JPRS L/9510

28 January 1981

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 2/81)

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: 2007/02/08: CIA-RDP82-00850R000300070045-2

JPRS L/9510

28 January 1981

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 2/81)

CONTENTS

HARDWARE

UDC 681.3.29./16

CONSTRUCTION CONCEPT AND ENGINEERING-ECONOMIC CHARACTERISTICS OF LARGE-SCALE INTEGRATED MEMORY IN PROSPECTIVE UNIFIED SYSTEM COMPUTERS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 127-131

[Article by Viktor Vladimirovich Przhiyalkovskiy, candidate of technical sciences (Moscow); Andrey Borisovich Akinfiyev, candidate of technical sciences (Moscow); and Nikolay Mikhaylovich Sharunenko, candidate of technical sciences (Moscow); received by editor 22 Jan 80 (after revision, 24 Mar 80)]

[Text] ZU [memories] have largely determined and are determining the efficient use of hardware and software at all stages of development of computer technology. Future models of YeS computers require memories with a large capacity, high speed, high reliability and low cost. These requirements are currently being met with the extensive application of BIS [large-scale integrated circuits (LSI)] of memory.

Semiconductor LSI memories have already been used in models of the second phase of YeS computers as bases for building processor, buffer and microprogram memories, and the process of replacing the ferrite main memories in these models with semiconductors has started recently. The highest indicators for ferrite memories were obtained in the YeS3206 device [1], which has a 1M byte capacity and is mounted with power-supply sources in the standard rack of YeS computers.

Memories for models YeS1022, YeS1033, YeS1035 and YeS1045 were developed in a short time based on microcircuits of semiconductor memory of the 565RU1 with a 4K bit capacity, and the memory for the YeS1060, based on microcircuits of the 565RU3 with a 16K bit capacity. Using semiconductor circuits for memory has made it possible to improve engineering and economic indicators for memory. The cost of one bit of information in the device was lowered three- to four-fold compared to ferrite memories, while dimensions were reduced three- to eight-fold.

Since the ferrite core memory capability of storing data during accidental power supply variations has not been used in the majority of YeS computer applications, the expediency of replacing ferrite with semiconductor memory is obvious. Ferrite memories can be kept for those applications where data has to be preserved when power supplies are turned off. Parameters of main memories of second-phase YeS computers and modernized first-phase models are given in the table.

Evolution of the YeS computer main memories, reflecting the requirement of the computer and the capability of engineering the memory, is shown in Fig. 1. Growth in memory of the different machines is shown with an indication of the type of storage elements and degree of their integration.

Table

memory	computer model	capacity, M bytes	basic design	access time, ns	cycle time, ns
YeS3206	YeS1060, YeS1045, YeS1052	1.0 (64Kx144r)	rack	650	1200
YeS3262	YeS1022	0.5 (128Kx40r)	frame	800	1200
YeS3263	YeS1033	1.0 (256Kx40r)	"	800	1200
YeS3268	YeS1035	1.0 (128Kx72r)	"	800	1200
YeS3267	YeS1045	1.0 (64Kx144r)	"	650	read - 720 write - 840
YeS3266	YeS1060, YeS1065, YeS1052	8.0 (256Kx288r)	rack	520	850

The Series-3 computer processors, just as the Series-2, will include very high-speed memeories that include general-purpose registers, address matrix, address translation buffer, ultra high-speed buffer storage and microprogram or control memory. The highest speed is required of the address matrix and the translation buffer. Series-3 computers with a throughput of several millions of operations per second and a machine clock time range of 40 to 80 ns require the use of general-purpose registers, address matrix and translation buffer with an access time of no more than 10 to 15 ns. Since these memories need only a 1 to 2K byte capacity, they can be built with ESL [emitter-coupled logic (ECL)] memory microcircuits with a capacity of about 64 bits and standard access time of 5 ns.

Fig. 2 shows average computer throughput as a function of buffer and microprogram memory speed. Parameters of Series-2 YeS computers are used to show the relationship. Series-3 ultra high-speed buffer storage is to have an access time ranging from 20 to 40 ns and differs from Series-2 storage by the increased information capacity reaching 64K bytes. In the Series-2 YeS1060 model, the 8K byte buffer storage is built with 500RU410 microcircuits with a capacity of 256 bits and maximum access time of 40 ns [2]. Buffer storages for the higher Series-3 models are based on using ECL memory microcircuits with a 1024 bit capacity and access time of 10 to 15 ns.

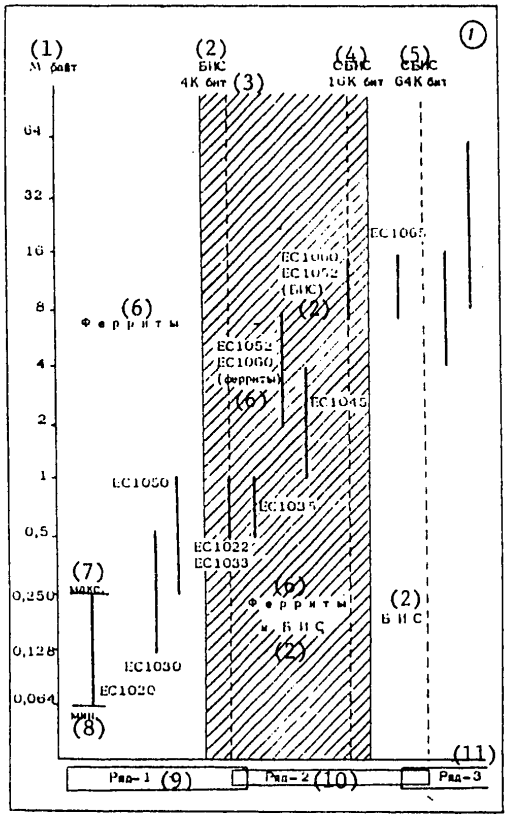


Fig. 1. Evolution of YeS computer main memories

Key:
1. M bytes
2. LSI
3. 4K bits
4. VLSI, 16K bits
5. VLSI, 64K bits
6. ferrites
7. maximum
8. minimum
9. Series-1
10. Series-2
11. Series-3

EC = YeS

It is assumed that ECL memory microcircuits with a 4096-bit capacity will not have sufficient speed for use in higher Series-3 models, but may be used in lower models.

It should be noted that ECL microcircuits have high power consumption (about 0.5 to 0.9 W per package), and this does not permit obtaining maximally possible density of packaging on the boards because of difficulties in ensuring thermal conditions. However, maximal density of packaging is needed not only to reduce size, but also to reduce length of communication and increase speed. Actually, it can be expected that high-speed buffer storage with a capacity of 64K bytes can be placed into one standard panel of YeS computers when microcircuits with a degree of integration of 1024 bits are used.

Microprogram or control memories for future models must have the same high-speed as the buffers: access time should range from 20 to 40 ns,

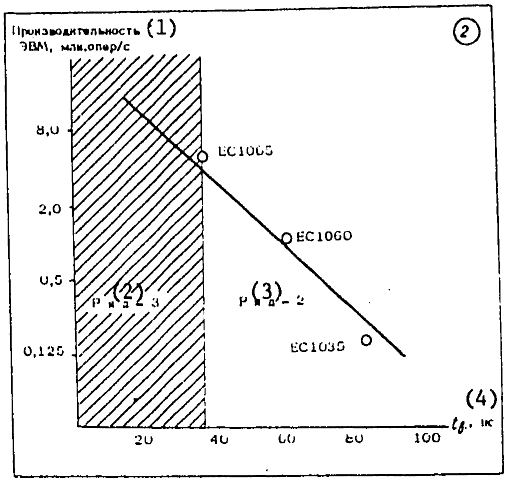


Fig. 2. Average computer throughput as a function of buffer and microprogram memory speed

Key:
1. computer throughput, millions of operations per second
2. Series-3
3. Series-2
4. access time, ns
   EC = YeS

while the capacity of these memories may reach 128K bytes or more. It is obvious that they can be built with the same elements having a capacity of 1024 bits, just like the buffer storage, and they can be put in two panels of the YeS computer.

The experience of developing and using microprogram memories in programmable microcircuits of PZU [read-only memory (ROM)] shows that the latter are used most advantageously in control devices, for example, in magnetic disk storage controllers. Using them in computer processors simplifies operation, but complicates changing microprograms.

Thus, two ECL memory microcircuits are needed for the processors in higher Series-3 YeS computer models: 1024 bits with a standard access time of 10 ns and 64 bits with a standard access time of 5 ns. The minimal nomenclature of the memory microcircuits for the processor makes it possible to raise the quality of the products and lower the cost for producing the models. In the process, testing the microcircuits is simplified, testing cost is reduced, test equipment is simplified, and the level of mastering the production of the microcircuits is raised both in connection with working out the TU [engineering specifications] and the quality of the microcircuits proper. Because of the small nomenclature of the memory microcircuits, solving the problem on standardization of the storage TEZ's [standard exchange cards] is simplified, which in turn permits developing cheaper and more efficient test equipment for them.

ECL microcircuits are the fastest, but also the most complex in a technological sense, which affects their cost. It can be assumed with a sufficient degree of certainty that in the next few years, the cost per bit of

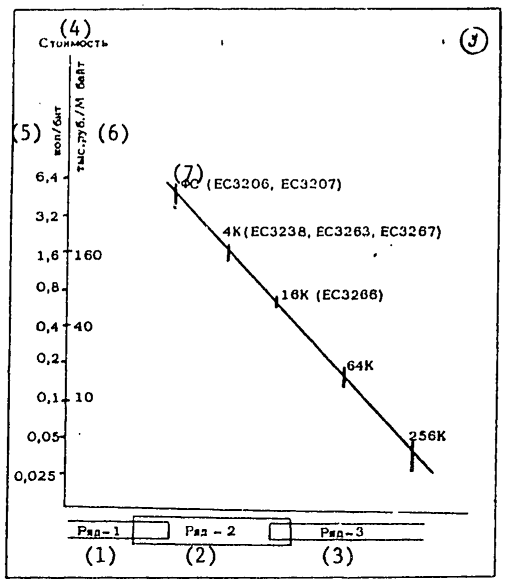


Fig. 3. Proportionate main memory cost as a function of degree of integration of the elements

Key:
1. Series-1
2. Series-2
3. Series-3
4. cost
5. kopecks per bit
6. thousands of rubles per M bytes
7. FS [ferrite core]

EC = YeS

information in processor memories will be several kopecks, and with a relatively large information capacity of memory (64K bytes for buffer and 128K bytes for microprogram memory) the cost of them is becoming comparable to the cost of main memory built with MOP [metal-oxide-semiconductor (MOS)] LSI.

The evolution of main memories is linked to the development of MOS LSI dynamic memory microcircuits.

In world practice, there is a steady trend of quadrupling the degree of integration of dynamic memory microcircuits every two years, and the cost per bit is being cut about in half during this period.

In addition to increasing the degree of integration and lowering cost, other characteristics of microcircuits are being improved. Thus, for example, a microcircuit with a capacity of 1K bits required several power sources, several special control clock signals and a complex system of synchronization; a circuit with a capacity of 4K bits has a simple system of synchronization, requires only one special control signal, has increased speed and is assembled in a 22-lead package; although a circuit with a capacity of 16K bits requires three power sources, it dissipates less power, is compatible on all leads with TTL [transistor-transistor-logic] logic circuits, and is assembled in a 16-lead package thanks to multiplexing of address buses.

Further improvement of dynamic memories is aimed at developing microcircuits with a capacity of 64K, operating from one +5V power source, having an access time of 100 to 150 ns, and arranged in a 16-lead package, of which one lead may remain free for future use when circuits with a capacity of 256K bits emerge. Decreasing the number of microcircuit leads from 22 to 16 and decreasing the package sizes makes it possible to put twice as many packages on boards since the small dissipated power of MOS memory microcircuits does not impose limitations on packing density. With 16K-bit microcircuits, a 16M-byte memory, which exhausts the possibility of the present addressing system, can be created in a YeS computer bay. Increasing the degree of integration of memory microcircuits to 64K and 256 K will permit placing a 16M-byte memory in two panels and in a half-panel, respectively, of a YeS computer.

The cost per bit in the conversion to an arrangement for 16K-bit microcircuits will be 0.6 kopecks/bit, while for 64 to 256K-bit circuits a cost of 0.1 to 0.03 kopecks/bit is expected.

Figure 3 shows the trend of change in proportionate cost of main memories as a function of storage element integration using YeS computer memories as an example.

If the degree of integration of dynamic MOS memory microcircuits quadruples every two years, then this means that during the development of a computer model, microcircuits may emerge with a higher degree of integration than that at the start of the development. So that the computer is fitted with the most advanced memory, the memory can be developed with microcircuits that exist at the time and then the model already developed can be modernized when the next microcircuit emerges. However, this means that full-scale development of two memories is needed, which naturally increases development costs and requires availability of certain resources.

Another way is to standardize memory as much as possible, and in the first place, to standardize memory interface. In the process, memory from a preceding model can be used in a new computer model at the stage of development and assimilation into production and be subsequently replaced when new microcircuits are available. In addition, this standardization will promote reducing the nomenclature of test equipment (for microcircuits, TEZ's and devices) and will allow memory production in a specialized enterprise.

Introducing standardized devices should be preceded by a detailed economic and engineering analysis, since one can expect that standardized devices will be somewhat more complex than specialized ones and naturally more expensive.

A few words must be said about static MOS memory microcircuits. These are technologically more complex than dynamic ones. This means that the degree of integration of static microcircuits will always be less than that of dynamic circuits, and the cost per bit higher. But static microcircuits do have the advantage of increased speed. As the technology advances, the high speed of static MOS memory increases and approaches that of bipolar memory. Thus, for example, MOS memories have been developed with a capacity of 1K that surpass the speed of TTL-circuit memories of the same capacity [4].

A definite area of application for static memories is in small-capacity peripherals in which it is desirable to not have control circuits requiring regeneration that dynamic memories need. Static memories are more desirable in devices operating in real time since dynamic memories are inaccessible for reference during the regeneration process which is three to five percent of the time.

Modern semiconductor memories for computers with high throughput are devices containing thousands of memory microcircuits and hundreds of electronic framing elements. If specific measures are not taken to enhance reliability, then the mean time between failures can amount to hundreds or even dozens of hours, which is clearly unacceptable.

The failure rate for elements in the break-in period is especially high: one to two orders higher than in a device that has been broken in [5].

To raise the reliability of the devices, first of all it is necessary to use the Hamming code that makes it possible to correct single failures or malfunctions that occur, and to perform regular preventive maintenance to replace the microcircuits that have failed. These measures permit raising memory reliability by one to two orders. For especially critical devices, microcircuits have to undergo electrothermal aging prior to being installed in a storage device. Reliability can also be raised by the manufacturer making a trial run of the devices for several days, introducing break-in of the devices into the TU [engineering specifications]. Even after this, the user will experience increased intensity of failures the first time. This means that an increased number of storage TEZ's has to be included in the kit of ZIP [spare parts, tools and accessories], and it is desirable that the user have service equipment to repair them.

The question on the need for service equipment is an economic and engineering one, since detection of malfunctions, diagnostics and repair can be done during preventive maintenance on the computer. An additional increase in reliability can be gained by increasing the amount of equipment for the electronic framing elements so that a failure of an electronic framing element does not lead to failure of several units.

As a general prinicple for raising reliability, it is advisable to use microcircuits with the least number of leads and the highest degree of integration. Memory reliability (mean time between failures) is raised when the quality of development of storage TEZ's is raised, namely when there is a small level of noise on both the supply and signal buses. From this aspect, the use of multilayer printed circuit boards is desirable.

Considering that as the degree of integration of microcircuits grows, the proportionate cost of memory will decline and the development of devices simplified, let us look at the possibility of building external storage with solid-state elements of memory. It is evident that solid-state memory elements can provide higher speed. However, considering that the cost per bit in contemporary VZU [external storage units] using magnetic disks and tapes is extremely low, the potential area of application of solid-state memory elements in the early stages will be limited to devices with relatively small capacity.

Thus, for example, we can look at the possibility of replacing the YeS5064 fixed-head magnetic disk storage unit that has a capacity of 11M bytes for storing an operating system. As potential replacements for this storage unit, let us look at semiconductor memories with random access made with MOS units and those with serial access made with PZS [charge-coupled device] structures, as well as storage made with magnetic bubbles (TsMD) and electron beam tubes (ELT) with a semiconductor target [6]. A comparison of these types of storage with a magnetic disk storage unit and with each other has to be made on the major characteristics of external storage: specific information density, rate of information exchange and cost per bit of information.

The speed of semiconductor and electron beam storage devices is appreciably higher than that of electromechanical ones; however, when a standard channel is used, the criterion of speed is not paramount, since system speed is limited by channel throughput. In addition, when power-dependent semiconductor memory elements are used, the problem of developing an economic emergency power supply system has to be solved.

Information density and cost per bit of information are largely determined by the degree of integration of the microcircuits for semiconductor main memories and charge-coupled device [CCD] structures of storage units, the degree of integration of microassemblies for magnetic bubble storage devices and the capacity of the electron beam target. Let us look at what degreee of integration is needed so that an 11M-byte storage unit and its power supply could be put into a YeS computer frame and so that the cost per bit would be 0.1 kopeck, which corresponds to the basic characteristics of the fixed-head magnetic disk storage unit.

Calculation (see fig. 3) shows that to obtain the needed information density and unit cost, the degree of integration of semiconductor microcircuits with dynamic MOS memory and of CCD microcircuits has to be 64K bits in a package. To obtain the needed information density in magnetic bubble units, the capacity of one magnetic bubble microassembly has to be 1M bits. This is because the microassemblies take up about four times more area on a card than a semiconductor microcircuit, since a magnetic bubble microassembly includes, besides the storage chip proper, inductors to create a rotating magnetic field and a system of permanent magnets to create a bias field. In addition, magnetic bubble microassemblies need a rather large number of electronic framing circuits, which take up three times more space on the cards than the magnetic bubble microassemblies [7].

Hence it is evident that to attain the same information density as semiconductor memory, the magnetic bubble microassembly has to have a degree of integration higher by a factor of 16 than semiconductor microcircuits. It should be noted that world practice shows just that relationship of capacities for magnetic bubble and MOS microcircuits. Thus, MOS microcircuits with a 4K-bit capacity correspond to magnetic bubble microcircuits with a 64K-bit capacity, and MOS microcircuits with a 64K-bit capacity correspond to magnetic bubble microcircuits with a 256K-bit capacity. One alternative solid-state external storage unit may be MOS structure storage with electron beam access [8]. The rated capacity of an electron beam tube with a 50 cm length and 10 cm diameter is 4M bytes which is adequate for building an 11M-byte external storage unit to fit in the YeS computer frame. This solution seems to be feasible in an engineering sense. With this degree of integration, the required cost of 0.1 kopeck/bit could be obtained.

Another application of large-capacity solid-state storage could be intermediate storage needed when building archival storage systems. The speed of intermediate storage with a capacity of about 100M bytes, acting as a buffer between the computer and the relatively slow, but large-capacity, magnetic tape storage units, largely determines the speed of the entire system of archival storage. A rough estimate shows that to construct a 100M-byte device to fit in a YeS computer bay and having a unit cost of 0.01 to 0.02 kopeck/bit, the capacity of semiconductor microcircuits has to be not less than 256K bits in a package, of magnetic bubble microcircuits--not less than 1M bits, and of electron beam tubes--not less than 4M bytes.

Conclusions. Future models of Series-3 YeS computers should include a set of processor memories with a capacity from 1 to 128K bytes with access time ranging from 10 to 40 ns; the element base will be ELS [emitter-coupled logic] memory microcircuits with a degree of integration from 64 to 4096 bits.

The main memories in the first phase of the Series-3 should have a capacity of up to 16M bytes with an access time of no more than 0.5 microsecond. The element base for these memories should be microcircuits with integration of 16 to 64K bits. In some cases, it is also advisable to use magnetic microferrite memories.

External storage units, used to store the operating system and traditionally on magnetic disks, may be replaced in Series-3 machines by solid-state integrated storage devices for enhanced speed and reliability.

BIBLIOGRAPHY

1. Sharunenko, N. M., and Fateyev, A. Ye. "General-Purpose Digital Computer Memory," in "Tekhnika" [Engineering], Moscow, Znaniye, 1977, pp 27-43.

2. Sharunenko, N. M. "Evolution of Unified System Main Memory," in "Vychislitel'naya tekhnika sotsialisticheskikh stran" [Computer Technology of the Socialist Countries], Moscow, Statistika, 1977, 2nd ed., pp 21-32.

3. Hodges, D. "Semiconductor Elements of Digital Storage Devices: Present Status and Prospects," TIIER, 1975, Vol 63, No 8, pp 54-67.

4. Keypis, R. "High-Speed Static ZUPV [Random-Access Memories]: Present Status and Prospects," ELEKTRONIKA, 1979, Vol 52, No 9, pp 40-57.

5. Madzharova, T. B. "Reliability of Large-Scale Integrated Circuits," ZARUBEZHNAYA RADIOELEKTRONIKA, 1978, No 1, pp 143-147.

6. Keypis, R. "Semiconductor Memory That Replaces Disk Storage," ELEKTRONIKA, 1979, Vol 52, No 12, pp 6-7.

7. Brayson, D.; Klover, D.; and Lu, D. "Set of Large-Scale Integrated Circuits for Combined Use with Magnetic Bubble Storage," ELEKTRONIKA, 1979, Vol 52, No 9, pp 23-32.

8. "Electron Beams Take Aim at 2G Byte Memory Systems," ELECTRONIC DESIGN, 1978, Vol 26, No 6, pp 21-22.

[408-8545]

8545
CSO: 1863

UDC 621.387.3

GRAPHIC DISPLAY WITH MATRIX GAS-DISCHARGE PANEL

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 66-69

[Article by Aleksandr Maksimovich Smolyarov, candidate of technical sciences, RRTI [Ryazan' Radioengineering Institute] (Ryazan'); Vladimir Vasil'yevich Mechetnyy, engineer, RRTI (Ryazan'); and Margarita Aleksandrovna Dudina, engineer, RRTI (Ryazan'); received by editor 27 Nov 78 (after revision, 7 Jul 79)]

[Excerpt] Basic technical characteristics of the device

Input code	seven bit (with sign-variable code, one bit is signed);
Maximum permissible rate of incoming information, kHz	300;
Control circuit voltage supply, V	+5±5% and +3±5%;
Power-supply voltage, V	+250 - 20% and +150 ±20%;
Element base	K133, K155, K141, K166, AL304

The GIP-10000 [7] DC gas-discharge panel is used as the GIP [gas-discharge display panel].

The merits of this GIP are compactness, low power-supply voltage, absence of a code-analog converter, sufficiently high precision (error ±1%) and brightness (not less than 100 cd/m^2), and resistance to mechanical effects. These displays may be widely used in ASU [automated control systems], ASUTP [automated systems for control of industrial processes], centralized monitoring systems, information-metering systems and as computer output devices.

[408-8545]

8545
CSO: 1863

UDC 681.3.48./14

FEATURES OF SPECIAL-PURPOSE DIGITAL COMPUTER DESIGN WITH A MICROPROCESSOR SET

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 38-43

[Article by Yuriy Yakovlevich Berson, engineer (Leningrad); Leonid Veniaminovich Gol'dreyer, engineer (Leningrad); Lev Yakovlevich Lapkin, candidate of technical sciences (Leningrad); Valentin Georgiyevich Nosov, engineer (Leningrad); Nikolay Petrovich Sedov, engineer (Leningrad); Vladimir Borisovich Smolov, doctor of technical sciences, LETI [Leningrad Electrotechnical Institute imeni V. I. Ul'yanov (Lenin)] (Leningrad); and Vladimir Tikhonovich Startsev, engineer (Leningrad); received by editor 22 Aug 77 (after revision 3 Apr 79)]

[Excerpts] The largest effect from building SVM [special-purpose digital computers], program compatible with the YeS computers, with BIS [large-scale integrated circuits] and MP [microprocessor] sets will be achieved with series assimilation of the K583 series IS [integrated circuit] intended for realization of this system of instructions. Since these integrated circuits are still in the development stage, while industry is currently producing microprocessors architectually incompatible with the YeS computers, this work discusses the possibility and advisability of using them in these special-purpose digital computers.

Conclusions

1. To build special-purpose digital computers with the YeS computer instruction set, it is advisable to use K584 IK1 microprocessors as the central processor for SVM with small and medium throughput and as a channel processor for SVM with high throughput.

2. It is possible to build an SVM with a throughput on the order of 1 million short operations/second of the format RX using an ALU [arithmetic-logic unit] for the arithmetic unit and a microprocessor for the channel processor.

3. To expand the possibilities of using a microprocessor as the central processor in an SVM with high throughput, program compatible with the YeS

computers, the microprocessor must meet the following additional requirements:
--have 16 internal general-purpose registers;
--provide independent access to two other general-purpose registers;
--have two-address operations in the system of operations, performable on operands stored in the various general-purpose registers;
--provide generation of attributes of overflow and zero value of the mantissa of the result.

Of the currently known foreign microprocessors, these requirements are most fully met by the Am 2901A microprocessor produced by Advanced Micro Devices.

BIBLIOGRAPHY

1. Drozdov, Ye. A.; Komarnitskiy, V. A.; and Pyatibratov, A. P. "Elektronnyye vychislitel'nyye mashiny Yedinoy sistemy" [The Unified System of Electronic Computers], Moscow, Mashinostroyeniye, 1976, 672 pages.
2. Cheglanov, L. V.; Klubkov, V. V.; Novozhilov, A. S.; and Filatov, V. N. "Special Purpose Digital Computer Made with Microprocessors with the YeS Computer Instruction Set," ELEKTRONNAYA TEKHNIKA, series 11, 1975, issue 2, pp 9-18.
3. Klubkov, V. V., and Madatov, A. A. "Computers with the YeS Computer Instruction Set," ELEKTRONNAYA TEKHNIKA, series 10, 1977, issue 6, pp 14-18.
4. Khorton, Engleyd, Makgi. "Integrated Injection Logic: A Major Advance in the Technology of Bipolar Large-Scale Integrated Circuits," ELEKTRONIKA, 1975, Vol 48, No 3, p 24.
5. Belous, A. I.; Boldyrev, V. P.; Kaloshkin, E. P. et al. "General-Purpose Single-Chip Four-Charge Microprocessor," ELEKTRONNAYA PROMYSHLENNOST', 1977, issue 5 (59), pp 55-57.
6. Rotnov, S. V.; Topolov, A. D.; Federov, Yu. T. et al. "Low-Power Transistor-Transistor Logic Microcircuits for Building Processor Systems," ELEKTRONNAYA PROMYSHLENNOST', 1977, issue 2 (56), pp 75-79.
7. Khasson, S. "Mikroprogrammnoye upravleniye" [Microporgram Control], Moscow, Mir, 1974, 2d ed., 478 pages.
8. "Microprocessor Data Manual," ELECTRONIC DESIGN, 1977, Vol 25, No 21, p 170.

[408-8545]

8545
CSO: 1863

UDC 681.3:621.797

COMPUTER MAINTENANCE QUALITY INDICATOR

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 21-23

[Article by Yuriy Artem'yevich Yershov, engineer, VNITI [All-Union Scientific Research and Industrial Design Institute of the Pipe Industry] (Dnepropetrovsk); received by editor 25 Jun 79]

[Excerpt] Table 2. Manufacturer's recommended time for preventive maintenance on the YeS-1022 computer

		Days computer operated in a week		
Type	Hours	5	6	7
semiannual	36	72	72	72
monthly	16	160	160	160
biweekly	4	48	48	48
daily	1	235	287	340
total		515	567	620

[408-8545]

8545
CSO: 1863

UDC 681.327.8

PROBLEM OF INTERFACING DATA TRANSMISSION EQUIPMENT WITH A UNIFIED SYSTEM COMPUTER

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 132-135

[Article by Petr Matsovich Ivanov, candidate of physicomathematical sciences; and engineers Vladimir Titovich Kerefov, Pavel Georgiyevich Teslya, Igor' Ivanovich Gridyakin, and Nikolay Alekseyevich Tkhishev; all from the KBF VPTO [expansion unknown] "Rossel'khoztekhsistema" [system for All-Russian Agricultural Equipment Association] (Nal'chik); received by editor 23 Jan 79 (after revision, 29 May 79)]

[Excerpts] YeS [Unified System] computer hardware currently permits interfacing data transmission channels with a computer input/output channel using data transmission multiplexors (MPD-1, MPD-2, MPD-3) [2]; however, funding limitations for acquisition of this equipment hinder their practical use and in a number of cases justify developing nonstandard hardware performing a similar function.

This work deals with the question associated with developing a single-address adapter to provide interface between the APD-MA [data transmission equipment; MA - expansion unknown] and the YeS-1022 computer. The single-address adapter or interface device (US) effects coordination of the physical parameters of signals and algorithm of operation of the computer multiplexor channel and the joining of the data transmission equipment, and performs the following functions:
--accesses APD in accordance with address sent from the computer;
--executes commands and instructions for computer control;
--controls modes of operation of the APD;
--controls flow od data between the computer and the APD;
--checks characters for correctness of data sent and received;
--informs computer on status of the APD and the US itself;
--handles emergency situations automatically and subsequently restores operation.

The interface device was designed on the base of a standard type frame, the K2KK6-6U3; integrated microcircuits of the 133 and 140 series were used as the element base. Controls and displays showing current status of the device are located on the front panel of the frame. Connectors are located on the rear wall of the frame. The interface device connects to the computer input/output channel with two pairs of "Nabor" connectors, similar to standard external devices of the Unified System (IKM-2 cables), and connects to the APD with two RPP 40G1-1T3 connectors.

[408-8545]

8545
CSO: 1863

UDC 681.326

OFF-LINE DEBUGGING DEVICE FOR THE 'ELEKTRONIKA S5-11' MICROCOMPUTER

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, 1980 pp 19-20

[Article by V. V. Sumin, candidate of technical sciences, A. B. Vasil'yev and N. M. Kalinin, engineers]

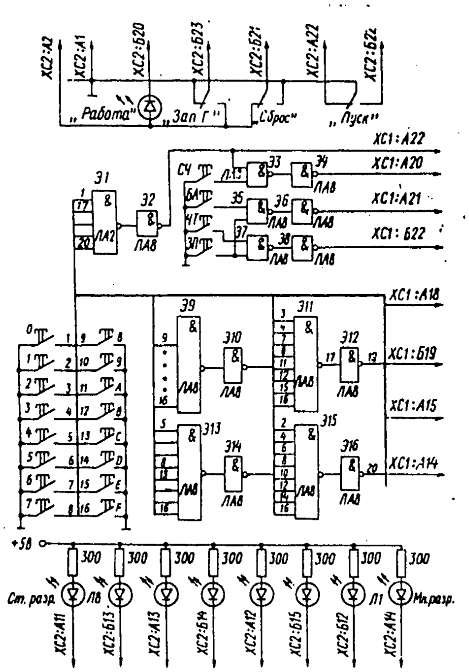
[Text] The very simple debugging device described in this article can be viewed as the first practical step toward assimilation of microcomputers on a chip for the overwhelming majority of users. The device is an uncomplicated console for input, monitoring and execution of programs; thus an "Elektronika S5-11" microcomputer outfitted with such a console becomes an off-line programmable automation device.

The functional arrangement of the device was examined by the authors [1]. The schemata of the debugging console and the memory unit will be given later with brief descriptions. Figure 1 shows the console scheme which includes the controls for monitoring the "Elektronika S5-11" microcomputer and for setting it in the operating mode: "Work", "Inhibit G", "Dump", "Start" (KhS2 : A2, KhS 2 : A1, KhS 2 : B20, KhS 2 : B 23, KhS 2 : B 21, KhS 2 : A 22, KhS : B22); functional buttons (SCh, BA, Cht, ZP) and a 0-F hexadecimal keyboard with combiner (E1-E16), connected to a digital input channel of 3 microcomputers (KhS1 : A14, KhS1 : A15, KhS1 : B19, KhS : A18, KhS1 : A20, KhS1 : A 21, KhS1 : B22, KhS1 : A22); point LEDS L1-L8 to display the information byte connected to the digital output channel of three microcomputers (KhS2 : A14, KhS2 : B12, KhS2 : B15, KhS1 : A12, KhS2 : B14, KhS2 : A13, KhS2 : B13, KhS2 : A11).

The diagram of the read-only memory (ROM) in which the console exchange routine is contained, is shown in Figure 2. The ROM unit contains elements (A1-A8) for interfacing with the address (ShA) and information (ShI) busses of the "Elektronika S5-11" microcomputers (KhS4 : B13, KhS4 : B14, KhS4 : A11, KhS4 : A10, KhS4 : B8, KhS4 : A3, KhS4 : B16, KhS4 : B17, KhS4 : A2, KhS4 : B1, KhS4 : A18, KhS4 : A20, KhS4 : B4, KhS4 : A19, KhS4 : B3, KhS4 : B7, KhS4 : A9, KhS4 : B11, KhS4 : A12, KhS4 : A16, KhS4 : B22, KhS4 : A17, KhS4 : B19); memory access shaper (I2-I6); address decoder (D, I1); diode storage H which are connected according to the console exchange routine.

The sequence of operations for the operating mode device may be described as follows.

1. Preparation for operation: powering up, sequential depression of buttons "Inhibit G", "dump", "Start", "Dump", "Start"; "Operate" light indicates the device is ready.



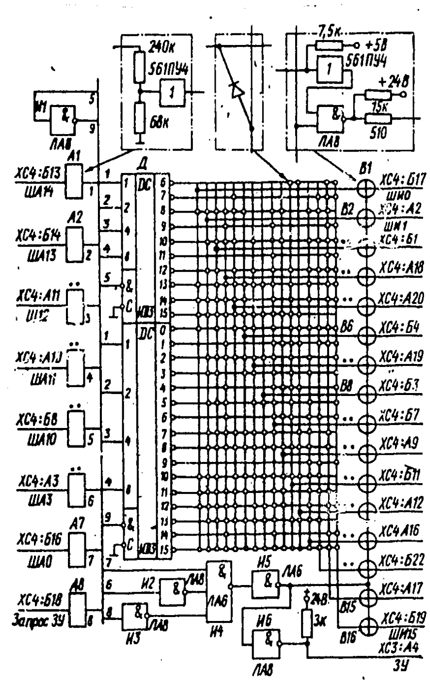


Fig. 1. Debugging console. Fig. 2. Debugging memory.

2. Input of a hexadecimal number into the information buffer (BI) (common register R5) of the microcomputer which is accomplished by the sequential depression of buttons 0-F (see Figure 1), starting with a high-order bit St. razr; the low-order byte of the BI is monitored by LEDs L1-L8.

3. Recording of the base address upon entry or read-out of information whose initial byte address is input into the BI of the microcomputer; the contents of the BI is transfered to the address buffer BA by pressing the BA button (common register R6).

4. Byte entry of information into main memory of microcomputer; the information byte is selected; the contents of the low-order byte of BI is entered according to the BA address by pressing the ROM button; as a result, the BA address is automatically increased by "+1" and a new value of the BA address is indicated; the high-order byte is recorded in the memory cell first if it has an even address; information may be input into the internal main memory according to the addresses 000E-00FF.

5. Byte read-out of information from microcomptuer memory. The code of the information byte whose address is indicated in BA is monitored with the LEDs by pressing the button ChT; it necessary to correct it, a new byte value

is input into the BI; replacement of the byte address is executed by pressing the ROM button.

6. Transmission of control of the debugged program is accomplished by pressing the button SCh; the contents of BI is recorded in the instruction counter.

7. Transmission of control to the console exchange routine, executed by sequential depression of the buttons "Dump", "Start".

To realize off-line debugging device it is necessary to have the following components: microcomputer "Elektronika S5-11" with power source connection circuit; stabilized voltage unit +24, +5 and +1.5V (0.7, 0.5 and 1 mA, respectively); type 155LA2, 155LA3, 155LA6, 155LA8, 155ID3 and 561PU4 ICs; AL307 or AL102 LEDs; triggering and switching buttons; point diodes of any type or diode assemblies; MLT 0.125 resistors.

It should be noted that the equipment part of the device may be simplified if there are more convenient components available (e.g., when 561LN1 microcircuits are used in the interface circuits the number of storage diodes is reduced by 50, and the use of the 155RYe3 component greatly simplifies the memory unit).

The off-line debugging device proposed in the article for the "Elektronika S5-11" microcomputer was realized in studies of the Volga-Vyatsk Regional Center for Debugging and Application of Microcomputers of the family "Elektronika S5" and has been successfully used in practice.

[43-8617]

8617
CSO: 1863

UDC 681.327.2.082.5

SEMI-PERMANENT OPTOELECTRONIC MEMORY

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, 1980 pp 28-29

[Article by L. P. Chaykovskiy, engineer]

[Text] The first attempts at creating semipermanent computer memory with light-pulse sensing of data were undertaken back in the early 1960s [1]. However, only the latest developments in high-speed emitters, photoelectric sensors and fiber lightguides have made it possible to develop semipermanent optoelectronic memories (POZU) with high operating values [2-5].

A typical aspect of the fiber-optic system of this kind of device is that it can be used to create a meomry with an information capacity equal to the number of emitters times the number of photoelectric sensors. In this case, the fiber-optic system consists of two light guide subsystems, the first of which interfaces emission with the mask and the second transmits it from the mask to the photoelectric sensors. This POZU deisgn has substantial shortcomings, however, which diminish its values (density of information recording, speed, overall dimensions). The most serious shortcoming is the presence of a discontinuity in the fiber-optic channel which predetermines increased optical losses as compared to a continuous light guide of the same length. They are a function of the magnitude of the gap between the lightguides and the solid angle of intercept of emission passing through the information mask, which in turn decreases with decreasing light guide diameter.

In addition to losses due to reflection upon entry into and emergence from the lightguide, these losses may reduce the level of emission energy flux acting on the photosensor to less than 1/10 of its initial value.

In this article a POZU system is described in which, in addition to retention of the relationship between information capacity and the number of optronic components (emitter and photoreceivers), a reduction in emission energy losses is guaranteed (Figure 1). This is achieved by structural features of the fiber-optic system consisting of regular light guides (RS). Each of these in turn consists of individual single fibers. The arrangement of individual fibers forming the input end of the RS corresponds to the arrangement configuration of the information positions in the mask.

Light channels (SK) of the single fiber in the input end form a discontinuous line, i.e., the part of the information mask surface comprising a POZU word is line-scanned. Thus each SK meets one series of sensed words. The light channels on the output side of the RS are collected into a packet forming the matrix; while the SK entering into one of the matrix columns abut against one of the photosensors of the receiving matrix. The information mask is arranged in the frontal plane of the input ends of the RS. Excitation of one emitter makes it possible to sense information word by word on the same matrix of photosensors.

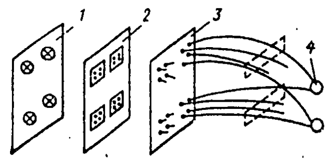


Figure 1. Structural diagram of POZU:
1) LED matrix; 2) information mask;
3) fiber optics system; 4) photodiode
matrix.

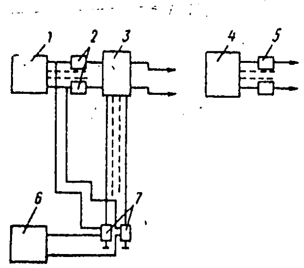


Figure 2. Block diagram of POZU control:
1) decoder; 2) current pulse shaper;
3) LED matrix; 4) photodiode matrix;
5) wide-band amplifiers; 6) decoder X;
7) switching circuits.

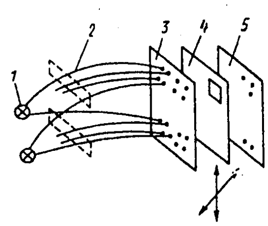


Figure 3. Structural diagram of mask generator:
1) LED matrix; 2) single fiber; 3) fiber-optics
system; 4) diaphragm; 5) photosensitive chip.

The manufacturing technology of these RS, from which the system is later assembled, is no more difficult than that of ordinary RS for image transmission and furthermore, it can be automated.

In this lies its substantial difference from known schemes of POZU design [6] where the logical positions in the information mask are connected to the photosensor matrix, since here we assume individual laying of lightguides. Evidently this design, with an information capacity of more than 10^3 bits (and consequently, including individual laying in the fiber-optics system) becomes really impracticable due to the extremely great labor-intensiveness of manufacture. At the same time, the fiber-optic system in this POZU design may be enlarged by increasing the number of base RS, which leads to increased packet height and increased number of SK forming a single column as well: the "tyranny of numbers" problem is eliminated.

A polymer lightguide RS was used as the base element of the fiber-optic system; several of its parameters are better than those of glass lightguides; it has a diameter of 0.1 millimeter and length of 200 millimeters. Each RS contained 100 individual SK (single fibers). The input end of the RS has an area of 3 x 3 square millimeters, on which are arranged 100 (10 x 10) input ends of the individual single fibers with a spacing equal to 0.3 millimeter. The input end of each RS is located 2 millimeters from its neighbor. The fiber-optic system of 400 RS arranged in a 20 x 20 matrix has a transverse dimension of 100 x 100 square millimeters.

When a photosensitive chip 100 x 100 square millimeters is used as the information mask, a POZU capacity of 4 x 10^4 bits is achieved; it is recorded in the form of individual words on the surface of the photochip. The distance between individual words is 2 millimeters. The presence of such intervals makes it possible to eliminate crosstalk between continguous sections of the photochip, though it also reduces effective information recording density.

GaAs lightdiodes with rated emission output of 10 mW at 100 mA (type AL 107-B) were used as the emitters. The control circuit, consisting of decoders, excitation current pulse shapers and switches, ensured selection of one of the lightdiodes of the emitter matrix (Figure 21), whose emission was covered by the opposite section of the information mask which contained one word.

The matrix of photosensors consisted of 100 Si photodiodes with a sensitivity of 450 microamperes/mW, intrinsic capacity of 10 pF, dark current of less than 10 nA. Each of the 100 photosensors was loaded with 1 kohm. Parallel sensing in 100 channels makes it possible to reduce stray noise between elements of the photosensor matrix to the minimum and obtain high reliability of information sensing and speed.

The entire process of decoding and excitation of the transmitting matrix, depending on the type of lightdiode, lasts 50 to 200 nanoseconds. The total access time is 70 to 250 nanoseconds, which is defined by the speed of the lightdiode. The signal tapped from the load resistor in the photodiode circuit was amplified by a wideband amplifier with gain from 3 x 10^2 to 10^3.

Because the information mask manufactured using a phototypesetting machine can be easily replaced, the reprogramming of POZU presents no problems.

Based on the fiber-optic system of the POZU an information mask generator was also developed whose structural diagram is shown in Figure 3. Series and parallel excitation of LEDs situated in the photodiode positions, where the diaphragm has an area equal to that occupied by one word in the front plane of one input end of the RS, made it possible to expose the photochip placed behind the diaphragm. The diaphragm was shifted to align with the input end of a second RS to record the second word.

References

1. Goffman, G., Dzheffreys, D. Permanent computer memory with data sensed by light pulses, In: Opticheskaya obrabotka informatsii [Optical processing of information], Moscow: Mir, 1966.

2. Optical memories, RADIOELEKTRONIKA ZA RUBEZHOM 43 1971 p 18.

3. Osinskiy, V.I., Chaykovskiy, L.P. Principles of design of semipermanent memories based on grid-type and fiber-optics systems. In: Vychislitel'naya tekhnika v mashinostroyeniy. Vyp. 1. Minsk: ITK AN BSSR 1974.

4. Al'tman, L. Achievements in optoelectronics. ELEKTRONIKA 45 No 4 1972.

5. Osinskiy, V.I., Chaykovskiy, L.P. Optoelectronic semipermanent memory with optical switching. In: Avtomatika i vychislitel'naya tekhnika. Vyp. 7. Minsk: Vysheyshaya shkola 1977.

6. Styuart, R.D. Optical memories. ELEKTRONIKA 39 No 4 1966.

[43-8617]

8617
CSO: 1863

UDC 681.327.12

TOUCH-SENSITIVE DATA ENTRY KEY PADS

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, 1980 pp 31-32

[Article by A. I. Chebarev, V. N. Revenko, candidates of technical sciences; A. A. Am:rdzhanov, D. L. Kamalov, E. N. Farzane, engineers; and F. Ya. Agayaev, mechanic]

[Text] Touch-sensitive (sensory) keypads have lately found wide application in man-machine systems [1] in which information signal activation (FIS) is done by the operator's touching the metallic keyboard surface which has an electronic circuit with a relay output characteristic. Analysis of existing methods of design of these keypads has shown that FIS realized with impact excited oscillation loops are the most sensitive and reliable circuits [2]. This method may be utilized in designing keyboards. If is necessary to enter a small number of functional control instructions, however, it is advantageous to use FIS circuits similar to those described in study [3]; the latter are made in the form of individual modules.

The basis of the FIS whose electrical circuitry is shown in Figure 1 is a self-excited oscillator AG in the oscillation loop, which reacts to human interaction by abruptly decreasing oscillatory amplitude. The signal from the self-excited oscillltor AG reaches a threshold device PU which consists of an emitter follower, an equalizing capacitor C and a switching circuit which transforms the oscillation of the self-excited oscillator AG into a low-level voltage signal without operator intervention and a high-level signal when he intervenes. The entire FIS circuit of the keypad module is contained in three transistors V1-V3 of a type K1NT251 type transistor assembly.

The electrical circuit of a touch-sensitive data entry keypad TK is mounted on two PCBs inserted into special grooves inside the keypad housing, which is made of an insulating material (Figure 2). The housing base contains pin-type leads to connect the keypad to the control unit. At the end of the keypad module there is a contact surface connected to the FIS circuit which the operator touches. This area is available in two versions: as a metal plate or as a current-conducting section applied on a transparent glass substrate. In the latter case a notch is cut in the end of the module and a replaceable template indicating its functions is inserted.

The touch-sensitive keypad module developed by the Azerbaijan Institute of Petroleum and Chemistry imeni M. Azizbekov has overall dimensions of 20 x 20 x 65 millimeters, which fully coincides with the dimensions of electromechanical hermetic capacitor keypads used in standard data entry keyboards in computers and data display devices. This structural execution of the touch-sensitive keypad makes it possible to replace the mechanical contact keypads with touch-sensitive ones without redesigning the consoles; this increases the operating reliability and service life of the latter.

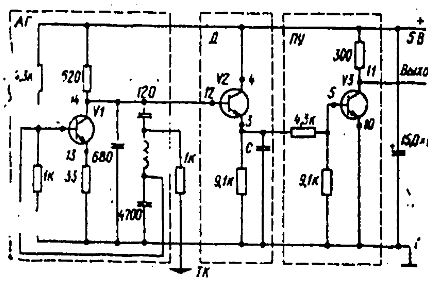


Figure 1. FIS Electrical Circuit.
D) detector; C = 0.015 pF.

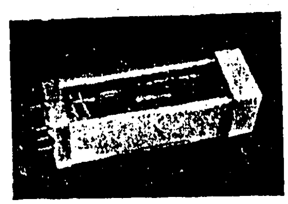


Figure 2. External appearance of a touch-sensitive keypad.

In view of the fact that this FIS circuit is a source of high-frequency noise, the housing of the keypad module is shielded to eliminate it.

The touch-sensitive data entry keypad described here has the following electrical parameters: output signal level with operator intervention V_{out}^1 greater than or equal to 4.5 V; without opprator intervention V^0_{out} less than or equal to 0.4 V; output current I^1_{out} equals 16 milliamperes at supply voltage of 5 VAC $\pm$ 10 percent and AG self-excited oscillator frequency of about 600 kHz in a range of operating temperatures from -10 to +60°C.

References

1. Yel'yashkvecih, S.A., Avtomaticheskoye upravleniye v televizorakh [Automatic control in televisions], Moscow: Energiya 1975.

2. Chebarev, A. I., Amirdhzanov, A. A., Kamalova, D.L. Questions of analysis of touch information activation. In: Problemy peredachi i preobrazovaniya informatsii. Kiev: IK AN USSR 1977.

3. Chebarev, A.I., Revenko, V.N., Amirdhzanov, A.A., Kamalova, D.L. Pre-CRT touch-sensitive data entry panel. PRIBORY I SISTEMY UPRAVLENIYA No 6 1979.

[43-8617]

8617
CSO: 1863

SOFTWARE

LANGUAGES OF REALIZATION FOR SYSTEMS PROGRAMMING

Novosibirsk YAZYKI REALIZATSII DLYA SISTEMNOGO PROGRAMMIROVANIYA in Russian 1979 signed to press 28 Nov 79 pp 1-15, 18-24

[Preprint by I. V. Pottosin, Computer Center of the Siberian Branch of the USSR Academy of Sciences, 130 copies, 24 pages]

[Excerpts] To define the subject of the work, one has to define what properties we shall consider determining for systems programming and what class of languages we will take for consideration.

1. Characteristics of Systems Programming. A factor in the evolution of information science in the latter half of the sixties and in the seventies is the isolation of the independent course of systems programming, with crystallization of the questions of its problems and methodology. Considering systems programming a key discipline in information science, Seegmeuller [1] defined it the following way.

"Systems programming is the field concerned with the description, design, properties, representation and functioning of specific sets of system algorithms. These sets of algorithms realize the linguistic and operational interface of a computing system under certain limitations, optimizational and dependent on the environment."

This definition of systems programming is somewhat indistinct; however, in our view, it can be taken as a first approximation, the refinement of which we shall engage in in those respects which we will consider necessary for the needs of this work.

Let us define a number of aspects of systems programming, important for further presentation, concerning its immediate program product.

Based on this definition, systems programming, as a rule, is a tool; its products are not directly applicable to solving some problem, but are used to provide a description of this solution. The link between systems programming and the technology of programming is organic, and the corresponding concepts have to go into the descriptive facilities of systems programming. By virtue of instrumentality, system programming products have to create an environment for themselves or access particular fragments directly, while the link to the environment in applications programming is indirect and implicit.

Systems programming does not create isolated programs, but /program systems/ [spaced for emphasis]. This aspect, which is becoming more and more pronounced in programming in general, is all the more essential for systems programming. The components of these program systems are quite heterogeneous to satisfy the diverse requirements of the users. Thus, for example, a translator is not created for a user, but a programming system that includes a translator as a basic, but not the sole part, and that contains a number of components that meet other aspects of creating programs, different from just an expression of an algorithm in a given programming language, namely, modifications, debugging, packaging of programs and so on. This dictates special attention to high-level facilities for program interface.

As a rule, a systems programming product either creates some program for a given specific machine and specific system, or realizes some execution with a dynamic link to an external medium, determined both by the computer installation itself and by the devices linked to it. In connection with this, the element of machine dependence for systems programming is substantially more important than for applications programming, the product of which, as a rule, processes some machine-independent information (with a precision up to the representation in the machine carriers).

It is characteristic of a systems programming product that it is used in some standard cycle of work on a computer which indicates the relatively high frequency of its execution and considerably increases the requirement for efficiency of the product (naturally, we are talking here about an end product, and not about experimental systems or test versions).

It is precisely systems programming that primarily deals with large program projects (large both in the size of the program product and in the number of developers), for which the duration of development is important by virtue of the variability of the collective of developers and the evolution of the program system itself. This aspect of systems programming considerably increases the requirement for documentability and understandability of its product.

Thus, to properly define the descriptive facilities that systems programming languages should possess, one has to consider that systems programming products are:

tools in some technology of problem solving;
multicomponent systems of programs;
creators of programs or realizers of execution;
elements of a standard cycle of processing of information; and the results of a multiyear effort by a changing collective.

2. Class of Languages Being Considered. Programming languages may be divided into three types according to their function:

Languages oriented to specification. These languages are intended primarily for fixation of the problem itself, and not the algorithm of its solution; described in them is /what/ has to be executed rather than /how/ it is executed. [//=underlined] Such languages are the so-called languages of a very high level (for example, SETL) or languages for formulating a problem for specialized intellectual packages of applications programs.

Languages oriented to algorithm description. The purpose of these languages is fixation of an algorithm in some abstract form not associated with the specifics of realization. Examples are the widespread high-level algorithmic languages (from FORTRAN to ALGOL-68). Of course the semantics of execution in these languages may be described by some abstract machine; however, in level of concreteness, an ALGOL-machine or PASCAL-machine is significantly different from a real computer.

Languages oriented to realization. Reflected in these languages are the concepts and properties associated with concrete realization such that a program in these languages already contains in an observable form the necessary details of realization. The simplest examples of these languages are the mnemonic codes (ASSEMBLER languages); the more developed languages of this type are the so-called high-level machine-oriented languages (MOYaVU), which we will discuss later.

Of course it would be more precise to speak of the orientation of the given descriptive facilities, and not of the orientation of the language as a whole, since there may be intertwined in a concrete language both facilities oriented to specification as well as facilities oriented to realization. Thus, if one intentionally limits himself in using the possibilities of the SETL language, the resulting program will differ little from an ALGOL program, and the designation of the number of bits or bytes in a value, as a data type characteristic, for example, in PL/1, or use of mnemonic code statements in AL'FA-6, predetermines the realization. At the same time, the determining set of descriptive facilities always allows us without any great question to place a given language in one of the types mentioned above.

On the other hand, for a certain class of problems or area of application, we may speak of /languages of realization/ as those languages in which a final program product is created (in contrast to /publication languages/, in which a description of a problem or algorithm is fixed with subsequent nonautomatic conversion of this description into a final program product). Thus, it is well known that very high level languages are used as realization languages for experimental artifical intelligence systems; however, more often, these languages are used as publication languages, as an intermediate form, which is then (after debugging and finishing) manually rewritten into some "lower" level language; ALGOL-68 is sometimes used not as a realization language, but as a language in which there is fixed a preprogram description of algorithms and data (at the level of technical design, for example); in the majority of cases for problems associated with computational mathematics, suitable languages for realization are the high level algorithmic languages, since the specification of realization is often not so important for computational problems.

In this paper, we intend to discuss what should be the linguistic tools of systems programming based on the current, knowledgeable systems programming users. Let us note at the ouset that we shall not be discussing languages of specifications that may be used in some stage of development of a program product for experiments with algorithms under development. Likewise, we shall disregard specialized languages that are used for a relatively narrow class of problems of systems programming, for example, as languages for describing certain characteristics of input languages in systems for constructing translators. Of course, these as well as other languages are useful tools; however, we shall dwell only on general-purpose realization languages for systems programming.

We will not conceal that it seems to us that systems programming realization languages of necessity are languages oriented to realization. An attempt to take as a realization language--just as for a number of areas of applications programming--a high level machine-independent language cannot be recognized--in any case for the current level of understanding of the problems of both languages and systems programming--as satisfactory in all respects. The combination of machine orientation and high level descriptive facilities in realization languages is emerging as a quite successful answer to current systems programming problems--this thesis was defended by the author in his work [2]. A similar thesis was substantiated earlier by W. Wulf in his work [3].

3. Necessity of Machine Orientation and High Level. For a long time, the primary tools for systems programming were assemblers and macroassemblers (and even now they are used quite extensively for the purposes of systems programming). While providing the capability of creating efficient programs, these machine-dependent systems essentially offered no transferability, made the process of programming itself complicated and offered few possibilities for understanding the text of a program. These shortcomings stood out quite saliently when compared to high level languages, from which assemblers differed above all by the lack of a clearly visible program structure and poor semantics of permissible information objects. Macroassemblers did not change the situation very much since the particular great compactness of the programs (compared to assemblers) made them neither clear nor natural--to the extent achieved in high level languages.

The attractiveness of high level languages in these respects gave impetus to their being used as systems programming languages. Attempts were made to use even FORTRAN and ALGOL-60, but since their utility for general purposes is not very great and they are oriented primarily to problems of computational mathematics, extensions of them were used as systems programming languages. The emergence of substantially more general-purpose high-level languages, such as PL/1 and PASCAL in particular, yielded more useful examples of their use as real systems programming languages (see the analysis of the possibility of using these languages as tools for systems programming, for example, in work [4]). But despite the number of successful applications of these languages for systems programming needs (what is meant here is not their use for purposes of specification and finishing of algorithms, but for obtaining a final program product, i.e., as realization languages), they did not substantially crowd assemblers and macroassemblers in systems programming. It seems to us that as realization languages, the high level machine-oriented languages in which there is a rational combination of both a number of the descriptive facilities of the high level general-purpose languages and the specific possibilities of machine dependency that characterizes assemblers and macroassemblers have become their real competitors.

Let us note first of all that the concept of machine orientation does not mean so fundamental a machine dependency and so fundamental a connection with the format and composition of machine instructions that there is in assemblers. Machine orientation includes the semantic association of language concepts with such characteristics of concrete machines as the structure and length of a machine word, the system of addressing, the availability, methods of use and function of registers and the like. Machine orientation of a language, therefore, does not preclude some possibilities of transferability of programs written in it, and we will later discuss the possible methods of transferability. We shall also include in the concept of machine orientation a number of semantic links not with the machine itself, but with the operating system--with the organization of channel operation, syncronization and so on.

One of the determining arguments in favor of machine orientation of systems programming languages is the criticality of the requirement for efficiency on its products. For systems programming problems, not only selection of the best algorithm is important, but also thorough realization of it, using all the capabilities and features of a given machine and its operating system. As the experience of using high level machine-oriented languages in systems programming problems shows, they yield in this respect essentially the same capabilities as assemblers (but a similar comparison of assemblers with high level general-purpose languages, as a rule, is not favorable to the latter). Taking the concrete features of a machine into account when formulating algorithms by using machine-oriented languages enables the program originator to control the thoroughness of realization. It is important to note that, as a rule, there is a quite clear correspondence between the concepts of these languages and their machine realization, while in high level general-purpose languages, the semantics of input constructions is formulated in terms remote from the semantics of constructions of the object program.

While for a number of applications areas high quality of object programs and good consideration of the features of realization can be achieved through automatic optimization of programs, the methods of optimization are still insufficiently developed for systems programming. Although the only well known optimizing programming system for machine-oriented systems programming languages--for the BLISS language [5]--uses a quite developed optimization technique, in our view it does not remove the need for clear concern for realization, but only eases this concern somewhat. Methods of formal description of machine-oriented optimization now under development based on realization of BLISS may make some progress in this area. Let us note that global optimization rather strongly distorts correspondence between input and output constructions which may be essential for systems programming and as a rule unimportant for applications programs.

It must be said that an argument for machine orientation of no little importance is the so-called "internal" machine orientation of algorithms for systems programming--they are often not only in the form of expression, but also in producible work associated with some concrete machine, and "know" its characteristics and its language. Translation algorithms are like that for example--in any case with respect to the object code generation stage. A rather complete reflection of the various characteristics of a machine, channels and external devices is also contained in many algorithms for operating systems. In comparing systems programming with applications, one may, though somewhat roighly, note their difference in the aim of applications programming being to develop a description of an algorithm for some processing of information, and in the aim of systems programming being to create a description of a program that has to be executed in some concrete environment--on a specific computer, within the bounds of a specific operating system, etc.

We will put into the concept of high level of languages the natural format of statements not associated with the format of machine languages, and the development of facilities of the task of control structures in the program to adequately develop the semantics of information objects and other linguistic capabilities that we will discuss later--i.e, namely those facilities that provide facilitation of programming and understandability of programs, and that means--the documentability and modifiability and other requirements on the program product of systems programming not being provided by assemblers and macroassemblers (as Wirth said: "One of the aims in developing PL-360 was to inspire the user with the capability of writing clearly and understandably while permitting full use of the 360 system equipment features" [12]).

With respect to systems programming realization languages, raising the level of the language does not lead to substantial acceleration of the process of programming proper and does not reduce the requirement for programming skill of the developer as occurs for applications programming since this rise is not accompanied, as a rule, by removal from the peculiarities of realization, form machine orientation, from concern for realization and representation of data. High level machine-oriented languages require good programming skills and the ability to express in their terms the realization approaches, but for the programmer with /high/ skills, they enable making the program clear, readable, understandable and modifiable, reducing the number of mistakes and facilitating debugging and verification.

4. Two Generations of High Level Machine-Oriented Languages--Brief Review. Just as for high level algorithmic languages, for which the isolation of a first (FORTRAN, ALGOL-60, COBOL, etc.) and second (PL/1, SIMULA-67, ALGOL-68, PASCAL, etc.) generation is natural, one can pick out two generations of languages for high level machine-oriented languages. In general, the emergence of high level machine-oriented languages sufficiently usable for systems programming needs dates back to the second half of the sixties. First generation language examples were ALMO [8], EPSILON [10] and PL-360 [14].

In ALMO there are a number of important features of machine orientation with overall machine independence. These features are, for example, selection of types of storage and among them--index registers, and assignment of sizes of values in terms associated with the size and dividing of machine words. Some unnaturalness of constructions of ALMO (reverse Polish notation of expressions, for example) was dictated by the language being intended not only in particular for description of the translation process itself, but also for an intermediate language of translation.

The semantics of statements and objects of EPSILON were consciously and openly interpreted using a realization expression by constructions of machine language--so-called images. In part the semantics in EPSILON are associated with types of statements and with assigned structures, but full semantic definiteness is assigned by the image. Thus, a statement of the form

a[i]: = a[i] + 1

in EPSILON is interpreted as a change per unit of value of some position of a machine word if this value had been earlier derived as a positive whole number. But if this is not so, then the execution of this statement could be attended to only from the image compared to the replacement operator and selection of a[i]. Values in EPSILON do not have types, i.e., interpretation of a value is fully determined by operations. Control statements in EPSILON are primitive and limited to conditional and unconditional jumps (almost as in FORTRAN). Data structures are associated with the sequence of machine words, divided perhaps into syllables of arbitrary length.

PL-360 had advanced facilities for control structures and quite good correspondence of types of variables to formats of machine representations--precisely the features that served as the basis on which a number of production tools were developed for systems programming for various machines. Just as EPSILON, PL-360 permits easy establishment of correspondence between constructions of the language and their realization--thus statements for repetition and selction as a control variable may be only that which has been correlated with a register; in the permitted types of values (byte, short integer, integer, real, long real), the standard formats and types of the IBM 360 machine language are easily attended to. Expressions in PL-360 (just as in EPSILON) are interpreted as notation of corresponding instructions--they do not contain parenthesis hierarchy and precedence of operators, to the extent that execution of the statement

R1: = R2 + R1

(R1 and R2 are designations of registers) is interpreted as the execution of the sequence

R1: = R2; R1: = R1 + R1.

PL-360 data structures are limited to linear sequence of the elements of the format indicated.

The BLISS language [6], which emerged at the very start of the seventies, is an example of second generation languages--it contains new, with respect to the preceding languages, capabilities as advanced forms of expressions and flexible capabilities for introducing data structures. Just as in EPSILON, BLISS values have no types, and in values one can consider values formed by a group in succession of the arranged positions of a machine word (but not as a described syllable structure). Expressions (and not statements) are in BLISS the basic construction of the language which coincides with machine orientation in the sense that calculation with an expression retaining value may be represented by remanent states of the machine registers (the adder, in particular). BLISS has advanced control statements. An important BLISS feature is the description of the access function. Thus, a description of the form

struct ar2[i, j] = (.ar2 + (.i - 1)x10 + (.j - 1))

associates with the name ar2 the actions corresponding to locating the address of the element of the array with 10 columns (the period before the identifier means taking the value according to the address associated with the identifier--in BLISS taking an address and taking a value according to an address are syntactically different). If after this description we introduce a description of the form

Own x[100], y[100], z[100];

map ar2 x : y : z;

which define for each of x,y,z the allocation in sequence of 100 machine words and association with the structure (access) ar2, then the statement

x[.k, .l] - .y[.l, .k]

assigns the transfer of the value from the element with indexes l, k of array y to the element with indexes k, l of array x. Although BLISS was designed for the PDP-10 and contains elements of some machine dependence (calculation with specific length of a word, consideration of the existence of registers and the like), these elements are not very significant--there are versions of BLISS for the PDP-11 and the IBM series.

The YaRMO language [15], developed for the BESM-6, took a number of the important features of BLISS such as the expression as the basic construction of the language, described functions of access, control facilities associated with structured programming, etc. Operating with data structures underwent further development in YaRMO--in addition to structures with a describable function of access in the language there emerged notations that develop syllable partitioning of a machine word of the EPSILON language; in contrast to BLISS, one may have for the same structure a different function of access and function of notation [16]. Intermodular associations are described more flexibly than in BLISS.

In the LIS language [17], machine orientation and machine dependence are concealed completely. Outwardly, this language looks like a high level language oriented to algorithm description. At the same time, there are syntactically isolated fragments for description of realization of data representation in which different facilities of indication of machine dependency are permissible. But access to data elements does not depend [18] on the realization of their presentation.

The development of systems programming languages has still not found descriptive facilities adequate for a number of practical needs--in any case, to the same extent as development of general-purpose algorithmic languages for the needs of applications programming. Let us discuss some properties of realization languages important to systems programming and try to point out these unresolved problems.

5. Data Structures and Their Realization. Flexibility of operation with data structures is a major requirement for systems programming languages. Of course this requirement is common with respect to modern programming languages; however, the specific nature of systems programming lies in that not only the wide range of information structures must be considered in them, but also the peculiarities of their machine representation. On one hand, there should be in the language an adequate set of types of values, and assigning types to objects of the program unquestionably provides good capabilities for statistical verification of semantics and flexible understanding of operational signs; on the other hand, we should, if we want to, control machine representation of values and sometimes have the capability of interpreting these machine values of variables a different way (for example, sometimes as bit, sometimes as symbol, sometimes as integer). As for data structures and set of types, one of the most successful languages in this respect as applied to systems programming is PASCAL; the various examples of its application to systems programming problems indicate the sufficient naturalness of these external facilities. Among the successful features of PASCAL are the capability of assigning a system of coding for objects of the scalar describable type and of introducing notations with variants. Assignment of interval types and describable scalar types in essence also provides certain determination of the realization of representation of values and arrangement of elements of notation in a machine word. Further development of PASCAL notations in LIS led to inclusion in notation as elements of procedures of operation with components of notation.

It is also traditional in the practice of systems programming to treat one and the same binary value as a value that may be referred to different content types and generated by various formats of a machine word. This also causes lack of standardization in realization languages such as EPSILON, BLISS and YaRMO. In limited cases (e.g., treatment of references as integer), this also occurs in LIS. The variety of formats that may be "put" on one and the same value is effected by describable partitioning into syllables in EPSILON and YaRMO.

In PASCAL, the assignment of a system of coding does not determine the binary values of codable values of the scalar type. LIS introduces (only in the realization part) the capability of explicitly assigning coding binary values. In EPSILON, one can introduce not only explicit representation of coding values, but also of classes of machine representation--so-called codes, and a set of values for which there is assigned either a range of binary values or identifying binary features in specific positions of the machine representation.

. . . construction of a program with detailing development even in BLISS one has to implement not automatically, while such actions could be automated.

Along with detailing development, a popular method of providing segmentability when developing complex programs is modular programming. It also serves as the basis for development of packages and complexes of programs. The fine analysis of the problems of developing modular programs, made by Goos and Kastens [23], showed that the majority of programming languages do not have convenient capabilities for expressing a modular structure and intermodular associations. This may be attributed to the primitiveness of existing rules for localization in languages not providing the needed flexibility in providing a link between the module and the environment. In building both a hierarchy of modules and a family of modules, we have to have convenient facilities for specifying an interface with supermodules and submodules and with adjacent modules in a set; for this, one has to regulate protection and access to local and global objects (data and procedures). It must be said that many of the realization languages for systems programming inadequately support modular programming just as high level algorithmic languages--in BLISS, for example, intermodular associations are organized, just as in conventional mnemonic codes, by the indication of global and external objects. Some progress was made in languages such as LIS where the unit of program structure consists of a data segment and a program segment which allows the entire interface to be grouped in the data segment, and YaRMO special descriptors [24] are introduced to indicate association with the external environment and to control access to local objects.

In systems programming practice, dating from the appearance of macroassemblers, there arose the technological approach of the introduction of problem orientation with the help of the accumulation and use of a macrolibrary. In one way or another, there are macro facilities in systems programming languages--at the level of simple macros of BLISS or the open procedures of EPSILON, as the basis for describable functions of access in YaRMO, and finally, as the basic construction of the language with free format of representation in SIGMA. Despite the availability of general-purpose macroprocessors, inclusion of macro facilities in systems programming languages remains advisable both for increasing the rate of macroprocessing (which must not be disregarded with intensive use of macros) and for improving the clearness and verifiability of results of macro substitution (for which the macro facilities must be tied to the syntactic structure of the language, for example, as in EPSILON where macro facilities for description and use are similar to procedures). Let us note that due to the desire to discipline machine dependency and to combine it with facilities of high level languages, it is natural to limit toleration of the use of constructions of machine language or mnemonic code, customary for a number of systems programming languages, only to texts of macro definitions.

We have already mentioned that semantic associations with the concepts and objects of a concrete operating system should also be included in the concept of machine orientation. Along with the desirability of this orientation, one would like to build appropriate constructions of a language by a method rather independent of a concrete operating system. This is one of the most unresolved problems of systems programming languages since some unification of the concepts of operating systems is required for this. And this unification could be built based on an essentially efficient, flexible and at the same time rather abstract model of an operating system, but for now there is no such generally accepted model. The concepts of such a model have been discussed from the point of view of systems programming languages in a number of works (for example, [25] and [26]). The problem is to include in this model sufficiently precisely, flexibly and universally the semantics of concepts such as, for example, processes, resources, communication, priority queues, monitors, properties of these concepts, and operations with objects of this sort.

It is rather difficult to formulate some complete set of descriptive facilities for an evolving field such as systems programming. Let us mention Chinin's work [24] which is based on the actual experience of using and developing the YaRMO language and which effectively expresses our current understanding of systems programming problems.

7. Transferability in Systems Programming. We have already noted that the concept of machine orientation is distinct from the concept of machine dependency. In principle, this makes it possible to suggest a requirement for systems programming languages such as ensuring transfer of software from one machine to another. Of course, a requirement for transferability is not absolute with respect to those systems programming algorithms in which definite machine dependency has been incorporated. For example, in the transfer of a translator for some machine-independent language, one may speak of the possibility of transferring all the translation algorithms except the algorithm for generation (with the usual approach to translation), as well as possibly some parts of the algorithm associated with a concrete operating system. Efficiency may require changing the machine representation of data with the transfer.

Although fully automatic transfer of programs in machine-oriented systems programming languages is not always possible, nor even always desirable, it could nevertheless be automated under some limited conditions. An approach to automation of the transfer of programming systems, based on an intermediate language, is offered in the ALMO system [8]. The ALMO machine-oriented language is a language of some abstract computer approximating a number of modern computers in its properties. It is assumed that there exist programming systems written in ALMO and translating from input languages not into an object language of a concrete machine, but also into an ALMO program. In this translation scheme, ALMO acts as an intermediate language--translation occurs in two steps, first into ALMO, and then from ALMO into object code (existence of a translator from ALMO for a given machine is assumed). In the transfer, a new translator from ALMO is created and all remaining software is transferred automatically. To make the programs being created more efficient where possible, ALMO has facilities for assigning certain characteristics of a concrete computer.

The SIGMA system [20, 21] has substantially more advanced facilities for describing both the characteristics and the object code of a concrete computer. According to the author [21], the language is a machine-oriented open language. The open nature derives from its set of parameters, the fixation of which determines the representation of the language oriented to a concrete computer. In the SIGMA system, a description of a concrete computer, along with a program to be translated, is the source data; to transfer the output program to a new computer, one need only compile its description according to the rules of SIGMA (let us note that only input programs are being transferred this way, and the translating or instrumental machine remains the same).

Assigned in parameters concretizing a computer are: length of word, its division and filling with retention of values, parameters of addressing, and so on, as well as formats for machine representation of various lists. The machine language proper is specified by masks and rules for filling them with values of parameters of macros. Experience shows that the facilities selected for description of a computer permit the description of an extensive class of machines--various machines such as the M-20, BESM-6, YeS computer and Hewlett-Packard 2000 have been described.

Great and essentially sufficient possibilities of transferability can be obtained in machine-oriented languages through organizational measures--by observing certain discipline during programming. The experience of using EPSILON, for example, indicates this. The required programming discipline is based on the machine-dependent and machine-independent aspects of concepts being especially differentiated in EPSILON [12]. Differentiated in EPSILON are standard facilities, the semantics of which are the same for various computers, and a concrete level of linguistic facilities definable for a given computer (there are different realizations of EPSILON for computers of the M-20, BESM-6 and Minsk-22 type--see [11, 13]). If we wish to maintain compatibility of algorithms with different computers, we provide beforehand for the use of machine-dependent facilities in fixed locations of the algorithm (it is desirable to group this usage in procedures or macros). Recommendations, similar in spirit, that machine-dependent details of an algorithm should be limited to interlacing of parameters of machines, in the set of which transfer is planned, have been made based on the use of ALMO in work [9].

In work [4], Goos rightly notes that as a rule data representation is machine-dependent, but the algorithm itself used to process it can often be expressed in a machine-independent manner. Then separating the algorithm description into the data description and the description of the algorithmic part proper enables the originator to transfer the algorithm, changing only the data representation, which in BLISS is provided by changing the description of the access function, and in LIS by changing the realization part in the data segment.

And so, despite the machine orientation of realization languages for systems programming, the experience of their use has built up a number of linguistic and organizational facilities for providing transferability of the program product with retention of both clearness and efficiency.

BIBLIOGRAPHY

1. Seegmueller, G. "Systems Programming as an Emerging Discipline," Information Processing 74, IFIP-74, North-Holland Publishing Company, 1974.

2. Pottosin, I. W. "Systemprogrammiersprachen. Beitraege zur Informationsverarbeitung" [Systems Programming Languages: Contributions to Information Processing], BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1977.

3. Wulf, W. A. "Issue in Higher-Level Machine-Oriented Languages. Machine Oriented Higher Level Languages," ed. by W. L. Van der Poel and L. A. Maarssen, North-Holland Publsihing Company, American Elsevier, 1974.

4. Goos, G. "Language Characteristics. Programming Languages as a Tool in Writing System Software. Lecture Notes in Computer Science," in Vol 30, "Software Engineering. An Adavanced Course," Springer Verlag, 1975.

5. Wulf, W. A.; Johnsson, R. K.; Weinstock, C. B.; and Holls, S. O. "The Design of an Optimizing Compiler," Computer Science Department Report, Carnegie-Mellon University, 1973.

6. Wulf, W. A.; Russel, D. B.; and Habermann, A. N. "BLISS: A Language for Systems Programming," CACM, Vol 14, No 12, 1971.

7. Wulf, W. A. "Programming without go to," Information Processing 71, IFIP-71, North-Holland Publishing Company, 1971.

8. KAMYNIN, C. C., and LYUBIMSKIY, E. Z. "An Algorithmic Machine-Oriented Language--ALMO," in the collection "Algoritmy i algoritmicheskiye yazyki" [Algorithms and Algorithmic Languages], VTs AN SSSR [Computing Center, USSR Academy of Sciences], issue 1, 1967.

9. BOGDANOV, V.V.; YERMAKOV, YE. A.; and MAKLAKOV, A. V. "Programmirovaniye na yazyke ALMO" [Programming in the ALMO Language], Moscow, Statistika, 1976.

10. POTTOSIN, I. V.; RAR, A. F.; KATKOV, V. L. "EPSILON--System for Automation of Programming for Symbolic Processing Problems," in "Trudy I-y Vsesoyuznoy konferentsii po programmirovaniyu" [Works of First All-Union Conference on Programming], Kiev, 1968.

11. KATKOV, V. L.; MOROZOV, V. P.; POTTOSIN, I. V.; RAR, A. F.; SEMENOVA, L. YA.; and KHOPERSKOV, A. YE. "EPSILON--sistema avtomatizatsii programmirovaniya zadach simvol'noy obrabotki" [EPSILON--System for Automation of Programming of Symbolic Processing Problems], Novosibirsk, Nauka, 1978.

12. Pottosin, I. V., and Rar, A. F. "Application of the Machine-Oriented Language EPSILON to Software Development."

13. SVETLAKOVA, F. G. "The MACROEPSILON Machine-Oriented Programming System," in "Trudy VI konferentsii KEVM BESM-6" [Works of the Sixth Conference on the KEVM BESM-6], Tbilisi, 1976.

14. Wirth, N. "PL 360, Programming Language for the 360 Computers," JACM, Vol 15, No 1, 1968.

15. GOLOLOBOV, V. I.; CHEBLAKOV, B. G.; and CHININ, G.D. "High Level Machine-Oriented Language for the BESM-6 Computer," in the collection "Razvitiye programmnogo obespecheniya BESM-6" [BESM-6 Software Development], Moscow, VTs AN SSSR, 1975.

16. CHEBLAKOV, B. G. "Data Structure Representation in a High Level Machine-Oriented Language," in "Trudy Vsesoyuznogo simpoziuma po metodam realizatsii novykh algoritmicheskikh yazykov" [Works of the All-Union Symposium on Methods of Realization of New Algorithmic Languages], Novosibirsk, VTs SO AN SSSR, 1975.

17. Ichbish, J. D.; Heliard, J. C.; Rissen, J. P.; and Cousot, P. "The System Implementation Language LIS," CII, Technical Report 4549 E/EN, Dec. 1974.

18. Ichbian, J. D.; Rissen, J. P.; Heliard, I. C. "The Two-Lever Approach to Data Independent Programming in the LIS System Implementation Language."

19. Ershov, A. P., and Rar, A. F. "SYGMA, a Symbolic Generator and Macroassembler," in the collection "Symbol Manipulation Languages and Technique," North-Holland Publishing Company, 1968.

20. Stepanov, G. G. "SYGMA, a Machine-Oriented Higher-Level Language."

21. STEPANOV, G. G. "Technique of Describing Computers for Adjustable Macroassemblers," in "Trudy Vsesoyuznogo seminara po voprosam makrogeneratsii" [Works of the All-Union Seminar on Questions of Macrogeneration], Tbilisi, VTs AN Georgian SSR, 1975.

22. Liskov, B.; Snyder, A.; Atkinson, R.; and Schaffert, K. "Abstraction Mechanisms in CLU," COMM. OF THE ACM, Vol 20, No 8, Aug. 1977.

23. GOOS, G., and KASTENS, U. "Yazyki programmirovaniya i postroyeniye modulyarnykh programm. Sozdaniye kachestvennogo programmnogo obespecheniya" [Programming Languages and Construction of Modular Programs. Development of Qualitative Software], Vol 1, Novosibirsk, VTs SO AN SSSR, 1978.

24. CHININ, G. D. "Yazykovyye urovni bez poteri effektivnosti. Sozdaniye kachestvennogo programmnogo obespecheniya" [Linguistic Standards without Loss of Efficiency. Development of Qualitative Software], Vol 1, Novosibirsk, VTs SO AN SSSR, 1978.

25. Hibbard, P. G., and Leng, P. H. "The Semantics of High Level Systems Programming Languages."

26. Brinch Hansen, P. "A Programming Methodology for Operating System Design."

[8044/0004 A-8545]

8545
CSO: 8044/0004 A

UDC 681.326

SOFTWARE PACKAGE FOR INTEGRATED MODELING OF MAN-MACHINE SYSTEMS WITH LIMITED TIME RESOURCE

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 10-15

[Article by Vyacheslav Vasil'yevich Kondrat'yev, doctor of technical sciences, GGU [Gor'kiy State University], and Aleksandr Sergeyevich Tsar'kov, engineer (Gor'kiy); received by editor 4 Apr 79 (after revision, 28 Jun 79)]

[Excerpts] In the early stages of designing automated systems, general purpose methods and means are needed to find prognostic estimates of their efficiency. However, in using traditional methods of modeling with mock-ups and static or dynamic test benches, finding such estimates becomes possible only at the detail design stage. The basic trends in solving this problem are currently linked to the development of methods of mathematical modeling of complex systems [1, 2]. In the process, simulation tests on computers are used as mathematical analogs of natural tests performed on test benches. This makes it possible to obtain system estimates of efficiency with a sufficient degree of prognostication.

The literature [2-4] reveals specific mathematical models, developed for these purposes, of technical elements of man-machine systems (ChMS) called mathematical test benches (MS). Let us note in particular the AVSIM test bench for studying the process of selecting and evaluating parameters of digital electronic systems [3], as well as the General Dynamics bench for comprehensive modeling of on-board equipment of aerospace apparatus [4].

Building mathematical test benches raises the problem of building unified mathematical models of a natural test that allow study of the interaction between the human operator and the controlled object taking external environmental factors into account.

However, practical realization of this sort of integrated model involves difficulties caused by the need to unite different formalisms within a single description, as well as by selecting an adequate model of the human operator as the basic element in the system [5-7]. There are also unsolved problems linked to the specific nature of application of integrated models in the early stages of design and with the methods of their realization on computers.

In this work, we describe the organization of an integrated model of a man-machine system operating under the conditions of limited time; the principles of realization of this model in the form of an information and computing package of programs are formulated and its structure is treated. Man-machine systems with limited time resource may include a quite broad class of transportation, power, radio engineering and several industrial chemical systems. Special attention is paid to forming and realizing a strategy of operating the complex oriented to deriving real-time data on system status at arbitrary moments of time and finding integrated estimates of their efficiency.

The first version of the package was realized on the "Minsk-32" computer and the MIR minicomputer. It was programmed in ASSEMBLER, FORTRAN and ALMIR. The subsystems for OMA [maintenance of machine tests and file] and UIM [control of tests with simulation model] contain 28 program modules, of which 9 belong to the "Minsk-32" SMO [software system] library. The higher level UIM subsystem is written in ASSEMBLER (YaSK [expansion unknown]) and contains 1,900 statements in this language. In this version, the package of programs has been used to find prognostic estimates during development of a number of transportation and navigation systems.

[408-8545]

8545
CSO: 1863

UDC 681.3.48./14

SOLVING THE MAIN PROBLEM OF EMULATION

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 24-28

[Article by Aleksandr Vasil'yevich Palagin, doctor of technical sciences, IK AN USSR [Institute of Cybernetics, Academy of Sciences, Ukrainian SSR] (Kiev); received by editor 13 Dec 79]

[Excerpt] Conclusion. The structures of the control code converter discussed have broad functional capabilities which make it possible to use it effectively in analysis of codes of control words, i.e., for solving the problem of emulation. From the description of the UPK [control code converter], it follows that in addition to its immediate function, it can also serve as a general-purpose logic element performing the operations of normalization, shifting, switching and conjunction. The formal description and quantitative estimates of the alternatives of structural realizations of the UPK make it possible to soundly approach the design of the structure of the emulating processors.

[408-8545]

8545
CSO: 1863

UDC 681.3.06:519.2:65

PROGRAM PACKAGE TO ORGANIZE OPERATION OF A MULTIPLE-COMPUTER COMPLEX OF SMALL COMPUTERS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 85-88

[Article by Yuriy Miroslavovich Korostil', engineer, VISP [All-Union Institute of the Welding Industry] (Kiev); received by editor 5 Jun 79 (after, revision 3 Aug 79)]

[Excerpts] Small computers are widely used now. The demand for these machines is growing as a result of their successful application in solving problems of a particular class. In connection with this, many users now have several small computers. In these situations, it is more advisable to operate them as a multimachine computing complex.

One of the substantial difficulties is the need to devise software to organize the work of the combined small computers. This work describes the program package developed with the M6000 computer processors that overcomes this difficulty.

Limitations of the Described Program Package. This version of the proposed program package allows description of up to 16 external devices for each processor in the system. Resource outlays for each of the computers in the system when using the proposed program package are expressed in allocation of a certain volume of main memory. This version of the MMS [interprocessor communication module] takes from 3 to 4K bytes. The time for satisfying a resource request when two computers are communicating is 1.5 to 2 ms.

[408-8545]

8545
CSO: 1863

UDC 681.3.06:621.317.799

HIGH-LEVEL LANGUAGE FOR INSTRUMENTATION WITH BUILT-IN MICROPROCESSORS (MICROCOMPUTERS)

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 44-46

[Article by Valeriy Viktorovich Barashenkov, doctor of technical sciences, LETI [Leningrad Electrotechnical Institute imeni V. I. Ul'yanov (Lenin)] (Leningrad); Nadezhda Tsanzhiyevna Bil'gayeva, graduate student, LETI (Leningrad); Aleksandr Filippovich Kazak, candidate of technical sciences, LETI (Leningrad); and Aleksandr Orestovich Timofeyev, candidate of technical sciences, LETI (Leningrad); received by the editor 24 Jul 79 (after revision, 30 Oct 79)]

[Excerpt] Described below is the system developed by the authors while designing meter generators with built-in microprocessors and microcomputers for programming in the various generators of this class. The system includes the language of the System of PRogramming with INTerpreter (SPRINT), and the interpreter from the SPRINT language into the internal language and the set of service programs (program-coder and debugging programs).

[408-8545]

8545
CSO: 1863

UDC 681.3.06:51

USING A HIGH-LEVEL LANGUAGE FOR MICROPROGRAMMING OF MICROPROCESSOR DIGITAL DEVICES

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 34-38

[Article by Sergey Aleksandrovich Mayorov, doctor of technical sciences, LITMO [Leningrad Institute of Precision Mechanics and Optics] (Leningrad); Aleksandr Valentinovich Marin, st. n. s. [senior scientific associate], NII [Scientific Research Institute] TEZ [expansion unknown] imeni M. I. Kalinin (Tallinn); and Vladimir Ivanovich Skoribskiy, candidate of technical sciences, LITMO (Leningrad); received by editor 19 Mar 79 (after revision 25 Jun 79)]

[Excerpts] Without dwelling in detail on the features, merits and drawbacks of the various microprocessors and single-chip microcomputers already covered extensively in the literature (for example, [1]), let us mention the microprocessor sets with addable capacity. These sets (series K584, K587 and K589) make it possible to design the structure of a digital device in which the economic and technical requirements are optimally coordinated. Their major feature is the microprogram principle of control. In the simplest controllers, microprogramming is a uniquely available method of control for the user and resembles programming on the surface, although the user has to have more detailed knowledge of the structure of the device than with programming. In designing these digital devices, of major importance is automation of the preparation of the micro-software which is a different task from that of computer software.

Existing programming languages of a high-level are not suitable for describing algorithms interpretable by microinstructions of a digital device. The known modifications of these languages (PLM, PLZ, PL/W at al.) [2], which allow for some features of the single-chip microcomputer instruction systems, are also not always convenient and need further simplification, new limitations and specialization if they are to be used for microprogramming.

This work discusses the application of a structural-functional microprogramming language [3] and the features of the translating system that considers the structure of the microporcessor when generating sequences of microinstructions. In the process, a method of structural interpretation of microprogramming language expressions is used that has been realized in the form of a general-purpose procedure.

BIBLIOGRAPHY

3. Marin, A. V., and Skorubskiy, V. I. "A Structural-Functional Microprogramming Language," in "Razrabotka, ekspluatatsiya i razvitiye sistem avtomatizirovannogo proyektirovaniya REA" [Development, Operation and Evolution of Systems for Automated Design of Radioelectronic Equipment], Moscow, MDNTP [Moscow House of Scientific and Technical Propaganda imeni F. E. Dzerzhinskiy], 1978, pp 128-132.

[408-8545]

8545
CSO: 1863

DATA BANK WORKING GROUP SEMINAR HELD

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, 1980 signed to press 4 Jun 80 pp 136-317

[Article by Vil'gel'm Al'bertovich Krakht, candidate of technical sciences, computing center of the Polytechnical Institute (Tallinn)]

[Excerpt] The Working Group on Data Banks (RGBD) of the Interdepartmental Commission on Computer Software under the State Committee on Science and Technology of the USSR Council of Ministers held its regular seminar from 16 through 18 October 1979 in Pyarnu (Estonian SSR). The seminar theme was relational and network modeling, languages and problems of realization. Participating in the seminar were working group members and observers as well as specialists from various organizations engaged in development of data base management systems, 73 people in all. There were 17 reports made and discussed that dealt with various problems and systems pertinent to the seminar theme.

[408-8545]

8545
CSO: 1863

ABSTRACTS FROM THE JOURNAL 'PROGRAMMING'

Moscow PROGRAMMIROVANIYE in Russian No 4, Jul-Aug 80 pp 95-96

UDC 681.3.06

ALGORITHM FOR STRUCTURAL SYNTHESIS OF PROGRAMS

[Abstract of article by Tyugu, E.Kh. and Kharf, M.Ya.]

[Text] A description is given of the automatic synthesis of programs based on a proof of the existence of a solution which is computed by means of a program. Algorithms are presented for finding a proof of the existence of a solution for three different classes of problems. Estimates are given of the complexity of algorithms for synthesizing programs.

UDC 51 : 621.391

REDUCIBILITY TO FREE ROUTINES

[Abstract of article by Nigiyan, S.A.]

[Text] A class of routines is isolated which do not have equivalent free ones. It is demonstrated that the property of a routine to have an equivalent free one is effectively unrecognizable and that there does not exist an algorithm which would construct an equivalent free routine for each routine having one.

UDC 681.3.06 : 800.92

RELATION MODEL OF COMPUTATIONS

[Abstract of article by Kleshchev, A.S.]

[Text] A study is made of a class of Church-Rosser systems called a relation model of computations. A predicate calculation language is used as the notation. Theorems of correctness for certain classes of theories are proved. Examples are given of writing parallel algorithms in the form of systems of axioms, along with rules for converting KS [coupling constituent] grammars into relation programs for parallel syntactical analysis.

UDC 681.322

UTILIZATION OF TABLES OF SOLUTIONS IN CONSTRUCTING AN ALGORITHM FOR CORRECTING ERRORS

[Abstract of article by Smolyar, I.V.]

[Text] A discussion is presented of writing an algorithm for correcting errors in communications of fishing vessels regarding the results of fishing (coordinates and catch), using the language of tables of solutions (TR's), and of using methods of Boolean algebra for translating TR's into a procedural form.

UDC 681.142.2

Q-TECHNOLOGY FOR STRUCTURAL PROGRAMMING

[Abstract of article by Antonets, D.K., Gladkov, V.P. and Solov'yev, A.Ye.]

[Text] A description is given of a technology for constructing structural programs based on the idea of a chained list. Standard routines are used to lengthen the itemization step. This approach makes it possible to improve the readability of structural programs.

UDC 681.142.2 : 518.5

UNIVERSAL TIME SHARING ALGORITHM

[Abstract of article by Silin, I.N. and Fedyun'kin, Ye.D.]

[Text] An algorithm is suggested which provides advantages for dialogue problems utilizing little memory capacity and machine time while maintaining time sharing among all problems according to external priorities, with sufficiently complete loading of the processor (or processors). The harmful competition among problems in reallocation of the memory is eliminated.

UDC 681.3.06

STRUCTURE OF THE FAP-KF PACKAGE OF PROGRAMS FOR GEOMETRICAL MODELING

[Abstract of article by Gorelik, A.G.]

[Text] The functional capabilities and structure of the FAP-KF package of programs designed for the automation of geometrical modeling and engineering graphic work are discussed. With regard to the user, the FAP-KF package represents an expansion of the Fortran language with geometrical variables and operations. The package has been designed for YeS [Unified System] computers.

UDC 681.3.06

SYNTACTICALLY CONTROLLED TRANSLATION OF A TESTING ASSIGNMENT INTO A CONTROL PROGRAM FOR MICROCOMPUTERS

[Abstract of article by Borde, B.I. and Kuznetsov, V.L.]

[Text] A study is made of the structure of software for a system for automating the processing of the results of mechanical tests on the basis of microcomputers. It is suggested that the control program be formed by means of syntactically controlled translation of the testing assignment.

UDC 681.3.06

OPTIMIZATION OF FINDING DEFECTS IN TECHNICAL DIAGNOSIS UNITS

[Abstract of article by Maron, A.I.]

[Text] Questions relating to finding and eliminating defects in computer software and hardware are discussed.

UDC 681.2 : 519

EXPLANATION OF LACK OF PROCEDURE BY MEANS OF THE THEORY OF CATEGORIES

[Abstract of article by Vol'dman, G.Sh.]

[Text] An attempt is made, utilizing the theory of categories, to provide a precise formal definition for the concept of the semantic equivalence of interrogations, a set of program realizations of interrogations and their degree of lack of procedure.

UDC 681.322

PROGRAM SYSTEM FOR AUTOMATIC RECOGNITION OF IMAGES OF VARIOUS CONFIGURATIONS

[Abstract of article by Deyev, V.V. and Tarasov, B.V.]

[Text] Algorithms are presented for the automatic recognition of images, employing the apparatus of Fourier-Haar series for describing individual images and their standards, as well as the results of the recognition of visual images of various configurations.

UDC 681.3.06 : 51

STRUCTURE AND ORGANIZATION OF A PACKAGE OF APPLIED PROGRAMS FOR STATISTICAL PROCESSING OF DATA ON UNIFIED-SERIES COMPUTERS

[Abstract of article by Sergiyenko, I.V., Stukalo, A.S., Vagis, A.G. and Parasyuk, I.N.]

[Text] A description is given of the structure and organization of the SOD-GS package of applied programs, designed for solving a class of statistical data processing problems on unified-series (YeS) computers. Singled out as the basic software components of the SOD-GS are the input language, the monitor and the package's data base.

UDC 681.142.2

PROGRAM MODULES MAKING IT POSSIBLE TO SAVE PAPER

[Abstract of article by Brazaytis, Z.P.]

[Text] A description is given of a procedure for modifying program modules which form fields of print and which execute their output, as well as of the organization of special modules making it possible to print on two sides. Questions are discussed, relating to improving the effectiveness of the utilization of computer resources.

[46-8831]

8831
CSO: 1863

UDC 681.3

ONE METHOD OF COMPLEX NUMBER MULTIPLICATION IN SPECIAL-PURPOSE ELECTRONIC COMPUTERS

Kiev KIBERNETIKA in Russian No 3, 1980 pp 136-138

[Article by V. V. Kolomeyko]

[Excerpts] In solving many tasks of automatic control, hydroacoustics, digital filtration of signals, etc, a very widespread operation is that of multiplying a complex variable by a complex coefficient, the value of which is known in advance, as a rule. For example, in accomplishing an algorithm of spectral analysis of signals practically all the operations of multiplication are the multiplication of complex variables by complex coefficients of the type $W = \cos \phi_1 + j \sin \phi_1$, the values of which usually are determined in advance by means of the recurrent correlation $W_{i+1} = W_i W_1$.

In the article a method of complex number multiplication is examined, one which permits simplifying the operation of multiplication of a complex variable by a complex coefficient (by any complex number that participates in several operations of multiplication). The fact that a complex coefficient is used as a cofactor several times permits representing that coefficient in a form most convenient for the operation of multiplication and as a result simplifying performance of the operation of multiplication.

Naturally the effectiveness of the proposed method of complex number multiplication depends on the level of specialization of the computer accomplishing the given method and the class of problems solved by the computer. In special-purpose computers of the type of digital spectrum analyzers the application of the given method permits reducing the volume of calculations by about 10-15 percent.

BIBLIOGRAPHY

1. Singleton, R. An algorithm for computing the mixed radix fast Fourier transform. IEEE TRANSACTIONS ON AUDIO AND ELECTROACOUSTICS, 1969, No 2, pp 93-103.

2. Despain, A. Fourier transform computers using CORDIC iterations. IEEE TRANSACTIONS ON COMPUTERS, 1974, No 10, pp 993-1001.

3. Yelfimova, L. D., Kolomeyko, V. V., Krivoy, S. F., and Petushchak, V. D. One method of performing the BPF computing procedure. In the book: Avtomatizatsiya nauchnykh issledovaniy i tekhnicheskoy podgotovki proizvodstva (Automation of Scientific Investigations and Technical Preparation of Production). Kiev, 1975, pp 43-45.

4. Kolomeyko, V. V., Moroz-Podvorchan, I. G., and Petushchak, V. D. USSR Author's Certificate No 607213. Published in Byulleten' izobreteniy, 1978, No 18.

5. Morris, L. A comparative study of time efficient FFT and WFTA programs for general purpose computers. IEEE TRANSACTIONS ON ASSP, 1978, No 2, pp 141-150.

6. Semotyuk, M. V., and Voyun, V. P. Operating devices for summing paired products and multiplication of complex numbers. USiM, 1978, No 3, pp 87-92.

[49-2174]

2174
CSO: 1863

UDC 51:681.3.06

IMITATIONAL SIMULATION BY MEANS OF THE SYSTEMS NEDIS AND GASP-IV

Kiev KIBERNETIKA in Russian No 3, 1980 pp 35-50

[Article by T. P. Mar'yanovich, S. S. Azarov, V. V. Gusev, M. A. Sakhnyuk and A. V. Shemshur]

[Excerpts] At the present time by the efforts of a group of the International Association for Mathematics and Computers in Simulation (IMACS) single classification tables have been prepared for representation of software for computer simulation, which permits very completely and compactly describing the various systems of simulation and distinctive features of their realization and application.

In the article the NEDIS system is described by means of the presented IMACS tables, which makes it possible to acquaint the developers of simulation systems with the given method of describing them.

Table 1. General information

Abbreviation and full name of the simulation system	NEDIS System for continuous-discrete system simulation
Address of developers	Institute of Cybernetics, Ukrainian SSR Academy of Sciences, Section 160 142/144, 40th Anniversary of October Prospect, Kiev-207, 252207
Literature books	Glushkov, V. M., Gusev, V. V., Mar'yanovich, T. P., and Sakhnyuk, M. A. Programmnyye sredstva modelirovaniya nepreryvno-diskretnykh sistem (Programmed Means of Simulating Continuous-Discrete Systems). Kiev, Naukova dumka, 1975, 151 pages.
descriptions	Gusev, V. V., Mar'yanovich, T. P., and Sakhnyuk, M. A. Sistema programmirovaniya NEDIS (The NEDIS Programming System). Kiev, Institute of Cybernetics, Ukrainian SSR Academy of Sciences, 1975, Part I -- 250 pages, Part II -- 186 pages.
articles	Gusev, V. V., Mar'yanovich, T. P., and Sakhnyuk, M. A. Comparative analysis of means of programming the simulation of complex systems. USiM, 1973, No 1, pp 1-7.

[49-2174]

2174
CSO: 1863

APPLICATIONS

UDC 025.4.036:608.3

'VEKTOR-II' REGIONAL AUTOMATED SYSTEM FOR SELECTIVE DISTRIBUTION OF PATENT INFORMATION

Moscow NAUCHNO-TEKHNICHESKAYA INFORMATSIYA. SERIYA 2. INFORMATSIONNYYE PROTSESSY I SISTEMY in Russian No 7, 1980 pp 7-16

[Article by F. S. Bunimovich, V. P. Zakharov, M. G. Krasikova, and P. G. Mordovchenko]

[Excerpt] 1. Introduction

Within the framework of work on the creation of a unified patent information system the Soviet Union participates in the activity of the International Patent Documentation Center (INPADOC), which has unified efforts of 46 countries of the world in the transition to machine legible carriers of current information about patent documents [1,2,3]. In the INPADOC (EDT) data base, received weekly in the USSR, are reflected publications of the official bulletins of patent departments of countries participating in INPADOC on practically all inventions registered in those countries (over 800,000 documents per year).

That data base contains 13 basic elements of bibliographic description of stored documents, including the designation of the invention in the language of the original, the name of the inventor and the name of the applicant (the company name).

The stability and high effectiveness of the arrival of that data base in the USSR (2-3 weeks after publication of the official bulletins) create objective prerequisites for its wide use in information servicing.

Active work on use of the INPADOC data base is done by the Leningrad Intersector Territorial Center of Scientific and Technical Information and Propaganda (LenTsNTI), which participates in the development and introduction of automated systems for scientific and technical information (ASNTI) of RSFSR information centers and the State System for Scientific and Technical Information (GSNTI) network of automated centers.

The LenTsNTI does that work in interaction with the Scientific Production Association "Poisk" of the State Committee for Inventions and Discoveries and, in particular, with the Leningrad branch of the "Patent" Enterprise.

The ASNTI of the network of intersector territorial centers of the RSFSR is created on the basis of the formed traditional system of information service, which enters the GSNTI as a regional unit [4].

In the first stage the INPADOC data base was organized in the LenTsNTI in subsystems which provide only signal information in conditions of selective distribution of information, which is very accessible for the acquisition of experience in the development and introduction of automated services.

In 1977 the LenTsNTI accomplished in conditions of selective distribution of information the servicing of 35 enterprises and organizations of Leningrad Signal Information, which contains 10 bibliographic elements of the INPADOC data base. Sessions are held once a month on 2150 permanent thematic questions.

The experimental results confirmed the advisability of using the data base for information servicing and indicated a need to create within the framework of the RSFSR automated system of scientific and technical information a separate subsystem which takes into account the specificity of the INPADOC data base itself, the contingent of its users and the technology of the information servicing.

Since April 1978 the LenTsNTI, jointly with the Leningrad branch of the "Patent" Enterprise, has been experimentally operating a regional system of selective distribution of patent information (RASIRPI), developed in the center, under the conventional name of "Vektor-II" [5]. That system represents the type-documantary subsystem of the RSFSR ASNTI and the regional link of the ASPI of the State Committee for Inventions and Discoveries on the territory of Leningrad in Leningradskaya Oblast.

The system was accepted into industrial operation in December 1979.

A brief description of the system is presented below, and also experience in its development and introduction.

[26-2174]

2174
CSO: 1863

INVENTIONS PUT INTO SERVICE ANNOUNCED

Moscow VNEDRENNYYE IZOBRETENIYA in Russian No 2, 1980 inside front cover, pp 1, 118-120

[Excerpts] Meaning of ISIREPAT Codes for the Identification of Bibliographic Data

(11) Inventor's certificate number
(21) Application number
(22) Date of submission of application
(51) International Classification of Inventions (MKI) index
(54) Name of invention
(61) Number of basic inventor's certificate or patent in relation to which the document in question is dependent
(71) Name of applicant
(72) Name of inventor

Upon requests by organizations and enterprises TsNIIPI [Central Scientific Research Institute of Patent Information] issues information on the location of technical documentation for Soviet inventions, on the addresses of inventors and (with the existence of information regarding introduction) on the address of the enterprise at which the invention has been introduced. Information regarding the introduction of inventions is sent by mail in a month's time.

The cost of information regarding the introduction of a single invention, the location of technical documentation and the addresses of inventors and applicant organizations is 50 kopecks.

Enterprises and organizations are served with this kind of information regarding Soviet inventions only after transfer of a subscription cost amounting to not less than 20 rubles to TsNIIPI's account No 608522 in Gosbank's Moskvoretskiy Branch in Moscow. When transferring money by check or postal order the notation "Introduction" must be indicated.

The fulfillment of a request is not guaranteed without transferral of the subscription cost.

Requests are to be sent to: 113035, Moscow, M-35, Raushskaya nab., 4, TsNIIPI, Reference Information Department.

G 05

(11) 641394 (21) 2439937/18-24 (22) 3 Jan 77 2(51) G 05 B 11/01 Bulletin No 1, 1979 (72) Ye.N. Kopytko and M.N. Gornichar

(54) Regulator for Systems with Transport Delay

Invention introduced 29 Dec 1976 at Novocherkassk Polytechnical Institute, Rostovskaya Oblast.

(11) 622044 (21) 2461034/18-24 (22) 10 Mar 77 2(51) G 05 B 11/14 Bulletin No 32, 1978 (71) Tomsk Institute of Automated Control Systems and Radio Electronics Scientific Research Institute of Automation and Electromechanics (72) V.A. Beynarovich, A.Ye. Deyev and V.I. Stepanov

(54) Servo System

Invention introduced 1 Jun 1977 at Tomsk Institute of Automated Control Systems and Radio Electronics.

(11) 595704 (21) 2309311/18-24 (22) 4 Jan 76 2(51) G 05 B 13/00; G 08 C 15/06 Bulletin No 8, 1978 (71) Taganrog Radio Engineering Institute imeni V.D. Kalmykov (72) N.G. Malyshev, B.Ya. Buyanov, V.I. Finayev and A.P. Mamonov

(54) Unit for Controlling Message Switching in Several Transmission Channels

Invention introduced 7 Dec 78 at Taganrog Radio Engineering Institute imeni V.D. Kalmykov, Rostovskaya Oblast.

(11) 488189 (21) 1764389/18-24 (22) 27 Mar 72 2(51) G 05 B 19/08 Bulletin No 38, 1975 (71) State Scientific Research Institute of Heat and Power Instrument Making (72) Yu.A. Zholkov, V.K. Kosarev, A.G. Ravvin, V.G. Chibrikov and S.A. Yurkina

(54) Unit for Automatic Resetting of a Machine Tool

Invention introduced 1 Jul 1976 at ZIL [Moscow Automobile Plant imeni I.A. Likhachev] Production Association, Moscow.

(11) 411435 (21) 1667694/18-24 (22) 23 Jun 71 2(51) G 05 B 19/18 Bulletin No 2, 1974 (72) Ye.D. Avakov and B.M. Yegorov

(54) Electronic Programmer

Invention introduced 9 Jun 1978 at OKBA [Experimental Design Office for Automation] Khar'kov Branch, Ukrainian SSR.

(11) 596914 (61) 411435 (21) 2429325/18-24 (22) 14 Dec 76 2(51) G 05 B 19/18 Bulletin No 9, 1978 (72) B.F. Flaksman

(54) Electronic Programmer

Invention introduced 9 Jun 1978 at OKBA Khar'kov Branch, Ukrainian SSR.

(11) 364926 (21) 149114/18-24 (22) 7 Dec 70 2(51) G 05 D 9/02; G 01 F 23/06 Bulletin No 5, 1973 (71) State Planning and Design and Scientific Research Institute of Automation of the Coal Industry (72) V.V. Biryuk, A.A. Shchelinskiy, A.Ya. Patskan and Yu.P. Chernik

(54) Regulator of the Level of Pulp in a Disk-Type Vacuum Filter

Invention introduced 5 Jan 1972 at Bykovo Experimental Plant, Il'inskoye Settlement, Ramenskiy Rayon, Moscow Oblast. Savings from introduction--152,979 rubles per year.

(11) 492860 (21) 1797447/28-13 (22) 7 Jun 72 2(51) G 05 D 9/04 Bulletin No 43, 1975 (72) I.A. Goryunenko, A.P. Derimov, V.D. Kokhanovskiy and A.N. Serdyuk

(54) Device for Regulating Level of Fluid in Tanks

Invention introduced 20 Dec 1978 at Central Locomotive Depot, Kursk.

G 06

(11) 370599 (21) 1608583/18-24 (22) 28 Dec 70 2(51) G 06 D 3/00; G 08 C 13/00 Bulletin No 11, 1973 (71) Donetsk Scientific Research and Planning and Design Institute of Automation of Mining Machines (72) V.I. Chirkov, K.A. Podgayskiy, A.S. Lazovskiy, V.A. Selin, Yu.M. Zotov and G.S. Bakrymov

(54) Pneumatic Signaling Unit

Invention introduced 1 Jan 1972 at Avtomatgormash Scientific Production Association, Donetsk, Ukrainian SSR.

(11) 491128 (21) 1963324/18-24 (22) 15 Oct 73 2(51) G 06 F 3/04 Bulletin No 41, 1975 (71) Scientific Research and Planning Institute of Overall Automation of the Oil and Chemicals Industry (72) R.M. Kurbanov, M.Sh. Guseynov and D.G. Isayev

(54) Unit for Reading Information from a Data Transmitter

Invention introduced 1 Jan 1978 at NIPIneftekhimavtomat [Scientific Research and Planning Institute for Complex Automation of Production Processes in the Petroleum and Chemical Industries], Baku.

[45-8831]

8831
CSO: 1863

CONFERENCES

SEMINAR-SCHOOL ON INTERACTIVE SYSTEMS

Kiev KIBERNETIKA in Russian No 3, 1980 p 147

[Article by L. N. Nekrasova and M. G. Khoshtaria]

[Text] The second seminar-school on "Interactive systems" was held in the city of Borzhomi, Georgian SSR, on 17-23 February 1980 by the Council for Automation of Scientific Research under the Presidium of the USSR Academy of Sciences and the Scientific Research Institute of Cybernetics of the GSSR Academy of Sciences, the Computer Center of the GSSR Academy of Sciences, the Georgian Scientific Research Institute of Scientific and Technical Information and the Computing and Data Processing Center of the GSSR Ministry of Finances.

Participating in the work of the seminar-school were 230 persons from 25 cities and 79 scientific institutions. Reports presented numbered 132, including 4 at plenary sessions and 7 lecture reports. Academician V. V. Chavchanidze, chairman of the organizing committee, delivered an address of greeting.

The work of the seminar-school proceeded in two sections: the first on "Problem-oriented interactive information systems" and the second on "Interactive languages." Section I in turn included seminars on "Investigation, development and realization of interactive information systems," "Data bases" and "Interactive computer graphics," and section 2 seminars on "Realization of interactive programming languages," "Interaction in a natural language," and "Vocal interaction and interaction in a natural language."

Ya. A. Gel'dfandbeyn, A. D. Zakrevskiy and V. N. Andryushchenko presented reports at the plenary sessions.

The reporters noted the need and advisability of wide application and use of effective interactive systems oriented toward mass use (for example, PRIMUS-2, TSO, etc) in solving problems of the national economy, of the creation and development of "user-computer" means of communication convenient and similar to natural languages for interactive systems.

One should note as one of the distinctive features of the introduction of the conducting of a seminar-school into practice the demonstration of the PRIMUS-2, TSO, TBILOS and PRIZ systems, etc, at the GSSR Academy of Sciences Computer Center.

At the seminar-school a resolution was adopted in which it is recommended that the future program of the seminar-school be reduced to 70-80 reports and that a third seminar-school on "Interactive systems" be conducted in February 1981.

[49-2174]

2174
CSO: 1863

PUBLICATIONS

ABSTRACTS FROM THE JOURNAL 'CONTROL SYSTEMS AND MACHINES'

Kiev UPRAVLAYUSHCHIYE SISTEMY I MASHINY in Russian No 4, 1980 pp 145, 147, 149, 151

UDC 62-50:007

SOME PROBLEMS IN THE DESIGN OF AN AUTOMATED SYSTEM FOR CONTROL OF SCIENTIFIC AND TECHNOLOGICAL DEVELOPMENT

[Abstract of article by Maksimenko, V. I.]

[Text] A general description is given of the purposes and principal tasks of an automated system for control of scientific and technological development and the structure of the system as understood now and in prospect. Experience in the designing of the system on some levels of control is presented.

UDC 621.384.74

EXPERIMENTAL INVESTIGATIONS AND QUESTIONS REGARDING THE PRACTICAL REALIZATION OF METHODS OF PLANNING DEVELOPING MULTIPLE-ACCESS COMPUTER CENTER NETWORKS

[Abstract of article by Zaychenko, Yu. P., and Popenko, V. D.]

[Text] The results of experimental investigations of methods of solving dynamic problems in planning the structure of developing multiple-access computer center networks are presented and an estimate is made of their effectiveness and algorithmic complexity. Questions regarding the realization and practical use of the developed methods are discussed.

UDC 62-52:681.3.06.44

EFFECTIVE CONTROL OF FREIGHT TRAFFIC IN A MULTIPHASE SUPPLY SYSTEM

[Abstract of article by Gritsenko, V. I., and Nazarenko, N. A.]

[Text] A model of spare parts control is examined by means of which optimum variants of freight traffic in a multiphase supply system can be found. Methods are proposed for solving control problems for the cases of determined and random deliveries.

UDC 62-52:681.3.06.2

MODEL OF A CRYPTOGRAPHIC SYSTEM

[Abstract of article by Valashov, Yu. K.]

[Text] A system of parameters characterizing the properties of cryptographic systems is proposed. For estimation of the parameters of real cryptographic systems it is convenient to use a model with readily calculable characteristics. Comparative characteristics of cryptographic systems of various types are presented.

UDC 002.5.65.011.56

DATA PROCESSING IN THE PRELIMINARY STAGE OF AUTOMATED SOLUTION

[Abstract of article by Pivovarov, A. N.]

[Text] The article presents the quantitative characteristics of very massive operations accomplished by man in the stage of filling out primary documents during automated data processing. The data obtained as a result of experiment are the starting data in the planning of the technological process of automated processing and in the calculation of its effectiveness.

UDC 681.3.05

CONSTRUCTION OF SEQUENTIAL OPERATIONAL UNITS WITH A CONSTANT COMPUTER CYCLE

[Abstract of article by Tarasenko, V. P., and Toroshanko, Ya. I.]

[Text] Questions regarding the construction of sequential operational units with a constant computer cycle are examined and distinctive features of the realization of the principal microoperations in such units are reviewed. The possibilities of combining several microoperations in a single machine are demonstrated.

UDC 681.3.29./16

DATA STORAGE CONTROL METHOD FOR ELECTRONIC COMPUTERS WITH LIMITED MAIN STORAGE VOLUME

[Abstract of article by Korniyenko, G. I., Dianov, V. I., and Dianov, M. I.]

[Text] A method is proposed for data storage control which greatly increases the working effectiveness of intricate hardware-software cystems. In addition, re-enterability of all program modules is achieved and there also is the possibility of relocating the instruction and constant areas in the control level permanent storage without extra programming.

UDC 681.3.48./14

TRANSFORMATION OF DATA STRUCTURES IN SPECIALIZED CELLULAR PROCESSORS

[Abstract of article by Fet, Ya. I.]

[Text] Questions regarding the hardware realization of operations of data structure transformation are examined. Definitions of some operations (replication, permutation, weighting, compression, expansion, etc) are presented. It is demonstrated that there are possibilities of effective performance of such operations in specialized cellular processors of two types.

UDC 681.3.1

CONSTRUCTION OF AN AUTOMATON MODEL OF A PERIPHERAL

[Abstract of article by Gurvich, G. A., and Lenchuk, V. S.]

[Text] The structure of an automaton model of a peripheral is proposed, one which permits formalizing the description of functions and time parameters of the specific device on the basis of which algorithms can be developed for the functioning of the peripheral and the control device.

UDC 681.324.001.63

METHODS OF INCREASING READ-OUT MEMORY SPEED

[Abstract of article by Yorchishin, V. Ya.]

[Text] Distinctive featurs and principles of the realization of structural, technological and informational methods of increasing read-only memory speed are examined. The methods are analyzed with respect to reduction of the effective read-only memory cycle and minimization of access time.

UDC 681.3.06./97/2

POSSIBILITY OF DESIGNING RELIABLE DATA FORM CONVERTERS BASED ON IRRATIONAL-RADIX CODES

[Abstract of article by Stakhov, A. P., Azarov, A. D., and Rubin, A. G.]

[Text] The possibility of designing reliable data form converters based on Fagonacci p-codes and "golden" p-proportions is studied. A mathematical analysis is made of enhancement of the digital-analog converter reliability. Reliability achieved when using p-codes is compared with that achieved by applying the method of duplication by substitution.

UDC 65.015.13

ORGANIZATION OF DISK-RESIDENT PROGRAMS OF REAL-TIME SYSTEMS

[Abstract of article by Yefimov, Ye. N., and Yarusov, A. G.]

[Text] Distinctive features of real-time systems constructed on the basis of disk-resident programs. A method is presented for controlling programs by means of a modification unit, and also a method of calculating the effective composition of programs for periodic readout of data from an object of control.

UDC 658.562

REALIZATION OF AN INTERACTION ANALYZER UNIVERSAL IN THE CLASS OF CF LANGUAGES

[Abstract of article by Kholodenko, O. A.]

[Text] The author describes a processor for grammatical parsing of the METATRANSLATOR system which enables the input string to be syntactically analyzed in direct interaction between programmer and parsing. The possibilities and spheres of application of the processor are described and the algorithm of the processor operation, consisting in a table-type descending syntactic analyzer, is presented.

UDC 62-52:681.3.06.2

INTERACTIVE SYSTEM OF JOB PREPARATION FOR YeS COMPUTERS

[Abstract of article by Khanykov, V. V., Rybakov, A. V., and Anan'ina, N. V.]

[Text] An interactive system of job preparation of YeS operating systems and disk operating systems is examined and an analysis is made of the user's work on job preparation in such systems.

UDC 62-52:681.3.06.2

THE POSSIBILITY OF COMPLICATION OF QUESTIONS IN COMPUTER-AIDED TEACHING

[Abstract of article by Grigorishin, I. A., and Nikitin, A. N.]

[Text] Questions in expanding the possibilities of computer-aided teaching systems are examined. The RTK syntactic analyzer is suggested for use as a universal means of checking the correctness of answers involving syntactic constructions.

UDC 681.3.06/91

ONE METHOD OF ROS PACKAGE APPLICATION IN COMBINATION WITH THE GENERAL TELECOMMUNICATION ACCESS METHOD

[Abstract of article by Korniyenko, I. A., Pilipchuk, A. Ya., Pilipchuk, Al. Ya., and Serykov, G. S.]

[Text] The authors describe a method of designing shared multicomputer centers with developed subscriber networks by combining ROS, the program designed to organize multicomputer complexes, with the general telecommunication access method. Distinctive features of the realization are presented.

UDC 681.3:658.5

THE FRAGMENT DIALOG SYSTEM FOR DATA MANIPULATION

[Abstract of article by Zenkov, V. V.]

[Text] A dialog system is described for work with data with a simple structure, written in FORTRAN to simplify its realization in various computer operating systems.

UDC 681.324.001.63

ALGORITHMS FOR CALCULATION OF MODELS OF COMPUTER SYSTEM STRUCTURES WITH DIFFERENT JOB CLASSES

[Abstract of article by Kruglyy, Z. L.]

[Text] The author describes models of computing system structures represented by closed networks of mass service systems with different classes of requirements. Algorithms of their approximate solution and modeling examples are presented.

UDC 621.382.8.001

COMPUTER-AIDED DESIGN OF LARGE-SCALE INTEGRATED CIRCUIT (LSI) TOPOLOGICAL PATTERNS

[Abstract of article by Abraytis, L. B., and Blonskis, I. S.]

[Text] A method of automatic synthesis of LSI topological patterns is described, one in which the stage of manual correction of interconnections is replaced by a new procedure, computerized deformation of the configuration of connections with compression of the topological pattern to the maximum element density.

UDC 681.142.3

UNIVERSAL AUTOMATED SYSTEM FOR SYNTHESIS OF MICROPROGRAMMED AUTOMATA (UASSMA-YeS)

[Abstract of article by Baranov, S. I., and Zhuravina, L. N.]

[Text] The structure, characteristics and main programs of a universal automated system of microprogrammed automata for YeS computers are described. The system solves problems in the synthesis of digital automata, the behavior of which is described in the language of algorithm flow-charts.

UDC 65.015.11

DESIGN OF MULTIFUNCTIONAL DIGITAL-ANALOG CONVERTERS WITH DIGITAL STORAGE OF APPROXIMATION PARAMETERS

[Abstract of article by Kalinin, G. A.]

[Text] Devices for reproducing a set of functions of a digital argument characterized by apparatus redundancy are analyzed. A procedure of computer-aided designing is given. A specific example of calculation is given for one set of functions.

UDC 65.015.11

APPLICATION OF PATTERN RECOGNITION METHODS IN A COMPUTER-AIDED WAREHOUSE DESIGN SYSTEM

[Abstract of article by Glaz, A. B., Kogan, M. I., and Yufko, V. V.]

[Text] The authors examine the possibility of using pattern recognition methods to solve some problems in a computer-aided warehouse design system. Algorithms are proposed for the construction of a solution rule on an insufficient teaching sample, and on the example of the choice of the warehouse height their efficiency is demonstrated in comparison both with regression analysis methods and with individual solution rules constructed on the basis of the condition of the minimum empirical risk.

UDC 62-52:681.306.44

METHODS OF DISCRETE INPUT AND REPRESENTATION OF MEASUREMENT DATA IN AUTOMATED SYSTEMS

[Abstract of article by Dubovik, Ye. A., and Macharadze, T. D.]

[Text] The authors systematize the principal methods of discrete input and representation in a computer of measurement data different in spectral characteristics and with a given precision.

UDC 681.142.2

SOFTWARE FOR ONE CLASS OF SCIENTIFIC RESEARCH AUTOMATION SYSTEMS

[Abstract of article by Il'in, V. D., Kurov, B. N., and Khrushchev, S. N.]

[Text] Software for scientific research automation systems involving processing of table functions is examined. In solving the interpolation, numerical differentiation and integration problems, cubic splines are used. The software was developed for SM-3 and SM-4 control computers. The programming language is FORTRAN-IV.

UDC 62-50:629.13.01

MAIN PROBLEMS AND PRINCIPLES OF CONSTRUCTION OF A GROUND-AIR COMPLEX OF CONTROL OF EXPERIMENTS CONDUCTED ON SPACE VEHICLES

[Abstract of article by Belyayev, M. Yu.]

[Text] Questions regarding the control of experiments conducted on space vehicles and orbital stations. The main problems in the control of experiments are formulated. An approach to the construction of a ground-air control complex is described and its characteristics and operating indices are presented. Described as an example is the functioning of the ground equipment for analysis and control of experiments conducted on space vehicles and orbital stations.

UDC 621.391

CLASSIFICATION OF NOISY SIGNALS BY THEIR PREDICTION COEFFICIENTS

[Abstract of article by Gudonavichyus, R. V., and Zykov, F. N.]

[Text] Problems in the recognition of noisy signals of the audio range which characterize the state of moving technical objects are discussed. The results of classification of four real signals by the coefficients of linear prediction and auto- and partial correlation are presented.

UDC 681.3.48./64

DIGITIZER OF GRAPHIC IMAGES BASED ON AN X-Y PLOTTER

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

[Text] The application of a potentiometric X-Y plotter for coding (digitization) of graphic information is described. A method is proposed for low-redundant coding of curved lines of a selective series of reference points. A new property of an X-Y plotter--reversibility--permits using it as equipment for graphic data preparation with registration on punched tape for subsequent computer processing, and also as an interactive graphic terminal similar to a graphic display for work in systems with microcomputers and programmed calculators.

UDC 52-62:681.3.06.2

INTERACTIVE EDITORS IN PUBLISHING

[Abstract of article by Ratner, L. M.]

[Text] An interactive editor is described, one which is a part of computer-aided publications preparation system. Distinctive features of the editing of published texts are demonstrated. Also examined are the conditions of change introduction and methods of addressing, and also means of change introduction, of image and marking control presented to the editor, technical editor and proofreader.

UDC 62-52:681.3.06.44

REALIZATION OF SYNTHESIS UNIT CONTROL SUBSYSTEMS IN A SYSTEM FOR AUTOMATED CONTROL OF AMMONIA PRODUCTION

[Abstract of article by Bondar', A. G., Medvedev, R. B., and Fedorov, A. V.]

[Text] Laws governing the construction of systems for automated control of ammonia production, introduced into industrial operation, are analyzed. The structure is presented and the work of the control programs is described, and a characterization is given of the main problems to be solved.

UDC 62-52:681.3.06.44

AUTOMATED SYSTEM FOR RECORDING AND ANALYSIS OF LABOR LOSSES AND MORBIDITY AMONG WORKERS OF INDUSTRIAL ENTERPRISES

[Abstract of article by Ogorodnik, I. M., and Kirilyuk, V. G.]

[Text] An automated system for recording and analysis of labor losses and morbidity among workers of industrial enterprises, based on the YeS computer, is described. The possibilities of the system and its advantages over the existing system for recording morbidity are revealed. The system has undergone experimental testing at two enterprises of the city and has been accepted for introduction at the remaining enterprises.

UDC 62-52:681.3.06.44

PROGNOSTICATION AND OPTIMIZATION PROBLEMS IN THE 'TRAUMATOLOGY' AUTOMATED MANAGEMENT SYSTEM

[Abstract of article by Pshennova, E. F., Pozhariskiy, V. F., and Matasova, N. F.]

[Text] A study is made of the applicability of network analysis and statistical modeling methods to solve prognostication and control problems in medicine. A number of optimization problems solvable by network modeling methods are formulated. A formulation and an algorithm for solution of the problem of prognostication of terms and of the worst outcomes of treatment with limited resources are presented.

UDC 681.3.51./6.42

DISTINCTIVE FEATURES OF SOFTWARE FOR SYSTEMF FOR OBSERVATIONS OF THE POSTOPERATIVE STATE OF PATIENTS

[Abstract of article by Mintser, O. P., and Goyko, O. V.]

[Text] Distinctive features of software of the SINAPS system [sistema nablyudeniya za posle-operatsionnym sostoyaniyem bol'nykh], developed on the basis of the M-6000 ASVT [modular system of computer technology] control computer complex. SINAPS is compared with the previously developed NR/2100 system. The structure of the software is examined and general data transmission circuits in the system and interconnections of separate modules in it are given.

UDC 007.001.8

PROGRAM PACKAGE FOR COMPUTER RESOURCES OPERATIONAL CONTROL IN DATA PROCESSING SYSTEMS

[Abstract of article by Pan'shin, B. N., Stepanenko, A. G., and Snigireva, L. K.]

[Text] A program package is examined, an extension of YeS operating systems which control computer resources during organization of a computational process in complex data processing systems. Descriptions are given of the functional structure of programs and the data base of the package, its operating scheme and data on prospective fields of application and trends in its development.

UDC 681.3.06:007

A PACKAGE OF 'LAYOUT' PROGRAMS

[Abstract of article by Stoyan, Yu. G., Gil', N. I., and Yeshchenko, V. G.]

[Text] The authors discuss the structure of a package of programs to solve tasks in the rational layout of plane geometric objects. A model of the subject domain is presented and the input languages of the package are described. Questions connected with the organization of the computer process and generation of the working program are discussed.

[44-2174]

2174
CSO: 1863

ABSTRACTS FROM THE JOURNAL 'AUTOMATION AND COMPUTER TECHNOLOGY'

Riga AVTOMATIKA I VYCHISLITEL'NAYA TEKHNIKA in Russian No 5, 1980 pp 95, 97, 99

UDC 681.324

USE OF THE BRANCH-AND-BOUND METHOD FOR TOPOLOGICAL OPTIMIZATION OF A DATA TELEPROCESSING NETWORK UNDER RESPONSE TIME CONSTRAINT

[Abstract of article by Yanbykh, G. F.]

[Text] The task of selecting the number and disposition of data processing centers is examined, as are the subscriber servicing zone for each data processing center and the throughput of computer complexes and data transmission channels according to the criterion of the minimum reduced expdnditures on the creation and operation of a data teleprocessing network under subscriber response time constraint. Each network element consists in an M/M/1 mass service system. Subscribers are connected to the data processing center by a radial circuit, and centers are interconnected on the one-to-one principle. A general scheme of organization of the branching process of global extremum retrieval is presented, formulations of the limiting tasks are given and methods of bound calculation are described. On the first branching level the number and disposition of data processing centers are fixed and the subscribers are zoned. On the second branching level the optimum productivity of data teleprocessing network elements is determined.

UDC 681.3-181:512.56

QUASIPARALLEL DECOMPOSITION OF MICROPROGRAMMED AUTOMATA

[Abstract of article by Baranov, S. I., and Yantsen, N. Ya.]

[Text] The concept of quasiparallel decomposition of automata is introduced. As a result of it the automaton is represented as a parallel network of component automata and an input converter. It is shown that any automaton permits quasiparallel decomposition. A method of quasiparallel decomposition of microprogrammed automata is examined. The method is illustrated by decomposition of a microprogrammed automaton with a matrix structure.

UDC 519.713

SOME NEW DATA ON LOGICAL DETERMINANTS AND THEIR APPLICATION TO THE DYNAMICS OF AUTOMATA

[Abstract of article by Levin, V. I.]

[Text] Supplementing previously known disjunctive normal forms expressing a logical determinant, conjunctive normal forms double them are obtained. It is shown that from the disjunctive normal forms discarded by a portion of the conjunctions an estimate of the determinant can be obtained from below, and from the conjunctive normal forms discarded by a portion of the disjunctions, an estimate of it from above. In this way bilateral estimates of the determinant are obtained which have a linear growth of complexity.

UDC 681.325.5

ORGANIZATION OF MULTICHANNEL ACCESS IN DATA PROCESSING SYSTEMS WITH A HIGH DEGREE OF PARALLELISM

[Abstract of article by Makarevich, O. B.]

[Text] A new form of servicing in a computer system with multichannel access conditions of separation of the computer resource is introduced. The probability of absence of queueing in such a system is estimated and an example of the hardware realization of the operation of resource control is examined.

UDC 681.325.657

MULTIFUNCTIONAL SYMMETRIC AND THRESHOLD LOGICAL MODULES BASED ON CELLULAR REGISTER STRUCTURES

[Abstract of article by Kalyayev, A. B., and Bozich, V. I.]

[Text] The construction of multifunctional logical modules on the basis of cellular commutation register structures is examined. For the realization of symmetric and threshold functions of the logic algebra the module is rearranged from the boundary inputs of the structure. The uniformity and simplicity of the structural cell and the minimum number of tuned inputs permit making multifunctional logical modules on the basis of cellular register structures in the form of large-scale integrated circuits.

UDC 519.2

DETERMINATION OF THE NECESSARY STORE VOLUMES IN COMPUTER SYSTEM DESIGN

[Abstract of article by Mussonov, G. P., and Paramonov, Yu. M.]

[Text] The volumes required for storage of the flow of requests for servicing in a computer system with use of the theory of construction of p-bounds are determined. Expressions are obtained for determining the volume of the store of applications

for processing in a computer system for different formulations of the limitations on the probability of their performance during separate and combined consideration of several application flows.

UDC 519.872.5

STRUCTURAL ADAPTATION OF A BILINEAR MASS SERVICING SYSTEM WITH AN AUXILIARY INSTRUMENT

[Abstract of an article by Gortsev, A. M., and Pottosina, S. A.]

[Text] A mass servicing system with three servicing devices and two incoming flows of requests of various types is examined. The first instrument services requests only of the first type, and the second only of the second type. A third (auxiliary) instrument can service requests of both the first and second types. Four different structures are distinguished in the given system. In very simple prerequisites (Poisson incoming request flows and exponential servicing) the problem of structural adaptation is solved, that is, when the intensities of incoming flows are unknown the system is adpated to the optimum structure in the process of functioning. Ideas of the statistical theory of estimates and verification of hypotheses, and also of an automaton with linear tactics, are used for realization of adaptive control.

UDC 519.718

RECURRENT SPLICING OF THE CHECK TEST OF A TWO-LEVEL AUTOMATON

[Abstract of article by Sklyarevich, A. N.]

[Text] The rules of splicing of the check test from sets of tests for verification of derivative subautomata are substantiated for a combination automaton, two-level in relation to a certain input signal. A recurrent procedure based on those principles is proposed for finding the check test, used for redundant and non-redundant automata. Examples of use of the procedure are presented.

UDC 681.326.06

CONSTRUCTION OF AN INPUT TEST SEQUENCE TO CHECK COMBINATION CIRCUITS FOR A SINGLE SHORT CIRCUIT

[Abstract of article by Dubrovskiy, A. V.]

[Text] The conditions required for a single short circuit of connections in a combination circuit by a sequence of input vectors are formulated. An analytical method is presented for synthesis of a sequence in which the conditions for short circuit detection are fulfilled.

UDC 681.324

ANALYSIS OF END-TO-END FLOW CONTROL IN COMPUTER NETWORKS

[Abstract of article by Boguslavskiy, L. B., Drozhzhinov, V. I., and Martirosyan, V. A.]

[Text] Various methods of limiting the load to the virtual connection of data sources and addressees are examined, as well as circuits for confirmation and repeated transmissions of messages during end-to-end control of data flows. The methods take into account the main distinctive features of data exchange procedures developed by the present time. An imitational simulation of methods of end-to-end flow control is made, including adaptive methods working under conditions of sudden variations of load on the connection. The conditions are determined for the effective functioning of the virtual connection, a comparative analysis is made of the methods of end-to-end flow control and control parameters optimum with respect to the given criteria.

UDC 681.113.92

A TIMING SYSTEM FOR ANALYSIS OF TIME DOMAIN SIGNAL PARAMETERS

[Abstract of article by Zagurskiy, V. Ya., and Vedin, V. Yu.]

[Text] A method of precise chronometric conversion in real time and a multichannel system of analysis of signal time parameters based on it are described. The proposed method of structural organization of the chronometric converter and the system assures multiplicity of function and permits obtaining high metrological characteristics.

UDC 621.317.3

DISTINCTIVE FEATURES OF CONSTRUCTION OF A DIGITAL STOCHASTIC CONVERTER OF NONLINEAR DISTORTIONS

[Abstract of article by Mikelson, A. K., Krauze, A. V., and Gonera, M.]

[Text] Distinctive features of the hardware realization of a method of digital stochastic conversion of the value of the nonlinear distortions coefficient. On the level of structural circuits the principles of construction of a measurer of medium capacity and a measurer of the first harmonics of the signal. An analysis was made of the hardware and methodical errors of the converter.

UDC 681.324

LOCAL DATA TRANSMISSION NETWORK WITH STORAGE COMMUTATION

[Abstract of article by Zavadskiy, V. M.]

[Text] The author describes the principles of construction of a collective-use data transmission network, intended for the creation of a system for the automation of several experiments with large data flows. The network has a loop structure.

Successive data transmission by a phase-frequency code with a cyclic frequency of 10 MHz is used. A linear procedure of package commutation simulates reference to a memory with direct access. For network servicing a parallel independent data transmission network is introduced--a system of monitoring, indication and control which is also used for the effective control of the simplest subscribers of the circuit.

UDC 631.327.01

CONSTRUCTION OF SYSTEMS WITH VARIABLE STRUCTURE FOR AUTOMATION OF SCIENTIFIC RESEARCH ON THE BASIS OF AN IEC INTERFACE

[Abstract of article by Domaratskiy, A. N., Domaratskiy, S. N., and Sitnikov, L. S.]

[Text] Questions of the application of an instrumental interface of the user for the construction of a collecting and processing system with a variable structure are examined. It is shown that application of the interface permits creating from autonomous instruments and peripherals user systems with or without a computer. An example is given of the organization of data exchange between the branch of an interface with an RDR-11 (M-400) mini-computer.

[50-2174]

2174
CSO: 1863

UDC 007.52:62-50

SCIENTIFIC PROBLEMS OF ROBOT ENGINEERING

Moscow NAUCHNYYE PROBLEMY ROBOTOTEKHNIKI in Russian 1980 signed to press 14 Feb 80 pp 2-4, 100-103

[Annotation, foreword and abstracts of articles from book edited by D. Ye. Okhotsimskiy and Ye. P. Popov, corresponding members of the USSR Academy of Sciences. Scientific Council for Theory and Principles of the Designing of Robots and Manipulators and State Scientific Research Institute of Machine Sciences imeni A. A. Vlagonravov, Izdatel'stvo "Nauka," 2750 copies, 103 pages]

[Text] Annotation

The collection is devoted to urgent questions in the creation and operation of robot engineering systems. It examines problems in the development and application of industrial robots and, in particular, the problem of use of robots in welding production; the main tasks in the development of the elementary base of robot engineering are reflected, both in general and specially for industrial and underwater robots; there are discussions of problems connected with elevation of the functional properties of an operator-manipulator system with reference to underground and logging machinery, and also to robot engineering systems for extreme media.

The collection is designed for a broad range of specialists in the area of robot engineering, and also can be useful to VUZ students of the corresponding specialties.

Foreword

The present collection is devoted to scientific and technical problems in the creation of robot engineering systems. Much attention has been given to consideration of the elementary base of robot engineering, which determined the improvement of existing and the creation of new, promising robots. In addition, the collection includes articles on individual questions in the creation and application of manipulation robots in the national economy.

The elementary base is the main foundation for the implementation of modern and prospective scientific developments in the area of robot engineering. At the same time the state of the elementary base is such that it forces us to turn serious attention to the complex elaboration of many varied physico-technical problems requiring the participation of various specialists and the organization of a continuous series of developments, from basic scientific research to their implementation in industrial production.

The main problems in the development of the elementary base of robot engineering are discussed in the collection. Questions relating to the elementary base of industrial and underwater robots are examined separately.

Hybrid computer systems for robot simulation during the completion of work on their control systems are described.

Some principles of the construction of technical vision systems and problems of perception of the environment are examined.

Methods of increasing the efficiency of biotechnical and interactive systems of control of robot and manipulator actuators are discussed.

A number of articles are devoted to the creation and application of manipulation robots to do underground and logging work auxiliary and some main operations (including welding) in production shops.

The collection is designed for a wide range of specialists in the area of robot engineering engaged in both scientific research and in the technical realization of robot engineering systems. It also had the goal of turning attention of representatives of a number of related specialties, who must be drawn in for the development of the elementary base of robot engineering systems.

UDC 519.95:62-50

MAIN TASKS IN THE DEVELOPMENT OF THE ELEMENTARY BASE OF ROBOT ENGINEERING

[Abstract of article by Popov, Ye. P.]

[Text] Progress in the development of robot engineering in all the numerous areas of its application will depend substantially on its elementary base. In the article a classification is given of the elements necessary for the development of robot engineering, with a brief characterization of their role and desired properties.

UDC 007.52

THE ELEMENTARY BASE OF INDUSTRIAL ROBOTS

[Abstract of article by Yurevich, Ye. I.]

The rapidly developing industrial robot engineering presents for the elementary base of robots specific requirements which impel one to pose the question both of the development of new modifications of already existing elements and of the creation of set-making equipment new in principle, including the search for and investigation of new physical principles of their action. This is dictated, in turn, by the main requirements for industrial robots, which are not completely satisfied today, mainly due to the absence of a corresponding elementary base. These are primarily the requirements of reliability and resources, rapidity, precision, mass and dimensions and cost.

The connection between these characteristics of industrial robots and corresponding requirements and their elementary base is examined in the article.

UDC 007.52:62-50

DEVELOPMENT OF THE ELEMENTARY BASE OF INDUSTRIAL ROBOT CONTROL SYSTEMS

[Abstract of article by Surnin, B. N., and Knauer, I. B.]

[Text] Questions of the construction of industrial robot control systems are examined in the article from the point of view of the contemporary state of their elementary base and the prospects of the development of individual functional units are designated: position sensors, conductors, program input devices and stores. Examples of some existing Soviet and foreign control system designs are described, distinctive features of the construction of the indicated units are shown, their technical characteristics are presented and prospective designs on the contemporary elementary base are]roposed. The article is intended for scientific, engineering and technical workers studying questions in the creation of industrial robot control systems.

UDC 551.46.08-519

ELEMENTARY BASE OF UNDERWATER ROBOTS

[Abstract of article by Yastrebov, V. S.]

[Text] The structure of the elementary base of underwater robots is examined. It is noted that the composition of an elementary base must also include mathematical models and algorithms for its behavior. The main elements of the effector system of underwater robots are enumerated which have been developed and tested and are being used on individual examples of research robots. It is noted that the main attention of specialists should be concentrated on the development of the elementary base of the sensor system of underwater robots. The elementary base of on-board computer systems is in the stage of investigation.

UDC 007.52:62-50

HYBRID COMPUTER SYSTEMS AND ROBOT SIMULATION

[Abstract of article by Kogan, B. Ya., and Petrov, A. A.]

[Text] The article examines the specifics of robot engineering systems which determine the great prospects of the application of hybrid computer systems for simulation in the process of development and investigation. It presents the results of development on the basis of third-generation hybrid computer systems [GVS--gibridnaya vychislitel'naya sistema] (the GVS-100 and GVS "Rusalka"), simulating systems oriented toward the investigation of robots. This system was used to simulate systems for controlling the motion of robot effectors. The results of simulation show that GVS permit simulating the dynamics of effectors of real-time work with the required precision, assuring in that case the possibility of half-scale simulation and flexible interactive working conditions.

UDC 62-506.2

PRINCIPLES OF CONSTRUCTION OF TECHNICAL VISION SYSTEMS

[Abstract of article by Pupkov, K. A., and Putilov, G. P.]

[Text] The article examines characteristic features of existing technical vision systems, the organization and composition of their hardware, the class of solvable problems and the factors explaining the limited possibilities of technical vision systems and the existing disproportion between the motor and information possibilities of present-day robot technological systems. A variant of technical realization of video data input-output is proposed, oriented toward the transmission of coded images in a regime of simulation of a magnetic tape store of a standard computer system with use of a buffer on a cathode-ray storage tube. The characteristic aspects of software being developed for technical vision systems are presented: preliminary tuning of a stereo-television system on the basis of estimation of distance with the use of longitudinal parallax, combination of the process of scene description with restoration of space characteristics, scene segmentation and the identification of objects with use of depth characteristics.

UDC 007.52:62-50

ENVIRONMENT PERCEPTION PROBLEMS

[Abstract of article by Platonov, A. K.]

[Text] The article examines problems in the organization of measurements of characteristics of the environment in information systems of sensitive robots. Two classes of problems are distinguished in the construction of of models of the environment. Ways to curtail information flows in technical perception systems are pointed out for each class. Models of perception systems of both classes are described, and the principles of organization of such systems are presented on the basis of analysis of experience in work with them.

UDC 519.95

METHODS OF INCREASING THE EFFECTIVENESS OF BIOTECHNICAL AND INTERACTIVE MANIPULATOR CONTROL SYSTEMS

[Abstract of article by Popov, Ye. P., Kuleshov, V. S., and Yushchenko, A. S.]

[Text] The article examines the principal methods of increasing the effectiveness of the work of manipulator robots intended for the performance of complex and untypical technological operations. A comparative analysis is given of automatic, biotechnical and interactive methods of controlling general-purpose manipulators. The principles of construction of manipulator robots with interactive control are formulated and the principal methods and problems in their planning are pointed out.

UDC 62-519:621.242

IMPROVEMENT OF A MANIPULATOR CONTROL SYSTEM FOR UNDERWATER WORK

[Abstract of article by Zagorodnyuk, V. T., Ivanov, S. O., and Yatskevich, V. A.]

[Text] Mobile manipulation machines are widely used for the extraction of minerals and the construction of underground structures. A functional diagram for control of a mobile manipulation robot, using a laser beam as a control channel, is described. To give the robot adaptive possibilities, it is proposed to use a biotechnical system of semi-automatic manipulator control implementing an algorithm of control based on the speed of the working organ in spherical coordinates.

UDC 62-5.002.5

SOME WAYS TO EXPAND THE FUNCTIONAL POSSIBILITIES OF THE SYSTEM OPERATOR-MANIPULATOR WITH RESPECT TO LOGGING MACHINERY

[Abstract of article by Afanas'yev, O. P., Ilyukhin, Yu. V., Lobachev, V. P., Tsvetkov, A. A., and Yurchenko, S. K.]

[Text] The article examines ways to increase capacity, reduce labor-intensiveness and reduce the psycho-physiological load on the operator of logging machinery, based on the application of a copying and programmed method of manipulator control. Correcting devices are proposed which expand the functional possibilities of the system operator-manipulator as a result of matching the dynamic properties of the conductors with the characteristics of the power source of limited capacity with the properties of the human operator. The task of determining the laws of manipulator control in automatic conditions assuring minimum expenditures of time on execution of the working operations is examined.

UDC 007.52:002.614

PROSPECTS OF DEVELOPMENT AND APPLICATION OF INDUSTRIAL ROBOTS

[Abstract of article by Yurevich, Ye. I.]

[Text] The article examines two main tasks of industrial robot engineering: the development of industrial robots and the creation of robotized processes and objects. Also discussed are the prospects of improving the main technical characteristics of modern industrial robots with programmed control and the creation of industrial robots with adaptive control. The stages of robotization of modern production are analyzed as one of the principal ways to accomplish complex automation and solve the manpower problem.

UDC 621.791.03-52.001

PROSPECTS OF THE CREATION AND INTRODUCTION OF WELDING ROBOTS

[Abstract of article by Tsigankov, Yu. K.]

[Text] Problems in the creation of work for arc electric welding in a medium of protective gases are examined in the article. Requirements for characteristics of control devices and welding equipment are substantiated. Some elements for sensitive robots are given and the conditions of their application are determined. The area of application for robots for arc welding is analyzed, and also the composition of the equipment of a robotized workplace for welding.

[16-2174]

2174
CSO: 1863

UDC 007.52:62-50

HYBRID COMPUTER SYSTEMS AND ROBOT SIMULATION

Moscow NAUCHNYYE PROBLEMY ROBOTOTEKHNIKI in Russian 1980 signed to press 14 Feb 80 pp 34-47

[Article by B. Ya. Kogan and A. A. Petrov]

[Text] In the creation of robot engineering systems an entire series of problems arises, the solution of which by analytical methods is very difficult or quite impossible. One of the most fruitful approaches to the solution of such problems is the method of mathematical simulation, based on the wide use of computer technology.

Machine simulation has obtained application in practically all stages of the development and investigation of robot engineering systems: in the synthesis of a system from idealized elements and units (for the estimation of various structures of the system); in the selection and identification of unknown parameters; for the consideration of distimctive features of the technical realization of individual devices and elements and the sensitivity of the system to them; in the estimation of the optimum characteristics of the system and its limiting possibilities; in the stage of half-scale investigations, when already manufactured elements of real working apparatus are connected to the system being simulated. In addition, models of various subsystems can be used directly in the control circuit for the implementation of methods of control with a standard model (forecasting, the construction of program controls, etc).

Specific distinctive features of the simulation of robot engineering systems can be reduced to the following:

--objects of simulation--essentially nonlinear dynamic systems with a large number of degrees of freedom;
--the need to accomplish complex algorithmic procedures with a large number of logical and arithmetic operations;
--the obligatory requirement of work on a real and in a number of cases a shortened time scale;
--the need to assure convenient commuication of the investigator with the system to be simulated, including directly in the process of machine experiments;
--the possibility of coupling the model with elements of real equipment.

In an overwhelming majority of cases to simulate robots at the present time digital computers are used which far from always can assure the necessary total rate of calculations, which leads to a need to simplify methods and lower the quality of control. On the other hand, an analog computer, although it provides rapid integration of ordinary differential equations, does not permit solving the problem on the whole, due to limitation of the algorithmic possibilities and the large volume of required equipment.

Many of the noted shortcomings of autonomous digital and analog computers can be successfully overcome when modern hybrid computer systems [gibridnaya vychislitel'naya sistema-GVS] are used [1]. In addition, simulation on a hybrid computer system permits finding a very rational distribution of control algorithms between the digital and analog parts for subsequent use in the program-apparatus realization of robot controls. An extremely wide spectrum of hybrid computer systems is known at the present time, among which the greatest prospects for robot simulation have been opened up by so-called balanced hybrid computer systems, in which analog and digital computers make a substantial contribution to the total computational process.

A simulating system oriented toward robot research is now being developed on the basis of third-generation balanced hybrid computer systemsm the GVS-100 [2,3] and the GVS "Rusalka" [4]. We will examine those hybrid computer systems and some results of their application to simulate a system for control of the movements of robot effectors.

The GVS-100 belongs to the class of balanced hybrid computer systems of medium capacity with a single-level hierarchy of control. Figure 1 presents a block diagram of the GVA-100. The GVA "Rusalka" consists of a system with a two-level hierarchy which thanks to its structure permits, all other conditions being equal, increasing the total capacity to 4 times that of single-level systems. Figure 2 presents a block diagram of the GVS "Rusalka." Both systems are equipped with developed hybrid software [2,3], and provision is made for the possibility of their flexible increase through enlargement of the main storage volume, the quantity of peripherals, analog racks and converter modules. Figure 3 shows the appearance of the GVS "Rusalka."

The digital part of the GVS-100 consists of a medium-speed digital computer (300,000 equivalent operations per second with respect to the Gibson-1) with developed order sets and interrupt systems. Provision is made for the possibility of work with eight peripherals. The external storage was on magnetic disks with interchangeable packages with a capacity of 768,000 words each.

The digital parts of the GVS "Rasulka" consist of several (up to seven) EMU-200 analog racks, connected to the general control panel of the system. From those panels the operator can control all the racks as a single system. In the GVS-100 through the general panel automatic control of the analog part is accomplished from the digital and data are transmitted regarding the state of the computational, logical and control elements of the analog computer. In the GVS "Rusalka" the general control panel also contains all the equipment for coupling the analog computer and converters with the digital computer, providing in that case elimination of duplication of execution of control instructions.

The GVS-100 and GVS "Rusalka" have the following principal technical characteristics:

	GVS-100	GVS "Rusalka"
System structure	single-level	two-level
Main memory capacity	up to 64 (32-digit words)	M-400--up to 24 (16-digit words) M-4030--256
External memory	on magnetic disks	M-400--on magnetic disks; M-4030-- on magnetic disks and tape
Total number of computing elements of analog part (per rack)		
linear	170	120
nonlinear	26	48
Total number of logical units	92	125
Static error of performance of operations, %		
linear	0.02	0.02*
nonlinear	0.2	0.1
Limiting frequency of periodization of solution, kHz	2	2
Number of lines of communication for control information between analog and digital computers	58	58
Number of conversion channels		
analog-digital	32	64
digital-analog	32	32
from them with multiplication	8	24
Converter digit capacity	13 binary digits + sign	13 binary digits + sign
Mean conversion time		
analog-digital	25	37
digital-analog	12	20.5
settings of coefficients	2000	940

*With use of microconductor resistances.

Let us note that the presence in the analog parts of both systems of a developed parallel logic, comparators, electronic keys, functional relays, electronic control systems, digital control coefficients and connections as regards control instructions with the digital computer assures convenience of coupling and increases the working efficiency. The GVS "Rusalka" also is characterized by the presence of a roughly exact system of digit-controlled coefficients (this permits reorganizing the coefficients in the process of solution), and also doubling the quantity of nonlinear solving elements.

The GVS-100 software solves hybrid problems on digital and analog computers in real time, accomplishes the work of the digital part in a multiprogram regime and automates the preparation of tasks for analog computers.

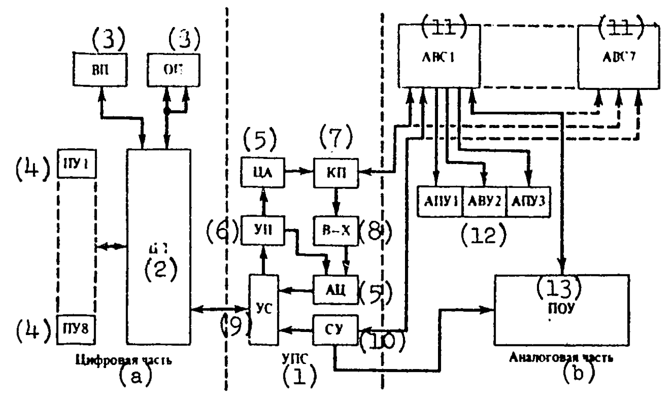


Figure 1. Block diagram of the GVS-100.

a -- Digital part
b -- Analog part
1 -- conversion and coupling devices
2 -- central processor
3 -- external store (magnetic disk) and main store (ferrite disk)
4 -- peripherals
5 -- digital-analog and analog-digital converters
6 -- converter control unit
7 -- patch panel
8 -- selector and storage
9 -- control and synchronization unit
10 -- control signals unit
11 -- analog racks
12 -- analog periphery units
13 -- general control panel
14 -- main-line channel

The programming system for the GVS-100 provides for the use of the internal language ALMO (this permits expansion of the system through the addition of the corresponding translators, for example: FORTRAN-ALMO, ALGOL-ALMO, etc) and AUTOCODE, performed by operators of communication between separate GVS-100 equipment and operators of analog computers according to signals from digital computers, specialized languages united into a system for automation of analog programming. The same system includes a service program library (including programs for communication with analog computers) and algorithms. All the library programs written in machine codes are stored on magnetic disks.

The adjustment system discovers errors in programs in different languages and introduces the necessary changes and additions into the debugged program.

The software of the GVS "Rusalka" consists of software added to the M-400 digital computer, supplemented by a hybrid assembler, an operational superstructure for the hybrid assembler, a hybrid dialog adjuster, a package of hybrid standard programs, a package of external functions assuring the joint work of the BASIC algorithmic language with a hybrid assembler. The structure of the GVS "Rusalka" software is presented on Figure 4. The hybrid assembler maximally takes into account apparatus characteristic of the coupling of the digital computer with the analog computer and the signal conversion device, which permits during the work of hybrid operators

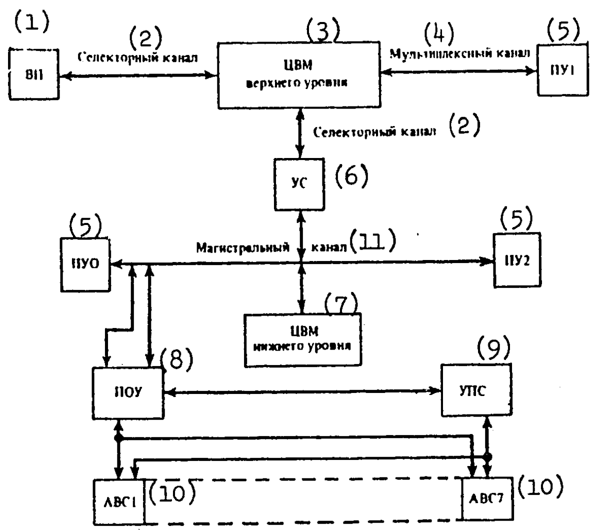


Figure 2. Block diagram of the GVS "Rusalka"

1 -- external memory (magnetic disk)
2 -- selector channel
3 -- upper-level digital computer
4 -- multiplex channel
5 -- peripheral
6 -- control and synchronization unit
7 -- lower-level digital computer
8 -- operator's control panel
9 -- conversion and coupling devices
10 -- analog racks
11 -- main-line channel

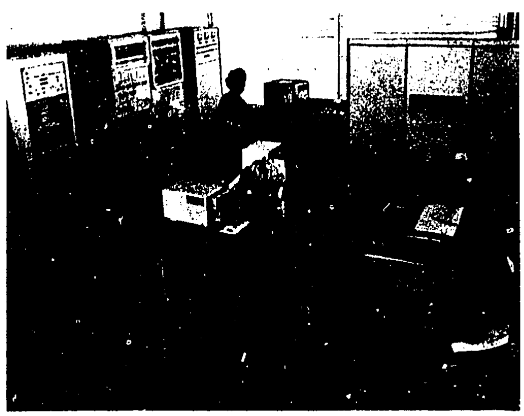


Figure 3. General appearance of the "Rusalka" hybrid computer system.

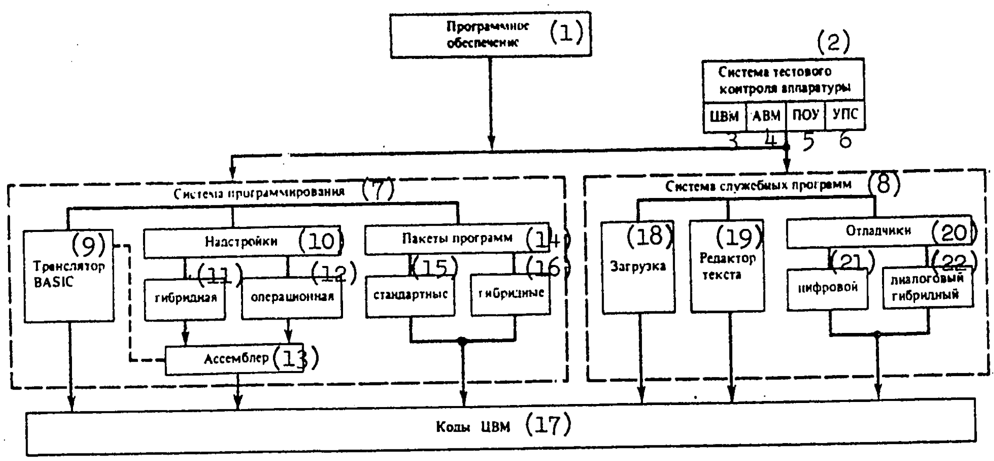


Figure 4. Strucutre of the GVS "Rusalka" software.

1 -- software
2 -- apparatus test monitoring system
3 -- digital computer
4 -- analog computer
5 -- general control panel
6 -- conversion and coupling unit
7 -- programming system
8 -- program service system
9 -- BASIC translator
10 -- superstructure
11 -- hybrid
12 -- operational
13 -- assembler
14 -- program packages
15 -- standard
16 -- hybrid
17 -- computer codes
18 -- load
19 -- text editor
20 -- adjustors
21 -- digital
22 -- dialog hybrid

obtaining the analog-digital and digital-analog conversion times, the establishment of electron coefficients, control of analog computer regimes, etc, close to the limiting possibilities of analog computers and signal conversion devices.

In a hybrid assembler the specifics of the work of a hybrid system in real time are taken into account, which is achieved, on the one hand, by minimizing expenditures of time during the work of system programs and, on the other, through the introduction of additional equipment into the coupling system, which assures priority handling of interruptions from the analog computer.

The introduction of the hybrid dialog adjuster, which permits adjusting both the digital and analog parts of the hybrid proram from a single panel, considerably reduces the time required to prepare a task for solution and creates greater convenience of communication of the user with the system. All the mentioned software is combined into a single system that assures efficient writing, adjustment and solution of hybrid tasks.

In addition, as regards the task of simulating the work of robot manipulator algorithmic and program software has been developed for an interactive regime of simulation on a signal conversion device with visual representation of the results on a display screen [2,3].

We will illustrate the application of the GVS under consideration on the example of simulation of a system for control of the motion of the robot effectors. We will examine a control system of a fairly common type (Figure 5), one combining program control with correction through feedback.

In accordance with the desired program changes of the generalized coordinates $\theta_{lim}(t)$, formed by the generator of program trajectories, the program moment calculation unit produces a calculation of the necessary hinge moments $M_{lim}(t)$ and the program control calculation unit calculates the necessary program control $u_{lim}(t)$, which arrives at the leads of the robot effector. In comparing the real motion $\theta(t)$ with the programmed motion, the K unit in the feedback circuit produces the correcting effect $\theta u(t)$ to eliminate the discrepancy. The information about the deviations of $\theta(t)$ from $\theta_{lim}(t)$ can also be used to tune the program trajectory generator, program moment calculation and program control calculation units by means of adaptation unit A.

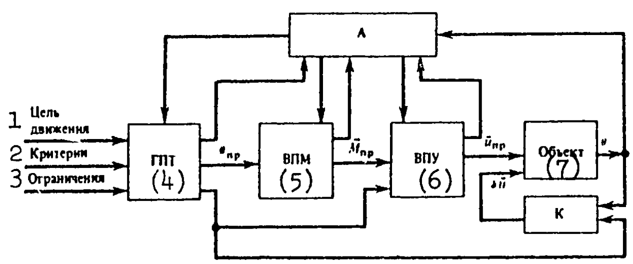


Figure 5. Block diagram of a robot effector control system.

1 -- Goal of movement
2 -- Criteria
3 -- Limitations
4 -- Program trajectory generator
5 -- Program moment calculation unit
6 -- Program control calculation unit
7 -- Object
np -- limiting
A -- Adaptation unit
K -- Correction unit

The generator forming program trajectories must function so that the effector completes the required movement in the presence of definite limitations on the allowable region G of change of generalized coordinates. The goal of movement can be, for example, achievement of the state R (θ), determinable by the given position of the effector in the working space or displacement of the end of the effector on the given curve or in accordance with the given law of motion F (R,t) = 0. In that case additional requirements can be set for the quality of motion, for example, the requirement of minimization of energy expenditures, with the possibilities of the GVS taken into account, the method of formation of program movements based on minimization of divergences W between the required and current state of the effector ($W = F(R,t)$) seems most promising. The systems of differential equations minimizing W, the construction of which was theoretically examined in [5-7], can be successfully solved by means of continuous algorithms which are very effectively realized on hybrid computers.

It is proposed to simplify the system of ordinary differential equations, the solution of which would converge to the required state, corresponding to a minimum value of W, from any allowable initial state with prescribed exactness; it would be stable in the required state; it would lie completely in the allowable region G; there would be as high requirements as necessary for calculation of the program moments and controls in program moment calculation and program control calculation units (see Figure 5).

In simulating the program trajectory generator a gradient system of minimization of W was used, one satisfying the enumerated requirements. For the case where the dynamics of each component of the effector (with consideration of conductors) can be described by a second-order differential equation, the equations of the generator have the form

$$\tau\ddot{\theta} + \dot{\theta} = \mathbf{f} + \mathbf{s}, \qquad (1)$$

where

$$\tau = \begin{Vmatrix} \tau_1 & 0 & \dots & 0 \\ 0 & \tau_2 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \tau_N \end{Vmatrix};$$

$$f_i = \begin{cases} \varphi_i = -k_i \frac{\partial W}{\partial \theta_i} & \text{at } |\varphi_i| \leqslant v_i; \\ v_i \operatorname{sign} \varphi_i & \text{at } |\varphi_i| > v_i; \end{cases}$$

t_i and k_i are coefficients; v_i is the maximally allowable absolute value of the rate of change of the generalized coordinate; $i = 1, 2, \dots, N$ (N is the number of degrees of mobility); s is the penalty function, which prevents emergence of the solution from the allowable region G.

The dependences of the program angles formed by the program trajectory generator, of their derivatives and also their trigonometric functions on time are used in the program moment calculation unit for calculation of program moments in accordance with the model of dynamics of the effector. If one starts from a Lagrangian equation of the second kind, that model can be described by the following equations:

$$\sum_{j=1}^{K}\left[\sum_{k=1}^{j} Tr(A_{ij}J_jA_{jk}^{\tau})\ddot{\theta}_k + \sum_{k=1}^{j}\sum_{p=1}^{j} \mathrm{Tr}(A_{jkp}J_jA_{ji}^{\tau})\dot{\theta}_k\dot{\theta}_p - - m_jGA_{ji}\bar{\rho}_j\right] = M_i, \qquad (2)$$

where

$$A_{ij} = T_0^1T_1^2 \dots QT_{j-1}^j \dots T_{i-1}^i \qquad (i \geqslant j);$$

$$A_{ijk} = T_0^1T_1^2 \dots QT_{j-1}^j \dots QT_{k-1}^k \dots T_{i-1}^i \quad (i \geqslant k, j);$$

T^j_{j-1} is the matrix of conversion of coordinates from the (j-1)-th system into the j-th:

$$Q = \begin{Vmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{Vmatrix};$$

J_i and m_i are the matrices of inertial and mass of the i-th member respectively; $G = \{g_x, g_y, g_t, 0\}$ is the acceleration of gravity; $\bar{\rho}_j$ is the vector of the center of gravity of the j-th member; M_i is the moment acting on the i-th hinge.

Now, being given a model of the used conductors, we can calculate the required program control u_{1im}. For example, for a direct current conductor we can write

$$u_i = \alpha_{1i}\ddot{\theta}_i + \alpha_{2i}\dot{\theta}_i + \alpha_{3i}M_i, \qquad (3)$$

where a_{ji} are constants.

The simulation of the control system under consideration (Figure 5) on GVS was done on examples of specific effectors of robots: three-membered legs of a walking apparatus and a four-membered spatial manipulator.

During hybrid realization of a program trajectory generator the penalty function was calculated with a digital computer which forms a procedure for by-passing forbidden zones in a space of generalized coordinates, and the remaining part of the system of nonlinear differential equations (1) was solved on an analog computer, which permitted generating program movements not only in a real but also in an accelerated (by more than one order of magnitude) time scale with an error of not more than 0.3°. Calculations of program moments and equations in accordance with algebraic equations (2) and (3) were made in a real time scale on the digital part of the hybrid computer system, although an analog computer could have been used for that.

To investigate experimentally the control system under consideration the GVS "Rusalka" was coupled with a real effector--the leg of a prototype of a walking apparatus equipped with electrical conductors. The processes of transition of the manipulator into the prescribed angular position (Figure 6) permit comparing the quality of execution of the programmed motion of one of the degrees of mobility when an ordinary positional tracking system (1) is used and during calculation of the program control by means of the model of control system (2) shown on Figure 5.

Experience in the application of hybrid computer systems thus shows that in their technical possibilities they permit simulating the dynamics of the robot effectors in a real time scale and with the required precision, assuring in that case the possibility of half-scale simulation and flexible interactive working conditions. Taking into consideration the specifics of robot engineering problems, one should orient present-day universal hybrid computer systems toward the solution of those problems in making a number of specialized devices.

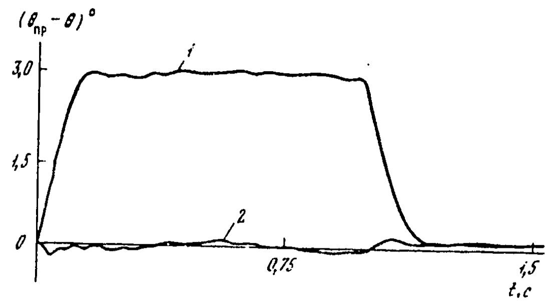


Figure 6 Error of execution of a program trajectory by a real effector during control by (1) a tracking and (2) a combined system.

BIBLIOGRAPHY

1. Kogan, B. Ya., and Petrov, A. A. On the possibility of using hybrid computer systems for simulating and controlling manipulator robots. In the book: Mezhdunarodnaya konferentsiya po iskusstvennomu intellektu (International Conference on Artificial Intellect), Repino, 1977, Annotations of Reports, Moscow: Scientific Council for the Complex Problem "Cybernetics" of the USSR Academy of Sciences, 1977, pp 10-11.

2. Gibridnaya vychislitel'naya sistema GVS-100. Apparatura i printsip yeye postroyeniya (The GVS-100 Hybrid Computer System. Apparatus and Principles of Its Construction). Professor B. Ya. Kogan, editor. Moscow, Institute of Control Problems of the USSR Academy of Sciences, 1974, 129 pages.

3. Gibridnaya vychislitel'naya sistema GVS-100. Primeneniye gibridnoy vychislitel'noy sistemy GVS-100 dlya resheniya zadach upravleniya (The GVS-100 Hybrid Computer System Application of the GVS-100 Hybrid Computer System to Solve Control Problems). Prof. B. Ya. Kogan, editor. Moscow, Institute of Control Problems of the USSR Academy of Sciences, 1975, 94 pages.

4. Kogan, B. Ya. Third-generation hybrid computer systems and prospects of their development. In the book: V Vsesoyuz. nauch-tekh konf "Dal'neysheye razvitiye analogovoy i analogo-tsifrovoy vybhislitel'noy tekhniki" (Fifth All-Union Scientific-Technical Conference on "Further Development of Analog and Analog-digital Computer Technology"). Summaries of Reports. Moscow. SOVETSKOYE RADIO, 1977, pp 6-9.

5. Kulakov, F. M. Organization of supervisory control of manipulator robots. I. IZVESTIYA AN SSSR, TEKHNICHESKAYA KIBERNETIKA, 1976, No 5, pp 37-46.

6. Kulakov, F. M. Organization of supervisory control of manipulator robots II. Ibid, No 6, pp 78-90.

7. Kulakov, F. M. Organization of supervisory control of manipulator robots III. Ibid, 1977, No 4, pp 51-66.

[16-2174]

2174
CSO: 1863

UDC 007.52:62-50

ENVIRONMENT PERCEPTION PROBLEMS

Moscow NAUCHNYYE PROBLEMY ROBOTOTEKHNIKI in Russian 1980 signed to press 14 Feb 80 pp 50-61

[Article by A. K. Platonov]

[Text] In psychology the process of psychic reflection of objects and events of the surrounding world directly acting on the organs of feeling is called perception. Modern psychology distinguishes sensation, that is, reflection by organs of feeling of separate properties of material things, and perception--a combination of separate sensations into an integral specific image. The act of perception has two important properties: constancy and selectivity. By constancy is understood the preservation of of the obtained integral image during change of sensations connected with change of distance, the observed foreshortening, illumination and other circumstances of observation of an object. By selectivity is understood the distinguishing of those details and properties of the integral image which are most essential from the point of view of the goal facing man at the moment of perception.

To what extent is is right to transfer concepts from the area of psychology into the area of technology? The substantially different level of complexity of modern technological devices and living organisms and differences in their elementary base and structural organization permit at the present time copying nature only on the functional level. This circumstance sometimes leads to a shift of concepts and the appearance of analogous terms, speculative interpretation of which can be excluded only by precise definitions of their meaning.

Let us agree to consider that the processes of "sensation" are accomplished for purposes of feedback of automated control systems, tracking systems and sensitive manipulators.

Problems in the creation of robots, in particular of industrial robots with sensitivity and underwater and locomotion robots, lead to a need to organize more profound logical processing of data arriving from sensor elements. There already is a technical base for that in the form of digital computers and hybrid computing systems.

The use of digital computers and hybrid computing systems in an information channel for the purpose of suppressing noise and eliminating systematic component errors of measurement by means, for example, of the least-squares method, Kalman filtration, or by applying more complex filters, means improvement of the process of sensation as a reflection of individual properties of the observed environment. The new quality of the information process--perception of the environment--can be considered as arising in systems in which the computational-logical data processing is directed toward the construction of an integral "image," or, in other words, a model of the environment. In that, and only in that, sense we will use the term "perception" with respect to technical systems, noting the complete divergence of technical processes from the psychological.

It should be noted that there are two classes of problems in the construction of a model of the environment. The first class is connected with work in a well-organized medium. This includes models of the environment of most industrial and mobile robots functioning in a medium adapted to the activity of man (shops, staircases, corridors, etc). Processes of perception of the environment in that case can depend substantially on apriori information about its properties (plan, description of objects and their properties) and on especially organized information signals (orientation markers and classifier markers, etc). Questions about the construction of the information system of such robots must be solved in a complex with questions about the informational organization of the environment to maximally simplify and facilitate operation of the system as a whole.

The second class of problems in constructing a model of the environment is connected with work in a previously unknown or poorly known medium. Such problems arise during the creation of mobile robots, for example, intended for work under water or on previously unknown terrain. Such systems can be created only when specialization of the robot's motor activity is taken into account. Subordination of the information system of a robot to the solution of a narrowly specialized task of motion permits, when the system is correctly organized, sharply limiting the information flow and simplifying the data processing logic.

Both classes of problems in the construction of a model of the environment can be solved by using special selectivity of perception, built into a system in the stage of its formation. We will present examples of such systems.

In 1974 the author of the present article together with Yu. V. Tyuring simulated on a display system of the Institute of Applied Mathematics of the USSR Academy of Sciences a perception system of the first class. Selected as the object of investigation was a manipulator with a three-stage tactile sensor device in the grab, which permitted determining the direction of the normal to the contact surface. The problem consisted in the creation of a system of perception of form and dimensions of a certain part in order to paint it later. The discipline of the environment was expressed in the following:

--the part was at a known place in space and had dimensions convenient for a robot;
--the surface of the part could consist only of parts of spheres, cylinders, cones and planes.

Consideration of these limitations greatly simplified the algorithm of perception and the construction of movements. In particular, the known location of the part permitted organizing a simple algorithm for the start of feeling, and the possible set of forms of faces, limited by second-order surfaces and a plane, sharply simplified the rules of feeling and the structure of data describing the model of the part.

Those surfaces permit determining their class from the direction of the normal at three points. Therefore the rule of feeling consisted in determining the class of the surface at the three points, separated from one another by a distance determined by the resolution of the sensor device, and subsequent scanning along a certain curve in order to search for a boundary of a surface of the given class. After the boundary has been found, further feeling was accomplished along the found boundary to its closure. Each boundary was felt on two sides to search for points of branching. The search for unconnected surfaces was made by checking the constancy of form within a given boundary according to some rule.

A model of the part was stored in the memory of a digital computer in the form of a search for discovered surfaces and files of points of surface boundaries. Also created were programs of access to the model of the part, programs which permit, in particular, constructing its image on a display screen.

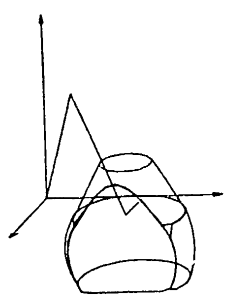


Figure 1. Image of model of a part on a display screen.

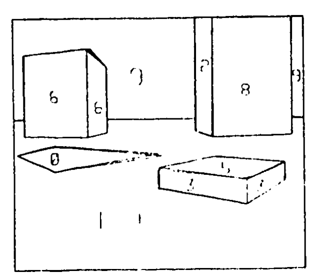


Figure 2. Display screen image of obstacles and the result of their perception.

Presented on Figure 1 is a representation, constructed on a display, of the process of feeling a part representing a sphere and a cone truncated by planes. The three lines starting from the origin of the coordinates represent a manipulator with a tactile sensor on its end. Each contact permits finding the direction of the normal to the surface of the part at the contact point and from that information determining which investigaged surface it belongs to. At the moment depicted on Figure 1 the boundary between the cone and the plane is being checked on the side of the plane.

A system of perception of the second class was investigated jointly with S. N. Shibayev. In examining a model of the system of environment perception of a mobile robot using photometric and range-fijder measurements the following assumptions were made:

--a photometer measures the integral illumination of the space within the angle $0.5 \times 0.5^{\circ}$;
--a range finder and a photometer can be directed toward any point in space;
--measurements are made to determine the possible direction of motion;
--motion occurs on a surface with low brightness, on which obstacles can be disposed in the form of rocks and pits;
--the sky brightness is maximal.

It proved possible to limit the functions of the constructed system of environment perception by defining the boundaries of obstacle contours without revealing their form and structure within the contour. This greatly simplified the system, as it permitted organizing examination along the boundary of a region of constant brightness. Range-finder measurements have been made only at certain points of the medium in order to distinguish real changes of the felief from spots of light and shadows. A model of the medium was created by recalling arrays of points of boundaries of the "earth,", the sky and obstacles. Figure 2 illustrates the result of examination of the scene of the system of perception with an active photometer. The double line shows the precision of restoration of obstacle boundaries. The figures designate the apparent brightness of the regions. The sign "+" shows the region corresponding to the earth. An obstacle with zero brightness is a pit, for the recognition of which the use of a range finder is required.

Several principal aspects should be noted in the presented examples of models of technical perception systems.

1. In both the presented systems not the entire space but its most informative parts--the boundaries of the regions of equivalence--were actively investigated or felt. By the region of equivalence in the model of the environment is understood the region, all points of which have an identical value from the point of view of the goal of the system. In the first system the regions of equivalence were the regions of the investigated part described by a single equation of the surface, by which the organization

APPROVED FOR RELEASE: 2007/02/08: CIA-RDP82-00850R000300070045-2

of the painting of that region was determined. In the second system the regions of equivalence contained points belonging to obstacles, the "earth" and the "sky" influencing the selection of the route of motion.

2. The selection of the direction of displacement of the sensor element in space (the sensor of the normal over the part in the first example and of the photometer and range finder in the second) was made on the basis of the results of measurements at each point of the trajectory of the sensor element.

The active character of the process of perception is also emphasized in psychology. However, in technical systems the organization of active examinination is especially important, as it permits curtailing the requirements for speed of the processor of the system and the volume of its memory. In the examples under consideration, activity of examination was displayed in the selection of the direction and spacing of sensor displacement, the change of displacement spacing being produced in accordance with rules similar to those for change of spacing in methods of numerical integration of ordinary differential equations. In each section of smoothness of the boundary of regions of equivalence the spacing increased and after separation from the boundary a search was made for the point of its deviation and the direction of motion was determined.

The environment examination activity can consist not only of control of the examination trajectory but also of control of the sensor aperture and switching the composition of measurements (for example, the photometer-range-finder). In general the perception activity represents a multi-level process.

3. The organization of examination activity in the cited examples was purely reflex--for each measurement there was a certain corresponding local rule (independent of pre-history) for the selection of the following measurement.

The question arises of whether, in order to solve the problems facing us, we can limit ourselves to the solution of perception problems on the purely reflex level or the addition of "psychological" qualities still implicit to the end, such as will, purpose, etc, which make the reflex connection less rigid. Let us note that the organization of the reflex character of the activity is simple and convenient both on the apparatus level (in the form of finite automata) and on the probram level (in the form of a dictionary of "situation-effect") or of series of conditional transfers.

The answer to the posed question evidently involves the possibility of creating multilevel data processing systems. Experience in the Institute of Applied Mathematics of the USSR Academy of Sciences in the simulation of a multilevel information system of spacing apparatus based on an optical range finder, in which each level was realized on the principles of a finite automaton, showed that a hierarchic system of finite automata permits organizing the necessary large number of reactions of the system to the external situation. This provides a basis for stating that on the current level of development of robot technology the reflex method provides a solution of many problems in perception of the environment under conditions of the use of hierarchic systems.

4. In the cited examples of perception systems the principle of breaking robot activity down into two phases has been clearly maintained: the organization of space (creation of a model of the environment, the distinguishing of reference points and passages, decision making regarding the route, etc) and the execution of work (in the process of which a model of the environment is used). Such a breakdown exists, evidently, always as the processes of information processing and the processes of construction and execution of movements exist in parallel. However, the degree of their compatibility can be different. The greater the separation of those phases in time, the larger the volume of memory required for storage of the model of the environment and the more complete and substantiated the decision regarding movement can be.

The striving to reduce the requirements for the digital computer storage and the total time required for the work, and also the need to organize robot motion in a certain space leads to a desire to maximally combine in time the work of the information system and the system for the construction of movements. However, their formal combination, as experience in the development of a walking robot control system has shown, can lead to the emergence of dead-end situations in the system, connected with the small long-range interaction of plans. To overcome that shortcoming it is necessary to increase the long-range interaction of plans and organize close interaction of the information system and the system for construction of movements and other robot systems directly in the process of environment perception.

Increase of the long-range interaction of plans requires the use of telereceptors, complexity of measurements and a deeper attraction of apriori information to increase the distance of perception. The possibility of increasing the long-range interaction of sensor elements plays a very large role, of course. Range-finder and photosensitive matrices appear to be very promising for that purpose. Figure 3 shows an optical range finder, developed in the Leningrad Institute of Mechanics jointly with the Institute of Applied Mathematics of the USSR Academy of Sciences, which permits measuring the distance to the terrain and controlling the direction of the line of sight, and Figure 4 a technical vision system based on a matrix of photosensitive cells developed at Vilnius State University.

The need to organize interaction of systems led to the emergence of an additional type of activity in the process of perception, the content of which is reaction of the system of perception to the requests of other systems, the formation of its own inquiries, the switching of sensors on and off and the organization of recurrent examinations of the environment in the presence of other aspects. The development of the perception system thus became a part of the system for construction of movements and other

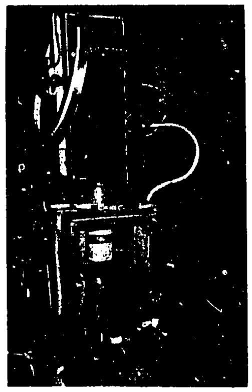


Figure 3. Optical range-finder.

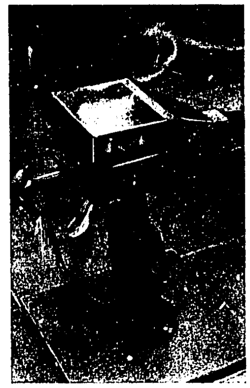


Figure 4. Technical vision system based on a photosensitive matrix.

robot systems (route selection, navigation, etc) and the structure of the entire robot control system on the whole was organized by means of special monitors as a function of the arriving information. The working rules of the monitors (the dynamic work distribution) have a purely reflex character and are realized by means of situation-effect apparatus or something similar. The situation at each moment is determined by the totality of external and internal signals of interacting elements of the system. Of great importance is the correlation between the velocity of motion and the rate of examination of the environment. For a greater rate of perception it is more advantageous to repeatedly examine the environment than to remember its characteristics.

Such an organization of the system of perception was first accomplished in 1974 in the task of simulating the overcoming by a walking apparatus of a complex cylindrical relief (Figure 5) and in subsequent years in the transition to the development of motion of a laboratory prototype of a walking apparatus with a range finder (Figure 6), obtained its confirmation and development.

Thus the problems of environment perception under consideration are determined by the need to create programmed equipment complexes allocated the possibility of active control of the process of perception in order to distinguish regions of recognition as regards the environment. Such complexes should be organized in the form of multilevel multicomponent systems with

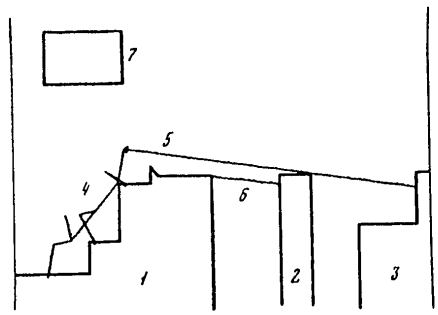


Figure 5. Display image of walking apparatus with range finder.

Key:
1-3 -- obstacles
4 -- walking apparatus
5 -- distance measuremen line
6 -- boundary of examined region
7 -- image in plane of reference contour and center of mass of walking apparatus

Figure 6. Laboratory prototype of walking apparatus.

reflex control on each level and dynamic change of the character of the interaction of individual components of the system. Methods of solution of those problems are clear even at the present time.

Together with that there is an entire series of problems, the ways to solve which are still not completely understood. We will mention some of them.

The problem of breaking a continuous flow of information down into required fragments, that is, the problem of message synthesis (the distinguishing of important characteristics) leads to a need to develop special devices --property detectors. The task of property detectors is to break complex forms down into simpler elements occupying an intermediate position between the goal of the figure and the mosaic of sensor element signals. Those intermediate elements form an "alphabet of perceptive units," the formation of which involves the essence of the task of perception and the possibilities of contemporary technological apparatus.

Apparently extremely important is the problem of the possibility of using reflex apparatus of the "situation-effect" dictionary to organize such functions as the selectivity of perception, the parallel character of information processing, programmed filtration and processes of concept formation. Preliminary estimates here have an extremely optimistic character.

An important problem is the struggle against noise on a low level of the perception system. The ordinary procedure often proves to be ineffective (especially under the conditions of a poor signal-to-noise ratio) and must be supplemented by a procedure for consideration of context and apriori information. This requires organization of interaction of the work of the highest levels of the system with the lowest.

Finally, the main problem is that of creating a perception system based on digital computers or a hybrid computer system with consideration of present-day possibilities of main-line logic, the use of pre- and postprocessors, homogeneous structures and associative memory. The greatest success here, evidently, should be expected on path of introduction of parallel analog processors into the lower levels of the environment perception system.

Problems of environment perception by robots have a universal character. However, the ways to solve them are different for different kinds of robots and are determined by the specifics of the problems solved by a robot. Therefore of great importance is experience in the construction of specific perception systems, which permits developing universal methods of solving this important technical problem.

[16-2174]

2174
CSO: 1863

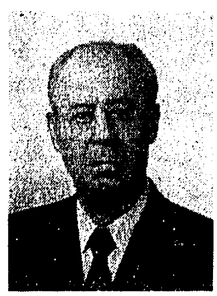
PERSONALITIES

KONSTANTIN NIKOLAYEVICH RUDNEV

Moscow PRIBORY I SISTEMY UPRAVLENIYA in Russian No 9, 1980 p 32

[Article in memoriam of K. N. Rudnev]

[Text] The staff, party committee, trade union committee and Komsomol committee of the Ministry of Instrument Construction, Automation Devices and Control Systems regret to inform the passage of Konstantin Nikolayevich Rudnev, at age 70, following a serious illness. He was a member of the CPSU Central Committee, deputy of the USSR Supreme Soviet, USSR minister of instrument construction, automation devices and control systems and Hero of Socialist Labor.



Konstantin Nikolayevich Rudnev was born in Tula in 1911. He was a member of the CPSU since 1941. In his more than 45 years of labor activities, Konstantin Nikolayevich trod the path from electrical fitter to great state leader.

After graduating from Tula Mechanical Institute in 1935 majoring in mechanical engineering, K. N. Rudnev worked in the scientific-research institution as a designer, head of testing and head of the design division. From 1939 through 1948, Konstantin Nikolayevich held the posts of chief engineer and director of several large enterprises and the scientific research institute of the defense industry.

In 1948, K. N. Rudnev was nominated chief of the Main Administration and a member of the staff of the Ministry of Armament; in 1952 he was made deputy minister of the USSR defense industry. Starting in 1958, K. N. Rudnev was chairman of the State committee of the USSR Council of Ministers on defense technology and USSR minister.

Starting in 1961, K. N. Rudnev worked a deputy chairman of the USSR Council of Ministers and chairman of the State committee on coordination of scientific research work and in 1965, he became USSR minister of instrument construction, automation devices and control systems.

Konstantin Nikolayevich Rudnev is known in our land as one of the great organizers of science and industry, a talented engineer, who greatly contributed to the development of the defense industry, instrument construction and computer technology, as well as the introduction of scientific achievements in production and development of economical methods and efficient forms of management of the national economy. Under his aegis, the ministry was first to realize a major economic experiment in transferring the entire instrument construction sector to a new system of planning and economic incentives.

Konstantin Nikolayevich Rudnev was highly cultured, had high principles, and was sensitive and attentive to people; he knew how to come to the aid in resolving difficult questions in the life of his staff and individual comrades.

The party and the government highly valued the services of Konstantin Nikolayevich. He was awarded the title of Hero of Scoialist Labor. He received six Orders of Lenin, the order of the October Revolution, two Orders of the Red Banner of Labor, the Order of World War II Second Degree and many medals.

Konstantin Nikolayevich Rudnev was elected a deputy to the USSR Supreme Soviet's sixth through ninths sessions. At the 22nd, 23rd, 24th and 25th congresses of the party he was elected a member of the Central Committee of the CPSU.

The shining memory of a true son of our homeland, a firm Leninist, our dear Konstantin Nikolayevich Rudnev will forever remain in our hearts.

[43-8617]

8617
CSO: 1863

- END -