interpreted during execution of the program.

General organization subroutines permit selection of the measurement unit, and page definition, identification and closure. Measurement units are selected prior to page definition and may not be changed within it. If a page has not been defined, then any calls to GRAFOR are ignored. After completion of output to the current page, it must be closed.

The dimensions of the working field on the paper are assigned during page definition. Certain dimensions exist (10x10 cm minimum and 26x60 cm maximum) that are established by default if any dimension is incorrectly assigned. Page borders may be outlined. Even if they are not outlined, they do exist invisibly and do not permit the pen to go outside the page limits. This is especially important when working with programs not yet debugged, since localization of the pen within the working field makes it possible to keep from spoiling a page prepared earlier.

The set of programs for constructing the main graphic elements is the base for creating applications software. Other possible uses include: construction of histograms, drawing of smooth (French) curves through a series of points, construction of isometric and perspective projections, etc. The main graphic elements include: straight lines, rectangles, polygons, arcs, circles and ellipses, grids, markers, alphanumeric and other graphic symbols. Straight lines are the most important element. A step graph plotter from any given point can jump to one of eight adjacent points.

The PLOT program makes it possible to move the pen to another point. In the process, the pen will draw a line on the paper as it moves if it is down.

There are two programs, BOX and RECT, to construct rectangles.

The POLYG subroutine makes it possible to construct a regular convex polygon or a regular n-point star.

A circle or an arc of a circle can be constructed using the CIRCLE subroutine. The circle approximates a polygon with a side equal to about 1 mm.

An ellipse or arc of an ellipse is constructed by the ELIPS subroutine.

A grid is constructed by the GRID subroutine.

A graph plotter is often used to portray various functional dependencies in the form of graphs. The problems arising in construction of graphs and consequently, the facilities available for this in GRAFOR may be divided into three groups:

1. Definition of mathematical and physical space. Used for this are the subroutines REGION, LIMITS and others.
2. Construction of polygonal curves.
3. Construction of the axes AXIS, XAXIS, YAXIS and others.

Axes are drawn through the zero value (if it exists) or through the main division closest to the zero value. Size of the basic division is specified during the call or is selected automatically.

Chapter 2. The GEOMAL Language

1. Pragmatics

GEOMAL is a high-level programming language. The field of its definition is the description of computational and geometric processes. The language has a block structure.

GEOMAL is an expansion of ALGOL-60 through the introduction of new types of values and expressions: vector and geometric.

The language may be used to represent computational and geometric objects of the following types: integer, real, Boolean, index, array, switch, procedure, point, straight line, plane, vector and the general geometric types (subtypes): point, line, surface and body.

Objects of the type integer, real, Boolean and index are some integer or real number, or some Boolean value: TRUE or FALSE, or some label—index of a specific position in the program written in GEOMAL. Objects of the type array are sequences of objects of the same type: integer, real or Boolean. Objects of the type switch are sequences of objects of the type index.

Objects of the type point, straight line and surface are some point, some straight line or some surface, respectively. They are considered defined if it is possible to compute the parameters completely describing them.

An object of the general geometric type represents an infinite, generally speaking, unconnected and unlimited set of points. Depending on size, it has a subtype: point, line, surface or body. An object of the subtype point may represent both one and several points. Similar statements are valid for the other subtypes too.

An object of the general geometric type is defined only when for an arbitrary point of space there is a possibility of determining whether or not it belongs to this object. An algorithm for generating a response to this question is a matter of specific representation of the language.

Objects in the GEOMAL language are divided into elementary and composite.

Elementary objects may be integer and real numbers, indices, values of the Boolean type or elementary objects with a name; the latter have a name and type assigned during their description, and a value. A name is a sequence of letters and numbers, beginning with a letter.

Composite objects are arithmetic, Boolean, name, vector or geometric expressions. They also have a type and value. An expression is a rule for obtaining a value of the corresponding type. Composite objects (expressions) are sequences consisting of objects, operations and punctuation marks that are compiled according to specific syntactic rules.

Objects of the procedure type are one or more objects of the type: integer, real, Boolean, array, vector, point, straight line, plane and of the subtype: point, line, surface and body. They represent a description of a procedure and call to a procedure and are called a function or procedure. A call to a function may be a

component of an expression and has a unique value of the type: integer, real, Boolean, point, straight line, plane and of the subtype: point, line, surface and body. A call to a procedure is a statement, but the description represents an expression consisting of declarations of a name of a procedure and objects newly introduced in it with possible indication of the type of these names and list of statements and operations that are to be performed on the objects to obtain new objects.

Operations in GEOMAL are represented by operations and statements.

In GEOMAL, there are arithmetic (+, -, X and others), Boolean ($\vee$, $\neg$ and others), and geometric (transfer, rotation and others) operations and relational operations ($<$, $>$, = and others). These operations are used to perform some operations on objects to obtain other objects of the specified types.

Statements in GEOMAL are the same as in ALGOL-60, but their possibilities have been expanded. These are the block and statements of assignment, transfer, repetition and procedure and the dummy, conditional and compound statements.

A program in GEOMAL is represented in the general case by a block. Semantics are given in terms of ordinary conversational language. Syntax is described using the language of (Backus) normal forms (BNF). BNF notation has the form:
$<$name$>$: : = $<$letter$>$ | $<$name$>$ $<$letter$>$ | $<$name$>$ $<$number$>$

The angle symbols $<$ $>$ mean nothing in themselves; they merely enclose a separate concept to avoid confusion when read together and are not GEOMAL symbols. The symbol : : = means "by definition is," and the vertical line | means "or."

The notation shown above is read: $<$name$>$ by definition is: $<$a letter$>$ or $<$a name$>$, after which is written $<$a letter$>$ or $<$a name$>$, after which is written $<$a number$>$.

Chapter 3. The GRAFIK Language

1. Pragmatics

In developing the GRAFIK languages, the authors were guided by the ideas of developing a high-level language intended for practical purposes and oriented to using it in various fields of geometric data applications using a computer together with various devices for input-output of geometric data.

1.1 Alphabet

Consists of the following basic symbols: letters, numbers, delimiters and service words.

Letters are the capital and small letters of the Russian alphabet.

Numbers are from 0 to 9 (Arabic).

Delimiters are symbols of arithmetic operations, parentheses and separators. Delimiters have a fixed meaning that is evident in the majority of cases.

1.2 Numbers and Identifiers

Are defined as in the ALGOL-60 language. Certain identifiers must be assigned to the standard functions of mathematical analysis. The list of assignments: sin, cos, tg, ctg, sh, ch, arcsh, arcch, arcsin, arccos, arctg, abs, sign, sqrt, ln, exp, arcth, entier, lg.

A string is any sequence of basic symbols not containing 'or,' or a dummy sequence.

These service words are incorporated: DUGA [arc], KDUGA [k arc], KONETS [end], DO [until], KRIVAYA [curve], TKRIVAYA [t curve], TOCHKA [point], LOMANAYA [broken line], TLOMANAYA [t broken line], NADPIS' [legend], OKRUZHNOST' [circle], POVOROT [rotation], PODPROGRAMMA [subroutine], PRYAMAYA [straight line], TPRYAMAYA [t straight line], ISKLYUCHIT' [exclude], PERENOS [transfer], SIMMETRIYA [symmetry], FUNKTSIYA [function], GRAFIK [graph], KOMMENTARIY [comment], PERO [pen], PREOBRAZOVANIYE [conversion], TSIKL [iteration], SHAG [step], NACHALO [begin], VYPOLNIT' [do], KONETS TSIKLA [end iteration], KONETS PODPROGRAMMY [end subroutine], VKLYUCHIT' [include], BSP [library of standard subroutines].

Service words are interpreted by the translator as one symbol. The semantics of the service words are explained in the presentation.

The arithmetic expression in the GRAFIK language differs from that in ALGOL-60 in that the symbol $\div$ is excluded from the symbols of operations of the multiplication type, and a variable and function designator are defined the following way: a variable is an identifier; a function designator is a function identifier (aggregate of parameters).

1.3 Statements

To draw figures in the GRAFIK language, a group of statements called fragments is used. The fragments are TOCHKA [point], TPRYAMAYA [t straight line], PRYAMAYA [straight line], TKRIVAYA [t curve], KRIVAYA [curve], DUGA [arc], KDUGA [k arc], LOMANAYA [broken line], TLOMANAYA [t broken line], OKRUZHNOST' [circle] and FUNKTSIYA [function].

The following group of statements is used to convert an image specified by a sequence of fragments. The statements are: PERENOS [transfer], POVOROT [rotation], SIMMETRIYA [symmetry] and PREOBRAZOVANIYE [conversion].

In addition, the dummy statement and the statement to assign values to variables are used.

1.4 Control Statements

Control statements are: transfer, subroutine call, VKLYUCHIT' [include], ISKLYUCHIT' [exclude], iteration and conditional.

Chapter 4. The GEOMETR Language

1. Pragmatics

This language is used to describe geometric data that should be considered source data for a procedure in ALGOL-60.

1.1. Data

Any fairly complicated machine-building part is considered as consisting of simpler component objects that are divided into standard, type, elementary and derivative.

Standard objects include design elements, the form and size of which are regulated by GOST [state standards] or other standards. For example, a blind hole obtained by drilling, a keyslot, a section of a shaft with splines, a threaded hole, a groove for outlet of thread, etc.

Type objects include the parts often found in the class at hand to combine surfaces and standard objects. For example, bosses of a certain form and planar areas.

Elementary objects include a point, straight line, circle, plane and circular cylinder. Elementary objects are considered oriented in space.

The side facing the body of a part is called the positive side of the surface that coincides partially or completely with the surface of the part. Positive and negative sides are distinguished for a line on a surface. A positive side of a line on a surface that does not coincide with the surface of a part body is selected arbitrarily.

Derivative objects are obtained as a result of a union, intersection and other operations performed on the objects described above.

In the input language, the three-dimensional image of the part, and not its projection on the plane of the drawing, is described.

Projections, profiles and cross sections indicated on a drawing in describing a part do not play a role in the inpur language.

To describe the geometry of a machine-building part, it is necessary to specify data on all component geometric objects, the relative position of the part's component objects and data of a theoretical-set nature.

1.2 Statements

A part description in the input language is a sequence of description statements separated by the symbol #. Each description statement contains one of the following types of data: description of a component geometric object of the part, indicator of deviation of the form of the surface, indicator of deviation of the relative position of the surfaces, indicator of material, indicator of hardness and certain other types of data.

The description of the component geometric object is the main part of the description in the input language. It usually begins with a label coinciding with the element number. A label is segregated by a colon. For planar details, all elements are numbered in such a way that in tracing the boundary, the body stays on the left, and for three-dimensional details, the component objects are numbered arbitrarily.

In describing a detail in the input language, the sequence of the description of the component objects is significant. Object description order must correspond strictly to a possible sequence of object construction which is defined by the nature of dimensioning on the drawing. In describing a detail as a base, only objects described earlier may be used. The description of other detail objects does not affect the position of those described earlier.

Chapter 5. The APPARAT Language

1. Pragmatics

The formalized geometric language APPARAT is based on ALGOL-60 and contains additional apparatus to describe geometric objects, their relative connections and quantitative characteristics. To this end, new types of values, types of operations, expressions and additional rules for notation of statements have been introduced into the language.

Statement types are the same as in ALGOL-60.

To depict new concepts, the set of basic language symbols was expanded.

The language syntax is described using the same apparatus of metalinguistic formulas as in ALGOL-60

Chapter 6. The IRIS Language

1. Pragmatics

The IRIS system is designed to output graphic data from the BESM-6 computer to peripherals--a plotter or television screen. The system language is based on the GPDL language from which the concept of a drawing marker and certain basic statements have been taken.

The system makes it possible to:
--join "computer" and "graphic" programs into a single complex (possible combinations are BEMSh-IRIS and ALGOL-60-IRIS);
--obtain "graphic" programs of any length and place them in the system file; and to
--include in the system and use as library procedures those written in the IRIS and BEMSh languages.

In the IRIS system, the process of reproduction has a static nature: First the user creates a program in which the drawing marker describes its three-dimensional drawing, then the program is translated and the machine draws the given object. The language has facilities that make it possible to consider the object from different points of view and to thereby introduce some dynamics into the drawing.

Central to the language is the concept of the drawing marker (or simply marker) in terms of which are explained the actions being prescribed by various designs. The marker is a pencil linked rigidly to a moveable system of coordinates (fig. 18). The drawing element (a pen) is positioned at the origin of the coordinates, and the Z axis with respect to the marker axis.

Shifts of the marker in reproducing the individual elements of the design are relative: each time it is instructed to what value and in which direction it is to be moved from the current position. In addition to shifts, the marker can be rotated relative to any axis and the drawing scale can be changed. Thus, the "position" of the drawing marker relative to some absolute system of coordinates is described by seven values: three angles of rotation, three coordinates and the scale.

The marker may be moved with the pen up or down and the pen color may be changed.

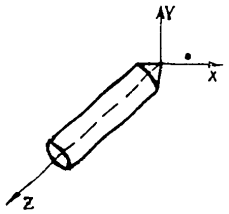


Fig. 18. Position of the drawing marker

Chapter 7. The SIRIUS Language (system for computing data that controls machine tools)

1. Pragmatics

Automation of programming for machining parts on equipment with numeric control and the computations made in designing complicated machines and mechanisms involve coding various geometric objects to input the corresponding data into a computer. It is most convenient to code the data in a language close to engineering. The SIRIUS language is oriented to solving geometric and concrete engineering problems. In connection with this, standard titles have been assigned to certain titles of data.

The SIRIUS language is based on the algorithmic languages of ALGOL-60 and COBOL and is described using (Backus) normal forms.

The language alphabet includes letters, numbers and special symbols.

<a letter> : : = [in Cyrillic] А | Б | В | Г | Д | Е | Ё | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | [in English] D | F | G | H | I | J | L | N | Q | R | S | U | V | W | Y | Z

<a number> : : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< a special symbol >: : = + | — | ¬ | ∨ | ∧ | ↑ | = | / | X | ; | : | , | [|] | : | = | " | ⟶ | ↑_ | _↑ | : ↑ : | SPRAVA [from right] | SLEVA [from left] | SVERKHU [from above] | SNIZU [from below] | BLIZHE [closer] | DAL'SHE [further] | VNUTRI [inside] | VNE [outside] | VPERED [forward] | NAZAD [backward] | VID [view] | OBKHOD [tracing] | VPRAVO [to the right] | VLEVO [to the left] | PA | PE | PEREM | METKA [label] | PEREYTI K [go to] | TO [then] | INACHE [else] | YESLI [if] | VYPOLNIT' [do]

Chapter 8. The OGRA Language

1. Pragmatics

The OGRA graphic language is intended for describing the elements of a drawing and the operations to form them in the process of automated projecting. This type of language will receive the greatest use in describing type graphic images forming in the aggregate files of graphic data of an automated system for projecting. The type graphic image included in a file may have a fixed configuration or be changed within broad limits with specified variations of positional, metric and logic parameters. The language forms the basis of a software system for graphic data display devices under development at the BSSR Academy of Science Institute of Engineering Cybernetics.

The OGRA language has two dialects. The input--OGRA-1--is the working tool of the programmer, and the internal--OGRA-2--is used in the process of automatic processing of graphic data. The description of the information in the internal dialect is formed by a special program (translator) for the data prepared using the facilities of the input dialect. The labor-intensiveness of the development of the translator and its adaptation to the changing conditions of the functioning of an automated system for projecting is largely determined by the the composition, degree of formalization and the dynamics of the input language grammar. In developing the grammar of a graphic input language, in the opinion of the authors, the following conditions must be met:

-- the language must include some base set of expressive facilities to describe graphic data, that is expanded with time;
-- field changes to the translator as the language expands must be minimal;
-- starting from the base set, the user compiles a working set of facilities for himself based on the concrete conditions of operation;
-- in conformity with the adopted working set, the user is given the capability of changing the composition and structure of the translator and other components of the software system with minimal outlays;
-- language terminology must be close to customary engineering terminology to facilitate the process of programming graphic data and checking the descriptions compiled;
-- language grammar must be formalized, yet sufficiently simple and convenient for practical use; and
-- the grammar must provide for inclusion of the language and realization of its programs in unified systems of translation.

A formal grammar meeting these conditions has been developed for the OGRA-1 graphic language. The formal grammar is represented by the ordered quadruplet G = (Vn, Vt, F, V) where Vn and Vt are nonintersecting sets of nonterminal and terminal (respectively) symbols of this grammar; F is the set of syntactic rules for the grammar; and V is the initial nonterminal symbol.

In the elaborated version of the OGRA-1 language, a concrete set of graphic objects has been evolved: point; straight line; segment; quadratic curve, including a circle; arc of a curve; French curve, prescribed by a point basis; contours compiled from these objects; shaded areas; alphanumeric and special symbols; type images as well as any combination of these graphic objects.

A graphic operation is some fixed aggregate of operations transforming a given graphic object into another or forming new graphic objects according to data on a given object. The result of a graphic operation is always a graphic object.

A service operation is some fixed aggregate of operations needed to organize the processes of input, storage, accumulation and output of data on a graphic object.

The statement is the basic element of the OGRA-1 language that describes a graphic or service operation. The set of language statements is divided into a subset of graphic and service statements. Each statement included in the graphic module has a structure of the type:
< a statement >: : = < label > < statement name > < statement information portion >

The statement information portion is formed using rules of the form:
xAy: : = xαy,
Bu: : = zαy,
where x, y, z, u and α are description strings; and A and B are nonterminal symbols.

A graphic module is any (dummy, in particular) sequence of OGRA-1 language statements that begins with the service statement NACHALO CHERTEZHA [start drawing] and ends with the service statement KONETS CHERTEZHA [end drawing]. There are two types of graphic modules: those with and those without parameters. Let us call a graphic module without parameters a graphic constant. A graphic module is considered regular is the Pi statement uses the results of the operations of only those Pj statements for which j < i where i and j are the numbers of the tracing of the statements in the graphic module.

A graphic module is the initial nonterminal symbol of the OGRA-1 language.

The set of terminal symbols of the grammar is formed from four subsets:

$$U_1 = \{A\} \cup \{\text{const}\} \cup \{sl\} \cup \{ID\}.$$

$\{A\}$ is the set of simple symbols of the alphabet and includes letters, numbers and special symbols.

(Backus) normal forms are used to describe the grammar:
< a letter >: : = [in Cyrillic letters] А | Б | В | Г . . . [in English letters] V | W | Z
< a number >: : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
< a special symbol >: : = + | - | / | , | (|) | ⌴ | $_{10}$ | ↑ | · | Х | = | ; | [|] | * | ‘ | ’ | ≠ | < | > | :
< a service word >: : = < a statement name > | < line type code > | < relational sign > | < separator > | NA | KONTS | KAS | L | P | V | N | PAR | PERP | GOR |

VERT | UG | POCH | PRCH | R | P | X | Y | > | < | NACH
<a statement name> : : = NCH | KCH | ZAGL | KOM | VYVOD | SK | KS | PERO | TOCHKA | PRYAM | OKR | OTR | DUGA | PER | SKR | PAR | GIL | EL | KR | LOM | KONT | TEKST | SHTR | RAZM | SIM | PP | VKL | ISKL | IZ | TI | POVT | GRUP
<a line type code> : : = STL | STN | SHTN | SHPTL | SHPTN | SHTL
<a relational sign> : : = ≠ | = | > | < | > = | < =
<a separator> : : = : | ; | ,

The following designations were introduced above:

{const} is the set of integers and real numbers. Rules for formation of numbers are the same as in the general-purpose algorithmic language ALGOL-60;

{sl} is the set of service words which are simple or service symbols. They include statement names, line type codes, object position indicators, dimension positions, movement directions, tangencies, concentricities, incidences, orientations, radii, diameters, angular size and operations on text symbols;

S is solid; SH is broken; SHP is dash-dot; TL is heavy; and TN is fine;

NA is indication of incidence (object A is located on object B; for example, a point lies on a straight line);

KONTS is indication of concentricity of objects;

KAS is indication of tangency of objects;

L, P, V, N are indicators of object position in relation to another object (left, right, above, below), dimension position in relation to the object (from within, from without) or the object position in relation to the dimension;

PAR and PERP indicate parallelism or perpendicularity of a straight line relative to another straight line;

GOR and VERT indicate a horizontal or vertical position of a straight line;

UG is the identifier for angle size;

POCH (PRCH) indicates arc orientation--clockwise (counterclockwise)--relative to its center;

> and < are arrow codes on dimension lines;

{ID} is the set of identifiers, consisting of letters and numbers.

Identifiers are used to form labels, which are names of objects and statements, and to specify metric and logic parameters.

The set Vn includes in this version the following nonterminal symbols:
<a nonterminal symbol> : : = < graphic module > | < Li string > | < Sj string > | < label > | < statement name > | < identifier > | < line type code according to the YeSKD [unified system of design documentation] > | < concentricity > | < incidence > | < tying in of object > | < direction > | < dimension of circle > | < object position > | < curvature > | < positional parameters > | < coordinate > | < dimension > | < value > | < dimension characteristic > | < object > | < option > | < orientation > | < angle of inclination > | < coefficients > | < dimension scheme > | < point > | < transfer > | < number of repetitions > | < condition > | < arrow formula > | < dimension list > | < text description > | < image code > | < image characteristic >

8545
CSO: 1863/140

APPLICATIONS

UDC 658.012.011.56.334.7

NEW BOOK ON COMPUTER APPLICATIONS IN LOCAL INDUSTRY

Moscow VYCHISLITEL'NAYA TEKHNIKA V UPRAVLENII MESTNOY PROMYSHLENNOST'YU in Russian 1980 (signed to press 2 Jul 80) pp 2, 175

[Annotation and table of contents of book "Computer Technology in Control of Local Industry" by Gennadiy Arkad'yevich Kremnev and Vladimir Pavlovich Gridnev, Izdatel'stvo "Legkaya industriya", 2,000 copies, 176 pages]

[Excerpts] Annotation

This book presents concrete features of the use of computer technology in the local industry of the Union republics.

Special attention is devoted to setting up automated systems at the highest and middle levels of management. Intersystem complexes of problems and model programs for solving them are reviewed.

The book is intended for engineering-technical personnel who are setting up production control systems and for the managers of enterprise services, production and industrial associations, and administrations and ministries of local industry.

Table of Contents

11,176
CSO: 1863/114

UDC 658.012.011.56.334.7

USE OF KEYBOARD COMPUTERS IN LOCAL INDUSTRY REVIEWED

Moscow VYCHISLITEL'NAYA TEKHNIKA V UPRAVLENII MESTNOY PROMYSHLENNOST'YU in Russian 1980 (signed to press 2 Jul 80) pp 162-163

[Excerpt of chapter 4 of book "Computer Technology in Control of Local Industry" by Gennadiy Arkad'yevich Kremnev and Vladimir Pavlovich Gridnev, Izdatel'stvo "Legkaya industriya", 2,000 copies, 176 pages]

[Excerpt]

Chapter 4. Machine Accounting Stations and Bureaus in Local Industry

One of the principal directions of further refinement of primary accounting at enterprises is mechanization. This is accomplished by centralizing accounting, using contemporary computer equipment, and incorporating advanced forms and methods of accounting.

Keyboard computers, including semiautomatic and automatic machines, have become most widespread at local industry enterprises for the mechanization of primary accounting.

The number of Iskra electronic keyboard machines has increased significantly. Eighty percent of the total number of computing machines introduced in operation in 1977 were electronic.

Tabular keyboard machines, including the Askota-170 bookkeeping machine and the Iskra-23 invoice-bookkeeping machine, have become widespread in recent times.

Askota-170/45 machines with TM-20 electronic multiplier attachments can be used to mechanize both accounting and planning calculations.

One TM-20 attachment can serve two machines at the same time.

The Iskra-23 invoice-bookkeeping machines have even greater capabilities.

Punched card computing equipment has been much less widely used to mechanize primary accounting in local industry than keyboard computing machines. Table 24 below represents the growth in supply of keyboard computers to local industry in the RSFSR.

Table 24.

Indicator	1970	1975	1977
Number of Machines	2,214	6,263	10,439
Cost of Machines, rubles	1,027,000	4,630,000	6,430,000
Number of Machines per Enterprise	1	3	6
Number of Machines per 1,000 Employees	4	10	17

Keyboard computers are mostly used by individuals at enterprise management services.

This is chiefly because of the size of local industry enterprises and the small volume of computing work.

However, when computer use is organized in this way the workload is low.

It is more progressive to consolidate computer equipment at machine accounting organizations: bureaus and stations.

In RSFSR local industry the machine accounting bureaus have concentrated roughly five percent of all keyboard computers.

11,176
CSO: 1863/114

DESCRIPTION AND RECOGNITION OF OBJECTS IN ARTIFICIAL INTELLIGENCE SYSTEMS

Moscow OPISANIYE I RASPOZNAVANIYE OB"YEKTOV V SISTEMAKH ISKUSSTVENNOGO INTELLEKTA in Russian 1980 (signed to press 30 Jun 80) pp 2-4, 135-137

[Annotation, foreword and abstracts from book "Description and Recognition of Objects in Artificial Intelligence Systems" edited by V. S. Gurfinkel' and V. S. Fayn, Izdatel'stvo "Nauka", 2300 copies, 137 pages]

[Text] This collection includes articles concerning the following three problems: mathematical modeling of changeability of a number of objects which are of practical interest (the speech process, certain types of images, etc.), the use of mathematical methods in medicine and some aspects of voice control of computers in man-machine systems. The book is intended for specialists in the area of artificial intelligence, pattern recognition and related areas.

Foreword

With time, significant adjustments have been made to the statement of the problem of pattern recognition, in the understanding of its content and its place in modern scientific-technical knowledge.

One of the main manifestations of this development is the increasing realization that the problem of recognition itself is, in a known sense, secondary to another problem: determining and describing the essence of change of a recognized object. In all cases in which the essence of changeability has been thoroughly researched, the use of that research to organize a recognition procedure has by now been studied sufficiently. The description of changeability also has value apart from recognition, since it opens the way for solving another problem in the area of artificial intelligence -- artificial generation of changes in an object (design, speech synthesis, automatic reproduction, etc.). This has brought about the recent appearance of a large number of items on the mathematical modeling of the regularity lying at the base of some changeable phenomenon or object. This trend is also evident in the present book: the articles by Ye. F. Yurkov and V. S. Nagornov, A. S. Omel'chenko and V. S. Fayn, Ye. P. Ponomarev and Yu. N. Prokhorov, V. N. Sorokin, and A. P. Vaynshtok are devoted to finding descriptions of the regularities which characterize changes in objects in varying problems which are of practical importance.

Unfortunately, it is very diffucult to find the essence of change or its connection with externally-observable characteristics of an object; even today it is not always possible to solve this problem. A classical example of this sort are problems of seismic forecasting or zoning. No less important are problems of medical diagnosis, which are also very difficult. However, the urgent requirement for their solution is a powerful stimulus for the application of new efforts. In the present collection, these problems are addressed by the articles from A. M. Alekseyevskaya and V. S. Pereverzev-Orlov, P. Ye. Kunin and V. P. Karp and Uy. B. Fogel'son.

Still another problem of artificial intelligence touched upon in the collection is concerned with the organization of dialog in a man-machine system. One of the methods of satisfying the requirement for maximum convenience and naturalness of human functioning in such a system consists of providing the machine with the capability of understanding voice commands. Research which has been underway in this area for several years is represented by the articles of S. N. Krinov, V. P. Savel'yev, G. I. Tsemel', as well as A. V. Vasil'yev, S. S. Raksheyev and V. M. Chizhkov and S. M. Shevenko.

The originality of the methods proposed and the importance of the themes raised will undoubtedly be of interest to specialists in the area of pattern recognition and prediction.

DESCRIPTION AND RECOGNITION OF OBJECTS IN ARTIFICIAL INTELLIGENCE SYSTEMS

UDC 621.391

USE OF UNIDIMENSIONAL FUNCTIONS TO FIND EMPIRICAL DEPENDENCY

[Abstract of article by Yurkov, Ye. F. and Nagornov, V. S.]

[Text] A method is described for using unidimensional functions to find the dependency between quantities according to experimental data. An algorithm for finding unidimensional functions, and results of experimental testing of a method, are given.
Three illustrations, eight bibliographic references.

UDC 621.391.19

METHOD OF TRANSFORMING PLANE CURVES BASED ON MOVING-POINT METHOD

[Abstract of article by Omel'chenko, A. S., and Fayn, V. S.]

[Text] A new method is developed analytically for transforming plane curves which models local interactions between points of a curve and is based on the theory of differentiable manifolds. The method is applied to solve one of the problems of correcting ice charts.
Three illustrations, six bibliographic references.

UDC 621.391.19

LOCAL ESTIMATION OF INFORMATION CONTENT OF PLANE CURVE

[Abstract of article by Omel'chenko, A. S.]

[Text] One possible approach is examined for estimating the complexity of tabulating a curve for the common special case in which a local as well as a global estimate is needed, where the complexity of the curve (estimated in the traditional manner) varies significantly in different segments. An algorithm is given which realizes a local complexity estimate. The algorithm is used in one of the problems of correcting ice charts.
Three illustrations, three bibliographic references.

UDC 534.42

ADAPTIVE LINEAR FILTERING OF SPEECH SIGNALS

[Abstract of article by Ponomarev, Ye. P. and Prokhorov, Yu. N.]

[Text] Recursive algorithms for adaptive linear filtering of a speech signal from a mix with additive white noise are examined. The proposed algorithm contains a section which estimates the parameters of the speech signal model recursively and a section which separates the signal from the noise. The relative independence of these sections produces a relatively simple signal processing procedure. Recursive parameter estimation algorithms facilitate the construction of estimates of autoregression parameters as well as the parameters of a multistage model. Estimate consistency conditions are presented, as are the results of digital modeling of a real signal.
Three illustrations, eight bibliographic references.

UDC 621.391.192.2

MECHANICS OF MOVEMENT OF THE TONGUE

[Abstract of article by Sorokin, V. N.]

[Text] A description is given of movements of the tongue as a solid body under the influence of movement of its root and rotation of the mandible, and for a model of elastic deformations of the surface of the tongue. Results of approximating X-rays of the vocal tract by eigenfunctions of elastic oscillations are given. The experiments utilized speech material in the form of X-rays of all voiced sounds in the Swedish language vocalized with long durations, and frames of X-ray films of the basic forms of articulation of Russian pronounced rapidly in isolation as well as in syllable context. The experiments made it possible to clarify the form of the boundary conditions for the root of the tongue. The mean approximation error for the front of the tongue was about 6% when five eigenfunctions were used.
Three tables, seven illustrations, thirteen bibliographic references.

UDC 621.391.144

ESTIMATION OF NOISE TOLERANCE OF REJECTOR ANALYSIS OF SPEECH

[Abstract of article by Vaynshtok, A. P.]

[Text] The noise tolerance of rejector analysis of speech is discussed. Results of computational experiments of estimating the parameters of formant oscillations of an artificial three-formant signal distorted by noise are presented.
One table, one illustration, five bibliographic references.

UDC 621.391

THE 'TWO-DOCTOR' PROBLEM IN PATTERN RECOGNITION

[Abstract of article by Alekseyevskaya, M. A. and Pereverzev-Orlov, V. S.]

[Text] Different doctors, describing the same patient using terms from the same dictionary, ascribe different meaning to the words and, consequently, speak different languages. The possibility of reducing formalisms of the type of decision and predictive rules obtained with one doctor to the languages of other doctors is noted.

UDC 621.391.19

RETROSPECTIVE RANDOMIZATION METHOD FOR COMPARING EFFECTIVENESS OF ALTERNATIVE TREATMENT APPROACHES

[Abstract of article by Kunin, P. Ye. and Karp, V. P.]

[Text] One possible approach to comparing the effectiveness of alternative treatments is considered when retrospective data, rather than "blind choice", are used in composing the compared groups.
One table, two bibliographic references.

UDC 621.391

REDUCING TRIAL-AND-ERROR IN CONSTRUCTION OF DISTINCTIVE FEATURES

[Abstract of article by Fogel'son, Yu. B.]

[Text] A trial-and-error algorithm for teaching programs to recognize objects is examined. An iterative process is proposed which finds the combinations of parameters which distinguish well the objects in classes which are to be learned. This process is based on the idea of gradual "pulling" of the required parameters by those which were selected during a preceding iteration. The convenience of using the algorithm for problems of medical diagnosis is shown.
Three bibliographic references.

UDC 621.391.199

SIGNIFICANCE OF CHANGE IN FUNDAMENTAL FREQUENCY FOR AUTOMATIC SPEECH RECOGNITION

[Abstract of article by Krinov, S. N., Savel'yev, V. P. and Tsemel', G. I.]

[Text] The advisability of using the dynamics of the fundamental frequency for segmenting a speech signal, especially vowel -- voiced consonant combinations making up continuous pronunciation, is examined. The temporal position of the maximum of the fundamental frequency and its relative increase as a function of context and speaker were examined.
One table, two illustrations, six bibliographic references.

UDC 621.391.199

GROUPING OF WORDS ACCORDING TO FEATURES OF REFERENCE SOUNDS AND SOUND COMBINATIONS

[Abstract of article by Krinov, S. N.]

[Text] An algorithm for grouping (dividing into groups) any sets of words according to the features of reference sounds and sound combinations is examined. The code (number) of a group is a sequence of digits which encode the types of sound combinations between adjacent vowels of a complete formation. A word recognition algorithm using the grouping algorithm is described.
One illustration, six bibliographic references.

UDC 621.391.199

OBTAINING WORD FEATURES AUTONOMOUSLY

[Abstract of article by Vasil'yev, A. V., Raksheyev, S. S. and Chizhkov, V. M.]

[Text] A system is examined for obtaining an initial description of productions up to 1.6 seconds long. The time sequence of the features is formed by means of a feature separating device and display interfaced with a typewriter. The use of this system made it possible to obtain the statistical material needed in developing the word recognition algorithm without using a computer.
Three illustrations, five bibliographic references.

UDC 621.391.19

MACHINE 'UNDERSTANDING' OF TEXTS IN NATURAL LANGUAGES

[Abstract of article by Shevenko, S. M.]

[Text] The ideas and formalisms used in existing and theoretical systems for understanding natural language are examined. Proposed is a critical approach

to the modern linguistic conceptions of machine "understanding" from the positions of systemic linguistics, which is represented primarily in the work of Soviet scientists.
Forty-four bibliographic references.

6900
CSO: 1863/141

UDC 681.51:61.396

MICROCOMPUTERS FOR INVESTIGATION OF COMMUNICATIONS CHANNELS

Kiev PRIMENENIYE MIKROPROTSESSOROV V LOKAL'NYKH SISTEMAKH in Russian 1980
pp 51-58

KORNEYCHUK, V. I., TARASENKO, V. P. and TOROSHANKO, Ya. I.

[From REFERATIVNYY ZHURNAL: TEKHNICHESKAYA KIBERNETIKA No 2, 1981
Abstract No 2.81.823 by V. I. Korneychuk, V. P. Tarasenko and Ya. I. Toroshanko]

[Text] One of the methods of improving the investigation of communications channels is development of measuring and calculating complexes based on computers for automated testing of communications channels (IVK AIKS) which take into account the specifics of the types of channels and the features of the types of tests. A microcomputer contained in measuring and calculating complexes for automated testing of communications channels performs the following functions: management of investigations, control of the blocks of the complex, measuring information processing, monitoring the efficiency of the blocks of measuring and calculating complexes and self-monitoring. A type K580IK80 multiprocessor comprises the basis of the microcomputer. An interrupt system supports the operation of eight programs. The microcomputer memory consists of a RAM and a ROM in the form of modules with capacity of 4K bytes each. The memory is monitored for oddparity. The microcomputer software provides processing of numbers of variable length displayed in a format with floating decimal. The microcomputers being developed in the measuring and calculating complexes for automated testing of communications channels will be used in investigations of wire and other types of communications channels.

6521
CSO: 1863/134

CONFERENCES, ORGANIZATIONS

CONFERENCE HELD ON PROBLEMS OF DEVELOPING AND USING COMPUTERS WITH HIGH THROUGHPUT

Moscow PROGRAMMIROVANIYE in Russian No 1, Jan-Feb 81 (signed to press 15 Jan 81) p 94

[Report by V. V. Ignatushchenko and E. M. Mamedli on All-Union Scientific and Technical Conference entitled "Problems of Developing and Using Data Processing Computers with High Throughput"]

[Text] The All-Union Scientific and Technical Conference, "Problems of Developing and Using Data Processing Computers with High Throughput," was held in Kishinev on 9 through 11 November 1979. About 500 scientists and specialists from 150 organizations from 30 of the country's cities took part in it. The theses of the section papers were published by the time the conference got underway.[1] In addition, plenary and problem papers were presented at the conference.

Four papers were read at the plenary sessions:

1. In his paper, "High-Throughput Multiprocessor Systems with a Variable Structure," I. V. Prangishvili discussed the principles of constructing and organizing software for multiprocessor computing systems with variable structure of communications and control and decisive fields. Basic features of such systems were analyzed.

2. In his paper, "YeS [Unified System] Series High-Throughput Computer Systems," N. M. Sharunenko discussed the paths of development of the structures, software and element base of the YeS series computer systems.

3. The paper, "Parallel Processing of Programs, Algorithms and Problems," by V. A. Val'kovskiy, V. Ye. Kotov and A. G. Marchuk.

4. And the paper, "Basic Concepts of the Theory of Software Reliability and Advanced Methods for Designing Reliable Complexes of Programs," by V. V. Lipayev.

---

1 "Vsesoyuznoye nauchno-tekhnicheskoye soveshchaniye 'Problemy sozdaniya i ispol'zovaniya vysokoproizvoditel'nykh informatsionno-vychislitel'nykh mashin'. Tezisy dokladov" [All-Union Scientific and Technical Conference, "Problems of Developing and Using Data Processing Computers with High Throughput." Theses of Papers], NTORES [Scientific and Technical Society of Radio Engineering and Telecommunications] imeni A. S. Popov, Moscow, 1979.

There were 136 problem, section and display section reports given at the conference in the following sections: A--Computer System Architecture; S--Systems Programming; Ch--Numerical Methods; N--Reliability and Diagnosing of Computer Resources; B--Data Base Management and Information Retrieval Systems; M--Models for Assessing the Efficiency of Computer Systems; and E--Elements and Devices of Computer Systems.

In the section "Computer System Architecture," in a problem report, I. V. Prangishvili, Ye. V. Babicheva, V. V. Ignatushchenko, L. V. Karavanova, G. M. Popova, E. G. Prokhorova and N. V. Stepanov discussed the principles of building computer systems, the structure of which is oriented to solving optimization problems through orientation of it to realization of a unified method of solving problems of large dimension, based on gaining experience through realizations of relatively simple problems and subsequent utilization of this experience for calculation of more complex problems.

In the section "Systems Programming," in a problem report, E. A. Trakhtengerts showed the connection between the structure of multiprocessor systems, the capability of parallel processing of a computer processor and the efficiency of computations.

In the "Numerical Methods" section, in a problem report, S. Ya. Vilenkin discussed methods of realization of numerical methods in multiprocessor systems with a common flow of instructions and distributed storage.

In the section, "Reliability and Diagnosing of Computer resources," P. P. Parkhomenko gave a problem report on diagnostic support of computer machines and systems. The basic stages of development of failure-resistant computers were outlined in the report.

In the section, "Models for Assessing the Efficiency of Computer Systems," Ya. A. Kogan discussed in his problem report the work done at the USSR Academy of Sciences' Institute for Control Problems on probabilistic modeling and assessment of quality of computer systems.

In the section, "Elements and Devices of Computer Systems," in a problem report, V. A. Mishchenko, V. D. Kozyuminskiy and A. N. Semashko discussed the problems of synthesizing the element base of digital systems for processing information on the basis of multiprocessor machines and building on this same basis new structures of information processing systems.

During the conference, there was a general discussion on the subject of "Parallel Computations and Parallel Computers." In the concluding plenary session, the measures needed to further develop the level of development and use of high-throughput computer systems were outlined.

8545
CSO: 1863/148

UDC 518.001.1

THE SCHOOL OF COMPUTER MATHEMATICS AND CYBERNETICS ON THE WATERSHED OF TWO FIVE-YEAR PLANS

Moscow VESTNIK MOSKOVSKOGO UNIVERSITETA, SERIYA 15: VYCHISLITEL'NAYA MATEMATIKA I KIBERNETIKA in Russian No 1, Jan-Mar 81 pp 5-13

[Article by A. N. Tikhonov]

[Excerpts] Specialist training is the most important task of the school of computer mathematics and cybernetics, which was established at Moscow University in 1970.

Here are few figures characterizing the school's development in the last 10 years. During the Ninth Five-Year Plan 425 highly skilled specialists in applied mathematics graduated from the school, while 1,270--that is, three times more--graduated in the 10th Five-Year Plan. The school's capacity has now been increased to 350 new students per year, and the total number of students exceeds 1,600 persons. This in fact determines the number of specialists we will train for the national economy in the 11th Five-Year Plan. The school now holds third place at Moscow University in the number of students admitted annually and the total number of students.

A strong pedagogical collective capable of successfully completing the tasks facing it evolved within the school in the 10 years. We have many prominent scientists and experienced teachers: four academicians and three corresponding members of the USSR Academy of Sciences, and twenty professors. Among them, there are three Heroes of Socialist Labor, seven Lenin Prize recipients, fourteen State Prize recipients, and recipients of Moscow University's Lomonosov Prize and the Lenin Komsomol Prize. Talented young graduates of the school and its graduate students help extensively in the educational effort. They represent more than half of our instructor staff (57 out of 107 persons). Considering the swift growth in the number of students, "self-help" was the only way the school could get its job done. We could not count on bringing in a large collective of ready-made specialists "from the outside" to help in the teaching effort.

Presently the school contains nine departments: computer mathematics, general mathematics, mathematical statistics, optimum control, operations analysis, mathematical cybernetics, algorithmic languages, systems programming, and computer complex automation. One of the school's important subdivisions is the Computer Scientific Research Center (NIVTs), outfitted with modern computers and enjoying the status of a scientific research institute. It recently celebrated its 25th anniversary.

The training process is inseparably associated with scientific research in the departments and the NIVTs. In addition to instructors and scientists, the active participants of this research include graduates and senior students. The school completed an extensive complex of scientific research on important problems of applied mathematics in the 10th Five-Year Plan. The work proceeded in five basic directions, in accordance with the USSR Academy of Sciences' long-range plan "Basic Directions for Development of the Natural and Social Sciences in 1976-1980":

1. Mathematical Physics and numerical methods in applied mathematics.

2. Probability theory and mathematical statistics.

3. Systems programming and computer software.

4. Discrete mathematics and mathematical cybernetics.

5. Theory of ordinary differential equations and equations in partial derivatives.

Within the framework of these directions, the school worked on 46 subjects, of which 12 were assigned by the State Committee for Science and Technology of the USSR Council of Ministers, three were included in the national economic plans of different ministries and departments, and eight subjects were in the USSR Academy of Sciences' coordination plan. Moreover the NIVTs worked on six subjects jointly with foreign scientific institutions.

The spectrum of scientific research being conducted at the school is very broad. It completed a number of projects aimed at building and analyzing mathematical models of various physical processes: Research was conducted on mathematical models for plasma physics in connection with the control of thermonuclear synthesis, electrodynamics in application to radio wave emission and propagation, nonlinear optics, analysis of processes in periodic media, geophysics with the objective of developing electromagnetic methods for finding minerals, and gas dynamics with the purpose of analyzing gas flow in nozzles and streamlining of solid objects.

Complex mathematical models of various phenomena and processes cannot be analyzed without creating effective numerical methods. The school and the NIVTs are conducting research on the theory of numerical methods, and an effort is being made to create problem-oriented program libraries. The finite differences method is the most effective one for solving a broad class of problems in mathematical physics. This is why the efforts of Academician A. A. Samarskiy to create a general theory of difference circuits is so significant (1). A cycle of research was completed in recent years with the goal of building difference circuits approximating various problems associated with elliptical and parabolic equations, the alternate-triangle method was developed significantly, difference circuits used to find rough solutions to differential equations have been studied, and effective methods for solving finite-difference equations were developed (2, 3).

One pressing problem of modern science is automation of observation processing and creation of the appropriate automated computer systems. The theoretical foundation for building such systems is the method of rectifying improper problems, which plays a fundamental role in many subdivisions of computer mathematics (4). This made it possible to create, on the basis of the unified methodology of a rectifying algorithm, effective numerical methods for processing and interpreting the results of

experiments in the natural sciences. Application of rectifying methods to problems of this class is fundamental: This permits us to examine more-sophisticated mathematical interpretation models, derive objective assessments of sought physical characteristics, and collect a maximum amount of guaranteed information on the phenomena under analysis. The school has created methods for interpreting the data of photonuclear experiments, for Mossbauer spectroscopy, for roentgenographic and electronographic analysis, for electromagnetic probing, and so on.

Important research is being conducted on operations, game theory, control optimization, mathematical statistics and queuing theory, systems programming, creation of information systems, and so on. It is impossible to describe all research conducted by the school during the 10th Five-Year Plan in a single article; therefore we will dwell in detail on, for example, a few applied projects aimed at solving important national economic problems. They provide an impression of the nature and level of research being conducted by the school.

1. Mathematical problems associated with the planning of emitting systems. The problem of creating emitting systems which excite an electromagnetic field with a polar diagram of special form, and which possess high energy parameters, has become especially important today. Development of sound methods for solving improper problems (4), the methods of mathematical antenna system modeling, and optimum planning theory have played an important role in this problem's solution.

One important aspect of the projects examined here is that the problem of computing the distribution of the electromagnetic field at the aperture of an emitting system was solved with a consideration for the mutual influence exerted by the emitted field and the fields present within the power supply input lines of the emitting system. This problem was solved by means of direct numerical methods presently available for solving excitation problems.

Development of mathematical models applicable to the planning of emitting systems and computer analysis of such models had priority significance to creation of new complexes of radar, communications, and radio-astronomy apparatus. For this research, Academician A. N. Tikhonov and professors V. I. Dmitriyev, A. S. Il'inskiy, and A. G. Sveshnikov were awarded the State Prize in 1976.

2. Numerical modeling of processes in Tokamak devices. One of the promising directions in the program of controlled thermonuclear synthesis is associated with Tokamak devices. Development of research aimed at modeling a thermonuclear plasma stimulated significant progress in theory and in experimentation. Use of computers to make computations made it possible to study complex mathematical models accounting for the design features of a device and the complex geometry of the magnetic field, to subject the process to detailed quantitative analysis, to assess the role of certain factors in this process, to reveal new properties and laws, and to examine integral processes.

Yu. N. Dnestrovskiy and D. P. Kostomarov were the first in the world to develop a transport model for the balance of particles and energy with the purpose of studying changes in density, temperature, and magnetic field in time and space (8, 9). The model has enjoyed worldwide recognition. It is presently being employed extensively not only in the Soviet Union but also in other countries (with the credit for its

creation being given to its authors). The model includes a combined description of the electronic, ionic, and neutral components with a consideration for their mutual transformations resulting from atomic processes.

Computations made with the transport model were used by the authors to study the results of experiments run with T-3, TM-3, T-4, and T-10 devices, comparing experimental data with theoretical considerations; they also analyzed the contribution made by various processes to a plasma's energy losses. Their analysis permitted them, in particular, to reach a fundamental conclusion as to the "classical" nature of the energy losses experienced by ions in Tokamak devices. Today this conclusion lies at the basis of all predictions of the behavior of a plasma in Tokamak devices of the next generation, presently on the drawing boards. These devices are intended to operate in conditions which cannot be achieved and studied experimentally with the presently existing devices. Information on the behavior of a plasma in such extreme conditions can be provided only by computer experiments. As an example the programs the authors developed were used to design the INTOR device (this is an international project being carried out under the International Atomic Energy Agency's sponsorship on the basis of a proposal by the Soviet Union).

A broad range of problems associated with magnetodynamic (MHD) approximation were examined. A package of programs has been developed to compute the MHD-equilibrium and stability of a toroidal plasma column (9, 10). A number of projects aimed at modeling nonlinear MHD processes in conditions causing development of stripping and overshorting instabilities on magnetic surfaces (9, 11, 12), and a three-dimensional code for analyzing nonlinear interaction of helical modes (9, 11) were completed.

Professors Yu. N. Dnestrovskiy and D. P. Kostomarov and senior instructor A. M. Popov were awarded the Lomonosov Prize, 1st Degree, in 1976 for their research on numerical modeling of processes occurring in Tokamaks.

3. Creation and introduction of aircraft automated planning systems.

In the course of the efforts to create a SAPR LA [automated aircraft planning system], the school's colleagues worked jointly with engineers to solve, for the first time, important practical problems associated with compactly describing an aircraft with a relatively small number of design parameters, they developed quick algorithms for computing the aircraft's basic technical characteristics, they developed optimization algorithms, and they tied individual subsystems of the SAPR together. The algorithms, programs, methods, and handbooks developed in the course of work on the research subject were introduced into planning practice. The system can be used to generate a set of plan variants that meet the selected effectiveness criteria, and to subsequently add the details to the plan. Thus important stages in planning, which had formerly been performed either manually or without systematic use of computer technology, were automated. The economic impact from introducing just one of the SAPR subsystems is about 1 million rubles per year.

For this work, a group of colleagues of Moscow State University's school of computer mathematics and cybernetics was awarded the prize of the USSR Ministry of Higher and Secondary Specialized Education in 1979.

4. The KRAB remote access system. It is difficult to overstate the significance of improving the forms of communication between man and computer. The effort of preparing problems for computer solution may be hastened by broadly introducing dialog systems and remote access systems. This can explain the arisal of numerous interactive software systems oriented toward different categories of users, different technical resources, and different classes of problems.

The school of computer mathematics and cybernetics has developed and introduced a remote access system (18, 19) which has come to be called the subscriber servicing complex (KRAB). In distinction from other systems having similar purpose ("Pul't", "DIMON", etc.), created for the BESM-6 computer, this system accounts for the unique ways computers are used at computer centers servicing a broad variety of users, such as universities and major scientific research institutes. Such use is unique in that the requirements of low resource capacity, higher "dependability" in relation to programming errors made by the general user, and simplicity of assimilating and understanding the mechanism of its operation are imposed upon the system. The KRAB system satisfies these requirements. In order to satisfy them, the system's developers had to conduct an entire cycle of research in order to study the nature of programs fed into the system, and reveal the most frequently encountered communication procedures followed by users when running their problems. It is important to note that their research was based on actual statistics describing general use of computers at Moscow State University; this guaranteed that the estimates at the basis of the system's planning and introduction not only at Moscow State University but also at many other of the country's computer centers using BESM-6 computers would be highly dependable. The KRAB system is presently being used in a number of Moscow's scientific research institutes, the Institute of Cybernetics in Kiev, in Novosibirsk, Tbilisi, and other centers. The economic impact from its introduction exceeded 800,000 rubles in 1980, according to documents submitted to the school. The high degree of protection offered to the system against random perturbations, presence of all basic functions inherent to remote-access systems, the high effectiveness with which the most frequently encountered procedures for editing, for running programs, and for creating personal archives are carried out, and simplicity of assimilation and maintenance predetermined this system's success. Attainment of these properties necessitated considerable efforts in systems programming, and a great deal of meticulous work to improve the operational system and to optimize the programs. The experience of creating and introducing remote-access systems is an important subject of further scientific generalization and improvement, the results of which will directly influence the effectiveness with which computers are used in science and in the national economy.

5. Creation of computer networks for Moscow State University's time-sharing system. The time-sharing system of Moscow University is one of the first branched computer networks of this sort in the Soviet Union. It is being created in order to provide the university's subdivisions direct access to high-power computer technology, and to permit its use in training and in scientific research. The time-sharing system has important significance to development of methods for automating experiments and observation processing. Creation of the time-sharing system is associated with development of the methods and resources of automating training and university administration. Professor I. M. Ternov, first vice-rector of Moscow State University, is the leader of the time-sharing system project, Academician A. N. Tikhonov is the scientific director, and Professor Ye. A. Grebenikov, director of the Computer Scientific Research Center is the chief executive.

The first generation of the time-sharing system was completed in the 10th Five-Year Plan. The new computer network created qualitatively new conditions for using computer technology, it has significantly raised the effectiveness with which apparatus is used, and it has accelerated and simplified the preparation of programs for computations. Work on Moscow State University's time-sharing system will be continued in the 11th Five-Year Plan.

11004
CSO: 1863/131

PUBLICATIONS

NEW TEXTBOOK ON WORK OF COMPUTING INSTALLATIONS

Moscow EKONOMIKA, ORGANIZATSIYA I PLANIROVANIYE VYCHISLITEL'NYKH USTANOVOK in Russian 1980 (signed to press 8 Jul 80) pp 2, 230-231

[Annotation and table of contents of book "The Economics, Organization, and Planning of Computing Installations" by Stepan Petrovich Kutsenko, Boris Vladimirovich Marinchenko, and Yuriy Grigor'yevich Krivonosov, Izdatel'stvo "Statistika", 10,000 copies, 232 pages]

[Excerpts] Annotation

This textbook was written for college students specializing in the organization of mechanized processing of economic data in conformity with the program of study "The Economics and Organization of Computing Installations."

The book concentrates attention on the organization of work of collective-use computing installations and networks of them, the necessary complement of computer hardware, planning, establishing norms, control of production and economic activity, and the like.

This book may also be useful to practical workers engaged in designing and organizing large information-computing systems.

Table of Contents

11,176
CSO: 1863/115

IDENTIFICATION OF MAGNETIC COMPONENTS FOR AUTOMATION AND COMPUTER TECHNOLOGY

Leningrad IDENTIFIKATSIYA MAGNITNYKH ELEMENTOV AVTOMATIKI I VYCHISLITEL'NOY TEKHNIKI in Russian 1980 (signed to press 9 Oct 80) pp 3, 123-125

[Foreword, table of contents and annotation from book "Identification of Magnetic Components for Automation and Computer Technology" by Yevgeniy Pavlovich Balashov, Aleksandr Aleksandrovich Voyevoda, Aleksey Arkad'yevich Smagin and Petr Ivanovich Sosnin, Izdatel'stvo "Energiya", 4700 copies, 128 pages]

[Text] Annotation

This book is devoted to one of the key problems in cybernetics -- identification of the characteristics of nonlinear objects. Using the example of analyzing magnetic elements -- cores with rectangular hysteresis loops -- two basic identification problems are explained: determining the order of the equation describing the magnetic element, and choosing the structure of the element equation. Examined are methods for determining the parameters of models of magnetic elements, and the arsenal of technical means for measuring and monitoring the parameters of magnetic elements. The book is intended for specialists in automation and computer technology and for the developers of devices which make wide use of magnetic and nonlinear elements; it may also be useful for students and aspirants in corresponding specialties.

Table of Contents

Foreword

The developers of modern technical systems are widely supported by information on the functioning of individual system components. This information, which is expressed with the help of defined linguistic forms, results from experiments which have been conducted. Most convenient for formalization are mathematical descriptions which reflect economically the quantitative relationships between current processes and, in physical interpretation, the quality of current processes. The present book examines a set of interconnected problems of defining the characteristics of nonlinear objects. A methodology is given for defining the order and structure of the differential equation which describes the behavior of a nonlinear element in the dynamic mode. The methodology is invariant to the description of the dynamic characteristics of a broad class of nonlinear objects.

Magnetic elements with rectangular hysteresis groups, which are used by the millions as memory elements in modern cybernetic complexes, are used as a specific example of an object of investigation. The difficulties in describing transitional processes in this class of electronic elements are well known due to the significant nonlinearity of their characteristics.

A complex of technical means for measuring and monitoring cores -- and instrumentation system for combined measurements and monitoring of dynamic parameters -- is examined.

The concluding section of the book examines problems of the integral estimation of the operability of data storage and processing systems from a position of information and design reliability and efficiency.

Remarks and comments on the book should be sent to 19041, Leningrad, D-41, Leningradskoye otdeleniye izdatel'stva "Energiya".

6900
CSO: 1863/143

UDC 658.012.011.56:681.3

NEW BOOK ON CONTROL COMPUTER COMPLEXES

Moscow ORGANIZATSIYA UPRAVLYAYUSHCHIKH VYCHISLITEL'NYKH KOMPLEKSOV in Russian 1980 (signed to press 23 May 80) pp 4, 270-271

[Annotation and table of contents of book "The Organization of Control Computer Complexes" by Anatoliy Anatol'yevich Myachev, Izdatel'stvo "Energiya", 13,000 copies, 272 pages]

[Excerpts] Annotation

This book sets forth the fundamentals of organizing control computer complexes based on contemporary small computers. It has a classification of control computer complexes and describes the special features of their use in control systems. The author presents the structure of contemporary small computers, which are used chiefly in control computer complexes. The book reviews the distinctive features of the logical organization of the basic components of these complexes, including devices for communication with outside objects. Recommendations are given on evaluating the technical level of the machinery of a control computer complex. The author devotes special attention to the logical organization of interfaces and interlinking units of various architecture. Means and methods of organizing the combined work of several control computer complexes are reviewed.

This book is intended for specialists in computer technology working in the field of the application of control computer complexes and for students at higher educational institutions.

Table of Contents Page

11,176
CSO: 1863/114

AUTOMATED CONTROL SYSTEMS FOR MACHINE BUILDING ENTERPRISES

Leningrad AVTOMATIZIROVANNYYE SISTEMY UPRAVLENIYA MASHINOSTROITEL'NYM PREDPRIYATIYEM in Russian 1980 (signed to press 6 Jun 80) pp 2, 8-13, 77, 83, 133, 135, 172-175, 260, 279, 282-283

[Annotation, table of contents, and excerpts from book "Automated Control Systems for Machine Building Enterprises", by S. A. Sokolitsyn and V. A. Dubolazov, RSFSR Ministry of Higher and Secondary Special Education, Izdatel'stvo Leningradskogo universiteta, 6,234 copies, 280 pages]

[Excerpts] This textbook examines the concept and classification of automated control systems for enterprises (ASUP's), and the supporting information, hardware, and software. Most of the attention is devoted to development of ASU (automated control system) functional subsystems for the machine building enterprise. It also examines the stages of ASUP planning and introduction, and the economic effectiveness of such systems.

The textbook is intended for students at engineering and economic VUZ's and faculties taking the course "ASUP Planning Fundamentals", and it can also be used by students in technical specialties studying the subject "Automated Control Systems".

Contents

ASUTP's [automated production process control systems] are intended to measure certain values and parameters describing the course of production processes, to monitor and maintain them at a particular level, and to control various units, facilities, and production processes depending on changes in external and internal conditions. As a rule these systems are narrowly specialized, they are limited to a single unit or facility, and they are employed most often in the control of continuous production processes (chemical, metallurgical, and so on). Examples of ASUTP's include the "Avtodispetcher" system, which monitors the work of the entire ammonium production complex of a chemical combine, and the converter division control system of a metallurgical plant's Bessemer converter shop (9).

An ASUTP can make use of both general-purpose and special digital computers, and analog computers. As was noted above, control often proceeds in production process control systems without man's participation, in which case they may be referred to as automatic control systems. Due to their specific features, systems controlling organizational or administrative units cannot be designed as automatic systems yet; they can only be designed as automated systems. Much attention is now being devoted to creating integrated control systems by merging an ASUTP with an enterprise ASU; in such systems, the ASUTP is the primary or lower level.

Only one-third of all ASU's in the USSR are ASUTP's (13). In the Ninth Five-Year Plan, the time for full compensation of capital investments averaged 1-1.5 years in relation to ASUTP's, and 2-3 years in relation to enterprise and association ASU's. This is why special attention is being turned to the development and introduction of ASUTP's in the 10th Five-Year Plan.

The directives of the 24th and 25th CPSU congresses referred to the need for initiating the effort to create the OGAS [State Control System]. Special attention was turned in this case to the fact that we must insure, "...from the very beginning, compliance with the principle of organizational, methodological, and technical unity of this system" ((1), page 298).

The main task of the OGAS is to unify separate ASU's and VTs's [computer centers] into a single state system, and create a single countrywide ASU on this basis. The State Network of Computer Centers (GSVTs) will be the technical basis of the OGAS. Creation of a State Data Transmission System (OGSPD) is intended as a means for insuring prompt information transmission. It will include special cable communication channels and switching centers (Figure 1).

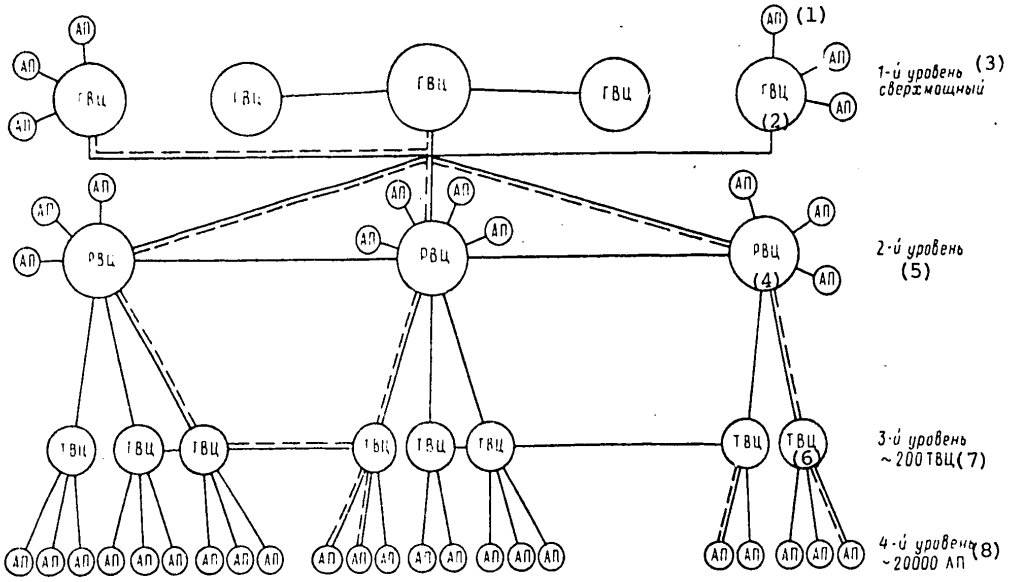


Figure 1. Block Diagram of the State Network of Computer Centers

Key:

1. Subscriber terminal
2. State computer center
3. First level--superpowerful
4. Republic computer center
5. Second level
6. Territorial computer center
7. Third level--about 200 territorial computer centers
8. Fourth level--about 20,000 subscriber terminals

The OGAS is being developed under the guidance of the USSR State Committee for Science and Technology (13, 14). A hierarchical structure corresponding to the existing system of state planning organs and to the sector principle of the national economy's control has been adopted as its organizational basis. The OGAS will consist of four levels.

The first or top level will be represented by the ASU's of state organizations (the USSR Gosplan's automated plan computation system, and the ASU's of Gosstroy, Gossnab, the State Committee for Prices, the State Committee for Labor and Social Problems, the State Statistical Administration, and other committees and union organizations), and OASU's [branch automated control systems], for example of industry, construction, agriculture, transportation, communications, and so on. Seventy-five to eighty automated systems will be created for union ministries and departments, and 15-17 automated systems will be created for state organizations in 1971-1980.

The second level consists of 15 republic automated control systems (RASU's)--equal to the number of union republics.

The third level consists of territorial time-sharing computer centers (TVTs KP). There are plans for creating about 200 such computer centers in the country's centers and major economic regions. Presently there are 123 krays and oblasts and 20 autonomous republics in the USSR. The remaining 57 TVTs KP will be created in cities--in the centers of the largest and economically most-developed rayons. The TVTs KP will be used for control in the corresponding rayons, and as the basis for ASU's operating at enterprises and organizations, in which creation of separate computer centers would be economically unfeasible. Moreover they will provide communication between ASU's and VTs's of the fourth level with those at higher levels.

The fourth and last level will include about 25,000 cluster and individual computer centers to serve as the basis for automated control systems at individual enterprises and organizations.

All four levels of the OGAS will be interlinked by a system of communication channels characterized by different transmission speeds depending on the volume of data transmitted. Subscriber terminals outfitted with terminal units or minicomputers allowing users to communicate directly with the computer will be connected to VTs's at all levels.

5.6. Computing the Optimum Charge Composition: A machine building enterprise possessing casting shops must perform the task of choosing the optimum composition of the charge employed almost every day, with a consideration for the charge materials available at its warehouse. This task can be performed by the methods of linear programming.

Introduction of this method at a certain machine building plant reduced the cost of 1 ton of metal charge for Al-4 alloy from 441.5 to 381.8 rubles.

It is still often optimistically asserted in the literature and in ASUP creation plans that new systems could be introduced in 1.5-2 years. The experience of foreign and Soviet developers indicates that in fact, introduction requires at least 3-5 years (at which time introduction would still be far from complete), and that the system does not become profitable until at least after 5-6 years. The following stages of ASUP creation may be distinguished: from the initiation of planning to installation of equipment (1.5-3 years)--the preparatory period; from

Table 8. Characteristics of Domestically Produced Optical Character-Recognition Equipment

Model	Document Dimensions, mm	Printing Style	Characters Recognized (U--upper case letters, u--lower-case letters, PM--punctuation marks, CS--control symbols, D--digits)	Speed		Recognition Error Probability	Use
				Characters Per Second	Documents Per Minute		
"Blank-2"	(105×105)--(210×300)	OTsR-A, graphical fonts	D, 4CS	--	400	10^{-5}	Computer input, output on punched and magnetic tape
R711	(65×105)--(105×210)	OTsR-A, OTsR-B, stylized fonts	D, U, CS, and stylized cursive digits and control symbols	1,000	500 (max)	--	Computer input
"Ruta-701"	(210×150)--(210×300)	"Optima" typewriter's	D, 4U, and stylized cursive characters	180	20 (max)	10^{-5} (typewriter characters), 10^{-4} (cursive characters)	Computer input, output on punched tape and punched cards
VINITI-2 "ChARS"	(210×300)	"Optima" typewriter's	D, U, PM, CS	50 / 200	20 (max) / 60 (max)	10^{-4}	Computer input, punched tape output
"Sever-3"	(150×180)--(280×355)	Several typographic and typewriter printing styles	D, U, u, PM, CS	35-100	60 (max)	10^{-2}-10^{-4}	Computer input, punched and magnetic tape output

Table 10. Display Characteristics

Technical Characteristics	Models			
	YeS-7064	YeS-7066	YeS-7061	YeS-7063
Screen diagonal, mm	430	430	250	250
Working field, mm	250×250	320×180	150×200	150×200
Character height (dimension), mm	3.5	3.5 2.5	3.6 2.4	3.6 2.4
Number of simultaneously reproduced characters	1,400, 960	240, 480, 960	960, 1,024	960, 1024
Number of functional keys	32	16	18	22
Working (buffer) memory volume, bytes	4,096	4,096	1,024	1,024
Number of symbols per set	94	94	64, 96	64, 96
Maximum data transmission rate, kbytes/sec	500	500	100	100
Light stylus available	Yes	No	No	No
Printer available	No	Yes	No	No

Table 12. Basic Tasks of the Technical, Industrial, and Financial Plan

Section of the Technical, Industrial, and Financial Plan	Raw Data		Task No. and Name	Computation Results
	Raw Data File No.	Task No.		
1	2	3	4	5
Product production and sales plan	I. III, V VI, VII, VIII	--	1. Product production plan 2. Product sales plan 3. Production capacity use plan	Production program Cost of sold product, planned sales volume growth, gross production volume
	I, III, XV	1	3. Production capacity use plan	Section, shop, and plant production capacities. Equipment loading coefficients
Technical development and production effectiveness growth plan	IV, XV	1	4. Computation of savings in current outlays (except wages) due to measures introduced	Savings in materials and electric power, reduction of losses due to waste, depreciation deductions resulting from a decrease in labor requirements, etc. Productive capital required for implementation of the measures
	IV, XV	1	5. Computation of savings in labor and wages due to measures introduced	Savings in wages and reduction of work force in response to measures introduced
	XV	4,5	6. Determination of economic results of the plan for growth in production effectiveness	Measures included in the plan. Annual savings in terms of corrected outlays. Change in total profit, total and estimated profitability, and economic stimulation funds
Labor and wages plan	I, IV, VI	1, 5	7. Computation of the number and wages of the principal piece-workers	Number and wages of principal piece-workers, broken down into occupations and ranks

	III, IX, X, XVIII	5	8. Computation of the number and wages of auxiliary piece-workers	Number of auxiliary piece-workers broken down into occupations and ranks, wages paid in relation to expenditure estimate items
	III, IX, XVI, XVII	5	9. Computation of the number and wages of principal and auxiliary time-workers	Number and wages of principal and auxiliary time-workers, broken down into occupations and ranks, in relation to expenditure items
	XVI, XVII	5	10. Computation of the number and wages of engineers, technicians, accounting and clerical personnel, and junior service personnel	Number and wages of engineers, technicians, accounting and clerical personnel, and junior service personnel, broken down into categories and occupations, in relation to expenditure estimate items
	--	7, 8, 9, 10	11. Computation of the total number of workers and the total wage fund	Total number of workers broken down into categories, occupations, and ranks. Basic and supplementary wage funds, broken down into categories of workers and expenditure estimate items. Total wage fund.
	--	3, 11	12. Computation of summary indicators for the labor and wages plan	Average wages per worker, coefficient of supplementary wages
Material-technical supply plan	II, IV	--	13. Computation of the norms of material outlays per article	Specific and summary norms of material outlays per article, in natural and cost terms
	II, IV, VI, XI, XIII, XVIII	1, 4, 13	14. Computation of the amount of materials required for production and operational needs	Material requirements for the main production program, for spare parts production, and for incomplete production, and for repair and operational needs, in natural and cost terms

Profit, profitability, and product cost plan	III, XI, XII, XIII	4, 11, 14	15. Computation of outlays on equipment maintenance and operation	Expenditures to maintain and operate equipment
	XII, XIV	4, 11	16. Computation of other shop outlays	Other shop outlays
	I, XI	1, 15, 16	17. Computation of estimated rates	Norm of operational outlays per machine-hour, correction coefficients, total number of coefficient-machine-hours. Estimated rates for equipment maintenance and operation, other shop outlays, outlays on special production gear per coefficient-machine-hour, and per article of each name
	III, XI, XII, XIV	4, 5, 11, 14	18. Computation of other estimates of composite outlays	Outlays on auxiliary production shops, outlays on assimilation of the production of new types of products, transportation, procurement, general plant, and nonproductive outlays
	I, II, IV, XIII	4, 5, 13, 17, 18	19. Computation of planned cost	Planned shop, plant, and total article cost
	VII, XIV	3, 19	20. Computation of summary indicators of the product cost plan	Cost of sold products, product sales profit, total and estimated profitability
Financial plan	XIV	6, 11, 20	21. Computation of financial plan's indicators	Material incentive, production development, and sociocultural measures funds

Table 13. Basic Raw Data Used in Computation of the Technical, Industrial, and Financial Plan

No	Data Group Name	Subsystem in Which Formed
I	File of labor standards and summary labor standards per article	NSI
II	Material standards file	NSI
III	Equipment file	NSI
IV	File of article composition and usability of parts and assembly units in articles	NSI
V	Resource limitations	TEP [technical-economic planning]
VI	Data from OPP [operational production planning] subsystem (calendar plan standards, number of parts required for incomplete production, etc.)	OPP
VII	Price list for finished products	NSI
VIII	Product release, unloading, and sales accounting data (finished products remaining at warehouse, and unsold products)	Product marketing and sales control
IX	Service norm file	NSI
X	File of labor standards for jobs in auxiliary shops	Auxiliary production control
XI	Norms of material outlays per jobs and services associated with repair and operation	"
XII	Services of auxiliary and other shops, in relation to outlay items	"
XIII	Norms for expenditure of different forms of energy (electric energy, steam, air, etc.)	NSI
XIV	Cost of fixed capital and the norms of its depreciation deductions	TEP
XV	Data from the technical development and production effectiveness growth plan	TEP
XVI	Personnel schedule	TEP
XVII	Personnel file	Personnel accounting
XVIII	Volume of jobs performed by auxiliary workers	Auxiliary production control

equipment intallation to introduction of the system (3-5 years); from introduction to the time of full compensation of capital investments in the ASUP (2-6 years).

Unfortunately we often encounter cases in which enterprises install computers and other hardware without preparing the appropriate software and information support, with the hope that all of this could be done later on. Neglect of the preparatory stage leads subsequently to operation of the computer below its capacity, which results in large losses and raises doubts as to the feasibility of creating the ASUP.

The following sample values of operational outlay items to be used in summary computations are recommended in the procedure for determining the economic effectiveness of an ASUP: cost of data storage media (punchcards, paper tape, paper, magnetic tape, etc.) and auxiliary materials--1-2 percent of the cost of the hardware; outlays on current and preventive equipment repairs--2.5-5 percent of the cost of the computer; outlays on maintaining the computer rooms--2-2.5 percent of the cost of the buildings; outlays on computer room lighting, heating, security, and janitorial services--0.2-0.5 percent of the cost of the buildings; other outlays--0.25-0.5 percent of the cost of the computer.

11004
CSO: 1863/144

DATA TRANSMISSION AND PROCESSING METHODS

Moscow METODY PEREDACHI I OBRABOTKI INFORMATSII in Russian 1980
(signed to press 2 Sep 80) pp 2, 115

[Annotation and table of contents from book "Data Transmission and Processing Methods", edited by M. S. Pinsker and V. A. Garmash, Izdatel'stvo "Nauka", 2950 copies, 120 pages]

[Text] Annotation

This collection is devoted to statistical and probability methods of solving problems in the theory of data transmission and processing. Estimates of the correcting capacity of concatenated codes are studied, the areas of acceptable data rates for a channel with many users are found, and methods of statistical information processing are investigated.

The book is intended for scientific and engineering/technical workers.

Table of Contents

6900
CSO: 1863/143

UDC 519.95

NEW BOOK CONSIDERS EARLY STAGES OF DESIGNING INDUSTRIAL CONTROL SYSTEMS

Moscow ASU TP. PREDPROYEKTNAYA RAZRABOTKA ALGORITMOV UPRAVLENIYA in Russian 1980 (signed to press 13 Dec 79) pp 2, 294-295

[Annotation and table of contents of book "Automated Control Systems for Industrial Processes. Predesign Development of Control Algorithms" by Vladimir Il'ich Skurikhin, Vladimir Vasil'yevich Dubrovskiy, and Vladimir Borisovich Shifrin, Ukrainian SSR Academy of Sciences, Izdatel'stvo "Naukova dumka", 2,450 copies, 296 pages]

[Excerpts] Annotation

This book considers the methodology of synthesizing control algorithms, testing them, and selecting the best ones in the predesign stage of setting up automated control systems for industrial processes by using a block-type mathematical model of the industrial object of control to simulate the work of the control system under regular and emergency conditions. The authors describe ways to simulate an industrial object of control with continuous production, control algorithms for deterministic and random influences, modules to simulate the hardware of the automated control system, and techniques of optimizing design decisions using the simulation model.

The book is intended for scientific workers and engineers who are developing control systems and for graduate students and advanced undergraduates.

The book has 52 illustrations, seven tables, and 100 bibliographic entries.

Table of Contents

11,176
CSO: 1863/115

- END -