JPRS L/9540

10 February 1981

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 5/81)

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.

JPRS L/9540

10 February 1981

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 5/81)

CONTENTS

HARDWARE

DEVELOPING ELEMENTS FOR LOW-POWERED DIGITAL EQUIPMENT

Moscow ELEMENTY MIKROMOSHCHNYKH TSIFROVYKH USTROYSTV in Russian 1980 pp 2-5, 53-56

[Annotation, table of contents, foreword, and bibliography of book "Elements of Micro-Powered Digital Devices," by L.S. Gorn and B.I. Khazanov, Atomizdat, 56 pages]

[Text] This book is devoted to elements of economical digital measuring devices. There is a brief review of the fundamental properties of these elements and a more detailed consideration of structures using complementary MDP [metal dielectric semiconductor] transistors. The characteristics of base KMDP [complementary metal-oxide dielectric semiconductor] elements, integrated circuits with low and medium levels of integration, and pulse shapers and generators with KMDP structures. Typical examples of large integrated systems with KMDP structures are considered: internal memory units and several types of microprocessors.

The book has five tables, 18 illustrations, and 37 bibliographic entries.

Table of Contents | Page

Foreword

One of the most important and difficult problems of the many different problems facing the developers of instruments for radiation measurement is constructing a device with the least possible power input. A decrease in power input is necessary both to improve the reliability of equipment and when making instruments that have limited power supply.

A large group of articles in atomic instrument making are classified as "portable" instruments receiving power from small storage cells and batteries. Among them are instruments to evaluate the radiation situation (portable radiometers), dosimetric monitoring instruments (personnel dosimeters, dose rate meters), and equipment used in prospecting for and exploring radioactive and nonradioactive ore. In these instruments reducing power consumption makes it possible to prolong their work without changing the power supply or lowering the capacity of these sources by reducing the weight and dimensions of the instruments. The problem of minimizing power input is one of the key problems in making aerospace equipment, in particular instruments placed in artificial earth satellites and automatic interplanetary stations and used to investigate the characteristics of radiation flows in near-earth and interplanetary space. Reducing the power consumption of instruments makes it possible to put a larger amount of equipment in the spacecraft and receive more varied and reliable information on the processes or phenomena under study as the result of these very expensive experiments.

The reliability of equipment depends largely on its power consumption. Reducing dissipated power leads to a reduction in the difference between the temperature of the elements and the temperature of the environment and decreases the frequency of malfunctions in electronic components. It is especially important to mitigate heat condition in equipment that has a high density of radio elements. In large integrated circuits, for example, reducing dissipated power by 0.05 watts leads to a decrease in the temperature of the microcircuit that averages 10 degrees C and a doubling of its reliability.

To lower the power consumed by equipment used in radiation measurements requires comprehensive improvement in the characteristics of the devices that make up this equipment and, in particular, improving the efficiency of secondary power sources (transformers), developing economical analog stages (pulse amplifiers, shapers, amplitude discriminators, and the like), and making digital devices that have low power input. Each of these areas of work has its own methods of problem-solving and its own specific features.

For a particular instrument the resulting increase in economy when the power consumed by individual stages is reduced may differ and depend on the structural diagram of the equipment.

In recent years digital stages (cascades) have played an increasing part in equipment produced and the reduction in the power consumed by them is beginning to define the general characteristics of instruments. Therefore, equipment developers are focusing increasing attention on making economical (or, as they are frequently called, "micro-powered") elements for digital devices.

From the standpoint of continued development of the engineering of nuclear instrument making the problem of reducing power input should also be resolved first of all for storage and processing devices in digital form. It is with these devices that the prospects for building a new generation of equipment that has greater information value, lower error, and more convenient operation are tied.

Characteristics of Building Equipment with Microprocessor Elements

Like the microcircuits considered in the first sections of the book, owing to their highly refined design, functional, and circuitry features large integrated systems of microprocessor families can be viewed as elements of the equipment being built. But as already noted, these elements have significant features, the fundamental one of which is universality, which is achieved because they are programmable. The different varieties of microprocessors make it possible to build various types of equipment, both relatively simple and inexpensive equipment as well as complex information-measurement systems, going all the way to multiprocessor systems.

Construction of equipment using these elements makes it possible to give the equipment capabilities which are practically unattainable where logical circuits with a low level of integration are used. Among these capabilities are data processing in all stages of receiving and converting information; elaborate control functions for both individual devices and an entire system; adaptability (that is, reorganization of structure and modification of characteristics depending on results obtained or measurement conditions); an increase in precision which is achieved by, in particular, automatic calibration, error recording, and linearization of characteristics; automatic identification of trouble (self-monitoring); outputting data in the form most convenient to the operator, and others.

The use of these elements significantly improves design and technological characteristics. It makes it possible to reduce dimensions and weight, improve reliability, significantly cut the labor-intensiveness of equipment manufacture, lower its cost, and simplify (with the essential base) the design of instruments and systems.

Therefore KMDP large integrated systems may be considered the base elements of contemporary, economical program-controlled equipment.

Bibliography

1. Adams, Smith, "Comparison of Complexes of Microprocessor Sections," ELECTRONICS, 1978, Vol 51, No 16, pp 25-34; No 17, pp 32-43.

2. Al'tman, "Current Situation with Development and Production of Complementary Metallic Oxide Semiconductor Integrated Systems and Large Integrated Systems," ELECTRONICS, 1975, Vol 48, No 10, pp 25-41.

3. Barilko, Sh.,I., Gorn, L. S., and Khazanov, B. I., "Economical Scaler on Magnetic Cores," PRIBORY I TEKHNIKA EKSPERIMENTA, 1959, No 2, pp 91, 92.

4. Bukreyev, I. N., Mansurov, B. M., and Goryachev, V. I., "Mikroelektronnyye Skhemy Tsifrovykh Ustroystv" [Microelectronic Circuits of Digital Machines], Moscow, Sovetskoye Radio, 1973, 247 pp.

5. Valiyev, K. A., Karamzinskiy, A. N., and Korolev, M. A., "Tsifrovyye Integral'nyye Skhemy na MDP-Tranzistorakh" [Digital Integrated Circuits on MOS Transistors], Moscow, Sovetskoye Radio, 1971, 384 pp.

6. Gafarov, P. M., Minkov, Yu. V., Solomonenko, V. I., and Urosov, V. N., "Memory Unit with 4,096 Bits of Capacity on Single-Transistor Memory Cells," ELEKTRONNAYA PROMYSHLENNOST', 1978, No 8(68), pp 5-11.

7. Gorn, L. S., and Khazanov, B. I., "Uzly Radiometricheskoy Apparatury na Integral'nykh Skhemakh" [Assemblies of Radiometric Apparatus on Integrated Circuits], Moscow, Atomizdat, 1973, 228 pp.

8. Dshkhunyan, V. A., Kovalenko, S. S., and Mashkevich, P. R., "The BIS [Large Integrated Circuit] K587 Low-Powered Microprocessor Unit on Complementary Metallic Oxide Semiconductor Transistors," ELEKTRONNAYA PROMYSHLENNOST', 1978, No 5(65), pp 15-19.

9. Zakharov, D. S., Klimashov, A. A., and Sulimov, Yu. V., "Construction of Pulsed Cascades Using Integrated Circuits with CMOS Structure," VOPROSY ATOMNOY NAUKI I TEKHNIKI. YADERNOYE PRIBOROSTROYENIYE, Vyp 38, Moscow, Atomizdat, 1979, pp 100-108.

10. Karamzinskiy, A. N. (editor), "Integral'nyye Skhemy na MDP-Priborakh" [Integrated Circuits in MOS Instruments], Moscow, Mir, 1975, 345 pp, (translated from English).

11. Yen, "Construction of Pulsed Circuits on the Basis of Standard CMOS Triggers," ELECTRONICS, 1975, Vol 48, No 6, pp 51-56.

12. Krayzmer, L. P., "Ustroystva Khraneniya Diskretnoy Informatsii" [Units for Storage of Discrete Information], Leningrad, Energiya, 1969, 312 pp.

13. Lipman, R. A., "Magnitnyye Nakopitel'nyye Schetchiki" [Magnetic Storage Counters], Moscow-Leningrad, Energiya, 1967, 144 pp.

14. Mayers, "The Effect of Nuclear Emission on Semiconductor Instruments," ELECTRONICS, 1978, Vol 51, No 6, pp 53-57.

15. "Mikromoshchnaya Elektronika" [Low-Powered Electronics], Moscow, Sovetskoye Radio, 1967, 288 pp (translated from English, edited by Ye. I. Gal'perin).

16. Valiyev, K. A., Dyagilev, V. N., Lebedev, V. I., and Lubashevskiy, A. V., "Mikromoshchnyye Integral'nyye Skhemy" [Low-Powered Integrated Circuits], Moscow, Sovetskoye Radio, 1975, 250 pp.

17. "The Elektronika NTs-03 Microcomputer," ELEKTRONNAYA PROMYSHLENNOST', 1978, No 5(65), pp 54-55.

18. Afanas'yev, M. M., Bagradcv, B. A., Gusakov, V. M., et al, "Memory Unit with Potential Data Reading Capacity of 256 Bits," ELEKTRONNAYA PROMYSHLENNOST', 1977, No 3(57), pp 48-50.

19. Ramachandran, "Low-Powered CMOS Semipermanent Memory Unit with Erasure, Operated from a Single Voltage Source," ELECTRONICS, 1978, Vol 51, No 14, pp 35-42.

20. Khart, Slob, and Vulms, "Integrated Injection Logic -- a New Area in the Field of Bipolar Large Integrated Circuits," ELECTRONICS, 1974, Vol 47, No 20, pp 37-47.

21. Khorton, Engleyd, and Makgi, "Integrated Injection Logic -- an Important Advance in the Technology of Bipolar Integrated Circuits," ELECTRONICS, 1975, Vol 38, No 3, pp 24-34.

22. Bobrov, A. I., Gusakov, V. M., Krasil'nikov, A. A., et al, "Digital Integrated Circuits on CMOS Transistors," ELEKTRONNAYA PROMYSHLENOST', 1974, No 10, p 28.

23. Sharugin, I. I., "The Design of Digital Devices on Injection Logic Circuits," MIKROELETRONIKA, 1978, Vol 7, No 3, pp 247-260.

24. Englin, "Pulse Width Modulator on One CMOS Circuit," ELECTRONICS, 1977, Vol 50, No 13, p 55.

25. Arare, R. K., "Charge Coupled Devices -- Basic Operation and Principles," MICROELECTRON. AND RELIABL., 1976, Vol 15, No 5, pp 475-483.

26. Flammiger, F., and Schindler, D., "Monostable CMOS-Multivibration," RADIO FERNSEHEN ELEKTR, 1977, Vol 26, No 17, pp 563-564.

27. "4-Bit Processor Slice, CMOS or STTL. Macrologic," ELECTRONIC DESIGN, 1977, Vol 25, No 21, p 178.

28. "8-Bit Microprocessor, CMOS, CDP 1802," Ibid., p 130.

29. Herzog, J., "Stand und Fortschritte auf dem Gebit der CMOS-Bauelementetechnik," RADIO FERNSEHEN ELEKTRONIK, 1978, Vol 27, No 3, pp 160-163.

30. "Katalog Sowjetischer Hadleiter-Bauelemente Integrierte Scaltungen," Ibid., 1976, Vol 25, No 19/20, pp 645-646.

31. Kohlhaas, P., "Controlling Potential Static Charge Problems," ELECTRON. PACKAGE AND PRODUCE, 1977, Vol 17, No 1, pp 71-73.

32. "Memory Design Handbook," Intel Corporation, Santa Clara, California, USA, 1977.

33. "RCA First in CMOS," ELECTRONIC DESIGN, 1977, Vol 25, No 22 (advertisement on cover).

34. Turinsky, G., "Schaltungbeispiele mit CMOS-Schaltkreisen," RADIO FERNSEHEN ELEKTRONIK,1978, Vol 27, No 9, pp 551-553.

35. Turinsky, G., "Zusammenschaltung von Digitalen CMOS-Schaltkreisen mit anderen Logik-Familien," RADIO FERNSEHN ELEKTRONIK, 1977, Vol 26, No 3, pp 76-78.

36. Young, A., "Getting To Know the COSMAC Microprocessor Is Simple," ELECTRONIC DESIGN, 1976, Vol 24, No 22, pp 136-145.

37. Zimmerman, T. A., Allen, R. A., and Jacobs, R. W., "Digital Charge-Couple Logic (DCCL)," IEEE J. SOLID STATE CIRCUITS, 1977, Vol 12, No 5, pp 473-485.

[80-11176]

11,176
CSO: 1863

K587 SERIES DESIGNED FOR ELEKTRONIKA NTS-03

Moscow ELEMENTY MIKROMOSHCHNYKH TSIFROVYKH USTROYSTV in Russian 1980 pp 43-46

[Chapter of book "Elements of Micro-Powered Digital Devices", by L.S. Gorn and B.I. Khazanov: "Microprocessors on KMDP Structures"]

[Excerpt] Microprocessor Sections [MPS's]

There are two ways used by designers of large integrated circuits to make multiprocessor sections using KMDP [complementary metal-oxide-dielectric semiconductor] technology, as well as other types of technology. The first way involves making multiprocessor sections whose architecture contains all the basic structural elements (with the exception of a system control circuit) necessary to construct a functionally complete central processor but planned to process words with small bit configuration (most often four-digit) and allowing consolidation with the required length of words being processed without additional hardware in the central processor. When building systems using such microprocessor sections the designer has an opportunity to build up the bit configuration of the central processor arbitrarily, but will be limited to the framework of the architecture of the microprocessor sections selected. The K587IK2 multiprocessor section produced in the USSR will be considered below as an example of such multiprocessor sections.

The second way involves development and manufacture of multiprocessor sections that are not functionally complete in the form of large integrated circuits. In this case each multiprocessor section is a microprocessor of low bit configuration and the structure of the multiprocessor is broken down into the key structural elements such as the arithmetic-logical unit, the high-speed memory registers, the memory address registers, and others and these elements are produced in the form of large integrated circuits that permit both an expansion of bit configuration (element of each type) and their consolidation into various combinations to construct a central processor with the required architecture. Although it appears to involve the greatest complexity in building systems, the use of a family of this type of large integrated circuit provides the greatest design flexibility and imposes the fewest constraints. A typical example of such a family of large integrated circuits is the Fairchild Company's Series 4700 microcircuits based on KMDP technology (the MACROLOGIC family) [2].

The K587IK2 Microprocessor Section

The diagram of the multiprocessor section includes the structural elements necessary for computation and data exchange with other units of the system: the arithmetic-logical unit, the general-purpose register blocks which perform the function of high-speed internal memory, working register (accumulator), shift block, state register, three data input-output channels with exchange assemblies, a synchronization block, an expansion block, and a microinstruction receiving register with decoder.

The combination-type arithmetic-logical unit is planned to perform conventional arithmetic (incrementation of one operand, addition and subtraction of two operands) and logical (AND-OR, EXCLUSIVE OR, INVERSION) operations. Any of the general-purpose registers or the accumulator can be the source of operands. The result is also moved to one of the general-purpose registers or accumulator. The general-purpose register block contains eight four-digit registers.

Logical or cyclical shift operations are executed in the shift block. This block also generates the three state signals of the arithmetic-logical unit: the sign of the results, zero results, and expansion (the presence of units in shifts). These signals, acting on the corresponding triggers of the state register, orient them to the appropriate state. In addition to these triggers the state register also has one other trigger, the overflow trigger.

Data exchange with units exterior to the multiprocessor sections is accomplished by means of three channels and the exchange control circuits connected to them. All the channels are figured for two-directional exchange (that is, for receiving and outputting data). Channels K1 and K2 are used only for exchange with the accumulator; channel K3 receives data into the accumulator but outputs it from the state register. The exchange mode (receive or output) depends on the presence of the appropriate control signals fed to the exchange circuits from external units or issued by exchange units to the external units.

The microinstruction receiving register stores the code of the microinstruction (12 bits) to be executed and the decoder produces at its output the logical level which, acting on the remaining blocks of the multiprocessor sections and their connected key elements, insure performance of the required operations in the presence of synchronous signals from the synchronization block. The total number of instructions that control the operation of a multiprocessor section is 168. Two low-order bits of the microinstruction determine its format. Table 5 below gives the distribution of microinstruction fields for four possible formats. The content of bit positions 2, 3, and 4 of all formats code the operations of the arithmetic-logical unit; fields P_i and P_j contain the codes of the registers of the sources of operands and the receiver of the results of the operations, while the fields in bit positions 8, 9, and 10 give the codes of the operations of the shift, exchange, and source and data receiver. Bit positions 5, 6, 7, and 8 of the third format contain a constant and bit position 11 indicates the need to record the states in the state register and output the content of this register to channel K3.

Table 5. Formats of Microinstruction of K587IK2 Multiprocessor Section

Разряды формата(a)		Разряды операционной части микрокоманды (b)									
0	1	2	3	4	5	6	7	8	9	10	11
0	0	КОП	АЛУ(c)			P_i			P_j		ВС(d)
1	0	КОП	АЛУ			P_i		КОП	СДВ(e)		ВС
0	1	КОП	АЛУ			Константа (i)				ИП(h)	ВС
1	1	КОП	АЛУ			P_i		КОМ(f)	ОБМ		ВС

Key: (a) Bit Positions of Format;
(b) Bit Positions of Operations part of Microinstruction;
(c) Arithmetical-Logical Unit Op Code Field;
(d) Output Contents;
(e) Shift Register Op Code Field;
(f) Exchange Op Code Field
(h) Data Source and Receiver;
(i) Constant.

The microinstructions of the first format initiates register-register operations of the type (P_j) F (P_i) where F is a function of the arithmetic-logical unit in conformity with the op code. Eight different functions correspond to the three-position op code field of the arithmetic-logical unit: incrementation of the content of P_j and writing the result in P_j and the accumulator (we will write this operation concisely as $(P_j) + 1 \rightarrow P_j$, A); subtraction $(P_j) - (P_i) \rightarrow P_j$,A; copying $(P_j) \rightarrow$ A; logical "AND" (P_i) $(P_j) \rightarrow P_j$, A; addition $(P_i) + (P_j) \rightarrow P_j$,A; logical "OR" (P_i) $(P_j) \rightarrow P_j$, A; nonequivalence (P_i) $(P_j) \rightarrow P_j$, A; and copying $(P_i) \rightarrow P_j$, A.

The microinstructions of the second format cause "accumulator-register" operations which may be conditionally described by the relationship [(A)F(P_i)] CDA $\rightarrow$ A, where F as before is one of the functions of the arithmetic-logical unit and CDB is a shift operation on the result of the operation within brackets. There are four shift functions: logical shift to the left and right and cyclical shift to the left and right. The four other code combinations of the shift op code field code the following operations: inversion of the results, execution of operations F with carryover or borrowing, execution of operations F without a shift, and recording the result in the pair A, P_i or only in A.

Operations with the constant (third format) contained in positions 5, 6, 7, and 8 of the microinstruction can be represented in the form (CONST) F (I[SR]) $\rightarrow$ A, SR [state register], where I[SR] is the source of the second operand determined by the code of the data contained in positions 9 and 10. This source may be channel K3, the state register, or the accumulator. The result of these operations is always entered in the accumulator, and in two cases in the state register also.

The microinstructions of the fourth format initiate execution of operations in the arithmetic-logical unit (in conformity with the content of the op code field of the unit) on the operands stored in the accumulator and P_1; the result of the operations is always loaded into the accumulator. Before execution of these operations copies are made from channels K_1 or K_2 to the accumulator; the operations are completed by copying from the accumulator into one of the channels K_1 or K_2. The concrete form of the copies depends on the code contained in the exchange op code field. The list of microinstructions we have reviewed provides broad potential for using microprocessor sections to process data in information-measurement systems.

The sections make it possible to consolidate word formats (multiples of four) up to 32 bit positions in the multiposition processor without additional communication elements. In this case similar control lines are connected in parallel, while lines from the outputs of the channels are joined into common lines with a bit configuration that is a multiple of four. The distribution of carryovers and shift positions from section to section during performance of arithmetic-logical and shift operations is insured by an appropriate connection of communications circuits of the expansion block.

In addition to the K587IK2 section that we have considered, the K587 family includes several other large integrated circuits with microprogram control whose use simplifies the design of a microprocessor system: the K587IK1 self-contained module for processing and switching digital data, which is designed to organize an interface; the K587IK3 self-contained hardward multiplication module; the K587RP1 address generator for setting up microprogram control; and the K530AP2 two-directional line former. Each of these large integrated circuits dissipates less than 10 milliwatts of power with a supply voltage of nine volts. This makes it possible to build systems that consist of a fairly large number of large integrated circuits with a power input at the level of 0.1-1 watt. It should be noted, however, that these large integrated circuits are intended chiefly for specific use in the Elektronika NTs-03 microcomputer [17] and when they are used for simpler program-controlled instruments or systems difficulties may arise. For example, the general-purpose registers of the microprocessor section are hard to use as instruction counters or other system registers such as stack index indicator and the like as envisioned in many multiprocessor sections under development recently. The K587RP1 microprogram control large integrated circuit is made so that it is practically impossible for a user to program it. This significantly limits the possibility of developing a problem-oriented language for the system being designed.

[80-11176]

11,176
CSO: 1863

K176RU2 STATIC RAM

Moscow ELEMENTY MIKROMOSHCHNYKH TSIFROVYKH USTROYSTV in Russian 1980 pp 32-36

[Chapter of book "Elements of Micro-Powered Digital Devices", by L.S. Born and B.I. Khazanov: "Static Memory Units on KMDP Structures"]

[Excerpts] Large integrated circuits of main memory units with potential data reading on KMDP [complementary metal-oxide-dielectric-semiconductor] structures have been developed and are being produced to build various devices with centralized memory and low power input. Examples of such microcircuits are the K164RU2 and K176RU2 large integrated circuits with memory organization of 256 x 1 bits and the K564RU2 with 1024 x 1 bits (USSR).

These microcircuits are functionally complete devices that contain, in addition to the matrix of memory elements, address decoders, amplifiers of the signal being read, and control stages (cascades). They permit relatively simple construction of internal memory units with large memory size.

Figure 9 below shows the schematic diagram of the 176RU2 large integrated circuit. It is typical of the microcircuits mentioned above. The memory is based on a matrix of memory elements that has 16 rows and 16 columns. The four low-order bits of the address A_0 - A_3 select one of the matrix rows through decoder 1, while the four high-order bits A_4 - A_7 select one of the columns through decoder 2. The output read amplifier is an invertor with the switch boundary of transmission characteristic U_c shifted in the direction of lower voltages. A complex invertor with three states is used as an output stage (cascade), which makes it possible to join the microcircuits directly at the output while increasing the volume of the internal memory unit. The microcircuit outputs both the direct output signal on output line D and the inverse signal on line $\bar{D}$.

The control unit, which receives the "crystal selection," "write-read," and "data input" signals determines the working mode of the microcircuit. When level 1 is set on the "crystal selection" line the microcircuit is isolated from the data input and output lines; only if the level on this line is equal to 0 is it possible for control elements to influence other cascades and to read stored data. At level 1 on the write-read line data is

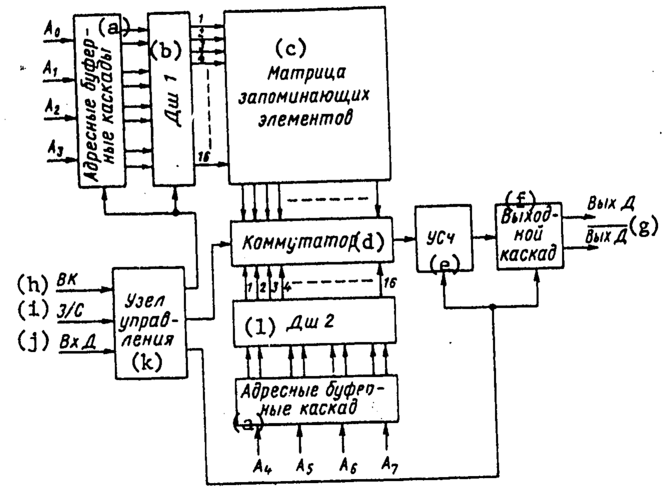


Figure 9. Schematic Diagram of K176RU2 Large Integrated Circuit of Internal Memory Unit

Kay:
- (a) Buffered address cascades;
- (b) Decoder 1;
- (c) Matrix of memory elements;
- (d) Multiplexor;
- (e) Read amplifier;
- (f) Output cascades;
- (g) Outputs D and D;
- (h) Crystal selection;
- (i) Write-read;
- (j) Data input;
- (k) Control unit;
- (l) Decoder 2.

---

read from the cell selected and when it is at 1 it is written into this cell; after the data is recorded the state of the cell is determined by the level of the signal (0 or 1) on the output D line.

The microcircuits we have considered permit the construction of a static internal memory unit with very low power input. The K176RU2 large integrated circuit, for example, can work at E_n = 4-15 volts and at E_n = 9 volts uses no more than 0.1 milliwatt in the storage mode (that is, less than 0.4 microwatt per bit of stored information). The Intel-5101 large integrated circuit works at E_n = 2-5.25 volts and consumes no more than 15 microamperes in the storage mode (that is, less than 70 microwatts or 0.07 microwatts per bit of stored information).

At the same time the microcircuits have entirely satisfactory time characteristics. The K176RU2 microcircuit has an access time of not more than 0.4 microseconds and a write not greater than 0.2 microseconds. The Intel-5101 microcircuit has an access time of less than 0.65 microseconds and a write time of less than 0.4 microseconds.

Figure 12 [not reproduced] gives time diagrams of the work of the K176RU2 microcircuit in the read and write mode; they are similar to the work diagrams of other types of large integrated circuits for internal memory units.

[80-11176]

11,176
CSO: 1863

SMALL AND MEDIUM SCALE INTEGRATED MICROCIRCUITS

Moscow ELEMENTY MIKROMOSHCHNYKH TSIFROVYKH USTROYSTV in Russian 1980 pp 14-19

[Chapter of book "Elements of Micro-Powered Digital Devices", by L.S. Gorn and B.I. Khazanov: "Small and Medium Scale Integrated Microcircuits"]

[Excerpts] Small scale integrated circuits.

Table 1 below shows typical microcircuits with KMDP [complementary metal-oxide-dielectric semiconductor] structure produced in the USSR and abroad. Included among them are various logical cascades, keys, universal logical cascades, cascades for interlocking KMDP and TTL gates, and triggers. Such microcircuits contain about 50 elements in one body.

Table 1. Typical Microcircuits on KMDP Structures

Function Performed, Number of Elements	Name of Microcircuit RCA Company	Series K176
Two 3 "OR-NOT" and an Invertor	CD4000	K176LP4
Four 2 "OR-NOT"	CD4001	K176LE5
Two 4 "OR-NOT"	CD4002	K176LE6
Three 3 "OR-NOT"	CD4025	K176LE10
Two 4 "OR-NOT" and Two Invertors	-	K176LP11
Four 2 "AND-NOT"	CD4011	K176LA7
Two 4 "AND-NOT"	CD4012	K176LA8
Three 3 "AND-NOT"	CD4023	K176LA9
Two 4 "AND-NOT" and Two Invertors	-	K176LP12
Four "AND-OR"	CD4019	-
Nine "AND" and Invertor	-	K176LI1
Four "EXCLUSIVE OR"	CD4030	K176LP2
Universal Element (two pairs of transistors and one invertor)	CD4007	K176LP1
Five Interlocking Cascades with Inversion	-	K176PU1
Six Interlocking Cascades with Inversion	CD4009	K176PU2

Function Performed, Number of Elements	Name of Microcircuit RCA Company	Series K176
Six Interlocking Cascades without Inversion	CD4010	K176PU3
Two D-Triggers	CD4013	K176PM2
Two J-K Triggers	CD4027	K176TV1
Four Two-Directional Keys	CD4016	K176KT1
Four RS Triggers with Output of States for Resolving Signal or High Output Resistance in Absence of Signal; R- and S-Signals 0 → 1	CD4043	-
Same, but R- and S-Signals 1 → 0	CD4044	-
Three 3 "AND-OR-NOT"	-	K176LS1

Table 3. Designation and Characteristics of Certain Series 4000 Microcircuits

Group of Integrated Circuits	Standard Designation	Function, Structure	Triggering	Average Power Dissipation, μw
Shift Registers	4015 (IR2)	Two 4-digit registers containing 4 D-triggers with sequential data recording and parallel, asynchronous data output from each trigger; mass reset at 0	0 → 1	10
	4035 (IR9)	Four-digit register with parallel data feed and synchronous output from each trigger; mass reset at 0	0 → 1	10
	4021	Eight-digit register with sequential or asynchronous parallel data recording and data output from the last 3 triggers	0 → 1	10
	4014	Same, but with synchronous parallel data recording	0 → 1	10
	4006 (IR10)	Two 4-digit or 2 5-digit registers with sequential recording and output of data from the fourth and fifth (in two registers) triggers	0 → 1	10
	4031 (IR4)	Four-digit register		

Group of Integrated Circuits	Standard Designation	Function, Structure	Triggering	Average Power Dissipation, μw
Pulse Counter	4024 (IYe1)	Seven-digit binary counter with output of state of each trigger; mass reset at 0	1 → 0	10
	4040	Twelve-digit binary counter with output of state of each trigger; mass reset at 0	1 → 0	20
	4020	14-digit binary counter with output of state of each trigger, except second and third; mass reset at 0	1 → 0	20
	4060	Same counter, but with pulse generator	–	20
	4017 (IYe8)	Decimal ring counter on five triggers with output of state in decimal position code	0 → 1	10
	4022 (IYe9)	Ring counter on four triggers with output of eight in octal position code	0 → 1	10
	4018	Pulse counter with assignment of necessary conversion factor between 3 and 10	0 → 1	10
	4029	Four-digit binary or decimal counter with assignment of counting volume and output of state of each trigger	0 → 1	10
	4059	Four-decade frequency divider with assignment of counting volume	0 → 1	10
Decoder	4028 (ID1)	Convertor from binary-coded decimal code to decimal position code; in selected position high level at output, but low level in other positions	–	10
	4055	Convertor of binary-coded decimal code into septal position code to control a liquid crystal; modulation of output level by low frequency signal (30–200 Hz)	–	15

Group of Integrated Circuits	Standard Desig-nation	Function, Structure	Trigger-ing	Average Power Dissipa-tion, μw
	4056	Same, without modulation but with internal storage of data	-	15
Multi-plexors	4051	Eight-channel multiplexor connecting the output with one of the inputs; control of binary code	-	1
	4052 (KP1)	Two similar 4-channel multi-plexors	-	1
	4053	Three similar 2-channel multi-plexors	-	1
Arith-metic Unit	4008 (IM1)	Four-digit full summator	-	10

At the present time the manufacturers of KMDP microcircuits produce a fairly broad assortment of medium scale integrated circuits. Each year new devices are brought into production. To illustrate the capabilities of these microsystems, Table 3 (above) gives a list of the devices included in series 4000 (in addition to those listed, other integrated circuits are included in the series; in addition integrated circuits of series 4500, 4800, and others are produced abroad). Some of these microcircuits are included in the K164, K176, and K564 series produced in the USSR (in Table 3 their standard designations are given in parentheses). In addition to them other devices are produced in these series of integrated circuits which do not have analogs in the 4000 series.

[80-11176]

11,176
CSO: 1863

YeS, SM AND ELEKTRONIKA SPECIFICATIONS

Moscow EVM DLYA VSEKH in Russian 1980 signed to press 14 Nov 79 pp 41-42

[Excerpts from book by Vladimir Aleksandrovich Myasnikov, Sergey Aleksandrovich Mayorov and Gennadiy Ivanovich Novikov, Izdatel'stvo "Znaniye", 77,380 copies, 192 pages]

[Excerpts] Table 2. Technical Specifications of YeS [Unified Series] Computers

Specification	Computer Model						
	1020	1022	1030	1033	1040	1050	1060
Mean speed, thousand operations per second	20	80	60	200	380	500	1000
Word length of processor, bytes	1	1	4	4	8	8	8
Capacity of main memory, K bytes	64-256	256-512	128-512	256-512	256-1024	512-1024	512-8192
Space occupied by main assembly, m^2	100	110	150	120	200	200-250	270-300

Table 3. Technical Specifications of Minicomputers

Specification	Computer Model						
	M-400	M-6000	M-7000	SM-1	SM-2	SM-3	SM-4
Mean speed, thousand operations per second	128	67	133	133	154	135	213
Word length (bits)	16	16	18	18	16	16	16
Capacity of main memory, K words	8-128	16-64	16-128	16-32	32-128	8-28	32-124

Space occupied by main assembly, m^2	15-40	20-40	20-40	10-40	10-40	10-40	10-40

Table 4. Technical Specifications of Microcomputers

Specification	Microcomputer Model			
	Elektronika S5-01	Elektronika S5-11	Elektronika 60	Elektronika NTs-03
Speed, thousand operations per second	10	10	250	160
Word length (bits)	16	16	16	16
Capacity of main memory, K words	28	128	4-28	64

[73-8831]

8831
CSO: 1863

UDC 681.142.34

HYBRID COMPUTING SYSTEMS

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY in Russian 1979 pp 2, 163-164

[Annotation and table of contents of book, Izdatel'stvo "Naukova dumka", 172 pages]

[Excerpts] Annotation

This collection contains articles on the theory of hybrid computation and the principles of developing assemblies for hybrid computers and computing complexes. The articles consider the techniques of hybrid modeling of problems of mathematical physics, the organization of work by parallel computing systems, development of the computing equipment of hybrid computing systems, and questions of the construction of problem-oriented systems based on integrated arithmetic units and stochastic computers. Some of the articles deal with peripheral devices and questions of analyzing the precision of hybrid systems and diagnosing them.

The book is intended for scientists and engineering-technical personnel.

Table of Contents

[81-11176]

11,176
CSO: 1863

UDC 681.325.5

USING HARDWARE FOR FUNCTIONAL OPERATIONS IN SPECIALIZED COMPUTERS

Kiev GIBRIDNYYE VYCHISLITEL'NYYE MASHINY in Russian 1979 pp 89-92

[Article by Yu. A. Plyushch, Institute of Electrodynamics of the Ukrainian SSR Academy of Sciences, Kiev: "Hardware Implementation of a Functional Conversion in Specialized Computing Devices"]

[Excerpts] One of the ways to increase the information productivity of computer equipment is to substitute hardware problem-solving for software methods [1, 2]. The determination of the scope of distribution into the software and hardware parts in specialized and functionally oriented computing systems and devices depends on many factors, in particular on the volume of equipment and the functional capabilities of the basic elements being used.

This article reviews certain questions of constructing matrix combination decision blocks which perform functional conversion in one cycle, with due regard for specific types of basic elements. As an example we have selected a series K155 microcircuit which is based on the functions "and/not" and "and/or/not." The typical features of using this basis are considered with the examples of constructing a matrix multiplier and, as a particular case, a device for squaring. However, everything presented below can be quite simply transferred to the construction of other matrix functional convertors which realize the interrelationship between digital input and output quantities by performing the operations of algebraic and (or) logical addition and multiplication of single-order quantities.

The structure of the matrix device for multiplying two numbers consists of a matrix of combination single-order summators interconnected by interdigital carry couplings and a matrix of combination single-order multipliers connected with the inputs of the single-order summators. In the binary system of notation it is customary to use logical multiplication cells ("and" circuits), in which the truth table coincides with the value of the algebraic multiplication table. Figure 1 [not reproduced] is a diagram of a matrix four-byte multiplier in which "and" circuits are used as single-order multipliers. As can be seen, multiplication requires 16 "and" circuits and 12 complete single-order summators. The multiplication circuit (see Figure 3 [not reproduced]) can be built quite simply on the basis of series K155 microcircuits. The registers of the multiplicand and multiplier can be made from type K155TK1, K155TK2, K1555TM5, and K155TM7 microcircuits. The matrix of combination multi-order summators and block A can be made using K155IM1, K155IM2, and K155IM3 microcircuits. The switching blocks B are quite simply made with K155KP2 microcircuits.

[81-11176]

11176
CSO: 1863

UDC 681.32

APPARATUS SUPPORT OF MASS CALCULATIONS

Novosibirsk PROBLEMY NELINEYNOGO PROGRAMMIROVANIYA. OPTIMIZATSIYA in Russian No 22 (39), 1978 pp 115-126

[Article by Ya. I. Fet]

[Excerpts] General Structure of the System

On the whole the system must represent a heterogeneous computer complex consisting of a central machine and a set of functional modules. The central machine (a medium-capacity universal computer) performs the role of a dispatcher, accomplishes coupling with peripherals, stores and services the system programs and produces simple (scalar) processing. Functional modules receive drom the central machine tasks in the accomplishment of mass operations and realize them autonomously.

Effective functioning of the system requires, of course, good organization of the distribution of work array-arguments and the exchange of intermediate results.

The architecture under consideration is not new. We distinguish here, however, the following distinctive features which seem important to us:

--the large-unit character of special processor functions;
--the high level of apparatus support (complex basic operations of special processors);
--the possibility of providing sufficient "universality" when there is a small number of special processors.

Using the descriptive definition from [32], we can call a traditional universal machine with its set of scalar operations a system of "low qualifications." The above-considered computer system with a set of functional modules oriented toward the apparatus realization of mass operators is a system of "medium qualifications." The question can be posed further regarding systems of "high qualifications," in which the apparatus support will be distributed over larger developments, separate important tasks and frequently used algorithms. Among special processors of that designation are, for example, sorting networks [33], BPF [fast Fourier transform] processors [14] and matrix processors for the solution of systems of differential equations [34]. If necessary, such processors can be included in a universal system, raising its "qualifications."

If one takes as an estimating criterion the number of basic operations (cycles) necessary for the realization of some standard set of tasks, the system of the described architecture can evidently assure high effectiveness.

BIBLIOGRAPHY

14. Yevteyev, Yu. I., et al. Apparaturnaya realizatsiya diskretnogo preobrazovaniya Fur'ye (Apparatus Realization of a Discrete Fourier Transform). Moscow, "Energiya," 1978.

33. Batcher K. E. Sorting Networks and Their Applications. In: AFIPS Conf. Proc., 1968, SJCC, Vol 32, pp 307-314.

34. Il'in, V. P., and Fet, Ya. I. Parallel computer for solution of difference equations of field-theory problems. Decision of the State Committee on issuance of an Author's Certificate on Application No 2306000 dated 30 Mar 77.

[69-2174]

2174
CSO: 1863

SOFTWARE

PROGRAM PRODUCTION ON BESM-6 AND YeS COMPUTERS: THE R-TECHNOLOGY

Moscow TEKHNOLOGICHESKIY KOMPLEKS PROIZVODSTVA PROGRAMM NA MASHINAKH YES EVM I BESM-6 in Russian 1980 signed to press 16 Jan 80 pp 2-11, 17-29, 261-263

[Annotation, table of contents, foreword, introduction and section 1.4 of Chapter 1 from book by Igor' Vyacheslavovich Vel'bitskiy, Vasiliy Nikolayevich Khodakovskiy and Leonid Ivanovich Sholmov, Statistika, 30,000 copies, 263 pages]

[Excerpts]. A new Soviet automated technology of programming is described, one which permits considerably increasing the labor productivity of programmers. In contrast with an individual technology of programmer labor it provides the possibility of organizing large teams of programmers, which is especially important in the creation of large program projects.

A description is given of the main concepts of the new programming technology and the realization of technological complexes for program production on BESM-6 and YeS computers.

The book is intended for programmers and specialists working with computers, scientific workers and associates of computer centers studying questions of system and applied programming. It will also be useful to students studying questions of programming technology.

Contents

Page

Foreword

Programming technology is a new, rapidly growing directing in programming. Interest in engineering questions about programs and details of the process of program production and in the formation of an industrial culture of programming is caused by increase in the labor-intensiveness and cost of program system production. At the present time the cost of program system production is several times that of computer hardware. The tendency is such that in 1990 the correlation between the cost of computer software and hardware will be the same as between the cost of a commodity and its packaging.

The reason for the situation that has developed is that programming formed as a creative activity difficult to discipline. Programming officially was called an art. Each programmer wrote programs without adhering to any sort of clear principles, standards or limitations. It became almost impossible to analyze what a program written by another man does, and practically impossible to completely debug such a program. The situation developed in which it was easier to re-write a program than to analyze what someone else had done or change it slightly. As a result about 80-90 percent of the functions existing at the present time duplicate one another functionally, and only 1 - 3 percent of the total enormous volume of accumulated software is actively used.

In the book offered to the reader a new and original Soviet technology (the R-technology) of programming is presented. That technology was formulated in the Institute of Cybernetics of the Ukrainian SSR Academy of Sciences on the basis of basic research on the theory of automata and machines with a high-level structural interpretation of languages. The basis of the proposed technology is a new view of the semantics of data processing as a process for which the structure of data with processing algorithms suspended on it is the starting one. A characteristic feature of the new technology in comparison with known foreign technologies (modular and structural programming, the HIPO-technology, top-down programming, etc) is the possibility of organizing the team labor of programmers and the possibility of creating effective problem-oriented complexes for the production of program systems. Such complexes have been produced for the principal Soviet computers, the YeS computers and the BESM-6 and have been checked in the creation of a number of program systems. The YeS RTK and BESM-6 RTK technological complexes for program production also are described in the present book.

The presentation of material in the book is made systematically and the material requires no preliminary special training. A description of the general principles of R-technology and RTK complexes is presented in the beginning, and then guidance for programmers for work on RTK complexes. Thus, the book will undoubtedly be useful to a large army of programmers and specialists studying questions of system and applied programming, and also for scientific workers engaged in forming an industrial culture of programming.

A. A. Stogniy, corresponding member of the Ukrainian SSR Academy of Sciences

Introduction

A characteristic feature of the development of programming on the basis of algorithmic languages and existing operating systems is a sharp increase in the cost of producing program systems. In 1971 the cost of program systems was the same as the cost of computer hardware. At the present time the software cost is several times that of the hardware, and program production is a serious problem in many developed countries.

Starting in 1970, to lower the cost of production of program systems and increase the labor productivity of programmers, in our country and abroad there has been wide use of methods which regulate a high professional level of program writing regardless of the language, operating system, computer or problem to be solved. Such methods have received the general name of programming technology.

In 1976 a new programming technology was developed on the basis of generalization of a large amount of experience in the production of computer-aided systems to produce real-time programs. It has received the name of the R-technology of program production, or two-dimensional programming technology. In that technology, on the basis of basic research on the theory of automata and machines with high-level structural interpretation of languages, some principles of the formal breakdown of the programming process into a number of simple subprocesses have been clearly distinguished, for the performance of which, earlier than in ordinary programming practice, means of automating the programmer's work by means of the computer itself are drawn in. The R-technology assumes a unified mathematical apparatus for the entire technological cycle of program production: planning, coding, debugging, documentation and operation. The new technology permits carrying out the complete automation of the programmer's work far more profoundly than has been done so far in the better Soviet and foreign technologies. The application of the R-technology permits organizing technological lines for the mass production of programs, planning, controlling large teams of programmers in the development of complex program systems, attracting to programming a broad circle of users who are not professional programmers and reducing by one half to two thirds the time required for the production of program systems.

Very promising is the use of the R-technology to process symbolic or textual information and to collect program product from available program modules, different in level and programming language. That class includes tasks in the construction of translators, microgenerators, packages of applied programs, ASU, automated design systems, information retrieval systems, etc.

To automate the work of programmers on R-technology, on the main Soviet computers, the YeS computers and the BESM-6, RTK technological complexes have been created. The RTK complexes are, in essence, the first automated technological program production lines. The R-technologies and RTK complexes have passed tests in a number of organizations.

The realization of technological complexes for program production on the YeS and BESM-6 computers is described in this book. The basic conceptions of the new programming technology are discussed first. Then a detailed description is given of the YeS RTK and BESM-6 RTK technological complexes for program production. The description of the complexes is sufficiently detailed and rigorous and can serve as a methodical and at the same time working guide for the programmer.

In its structure the book is divided into 12 chapters. In the first four chapters a description is given of the general conceptions of the new technology without reference to its specific implementation. The materials of those chapters reveal from general positions the essence of the new technology, its connection with known approaches to the automation of programming and distinctive features of the automation of various stages in the work of programmers in the new technology. In the remaining chapters a description is given of specific developments of RTK technological complexes for YeS computers (chapters 5-9) and the BESM-6 (chapters 10-11). Described in chapters 5 and 10 are the user's languages of interaction with RTK technological complexes, and in Chapters 6 and 11 the main questions regarding their operation. In chapters 7, 8 and 9 the main service packages for the effective work of users according to the new technology of YeS computers are described. There are three such packages: the STELZ metasystem, the interaction text editor and the structure assembler.

The STELZ metasystem is a new type of program system which permits creating evolutionarily developing programming languages. The construction of such systems became possible in principle thanks to distinctive features of information processing in the R-technology. By means of the STELZ metasystem a user can simply assign any recording element standardized by him (STandartizovannyy ELement Zapisi= STELZ), introduce it into the archive of the metasystem and apply it effectively further in his work. Practically no limitations are imposed on the syntax of STELZ assignment as well as on the method of its use in any program of the user. The purpose of the remaining packages is evident from their name.

The material of the book is designed for various readers--system and applied programmers, scientific workers, designers of new computers, students, etc. To assure accessibility of material of the book to a reader who is not a professional programmer, the first four chapters, in which the concepts of the new technology are described, use only Russian notation. In the realization of those conceptions on existing operating systems of YeS and BESM-6 computers the English notation customary for those systems was adopted. The conceptions of the new technology are described, not as dogma, but as a certain guide to action which the user can adapt creatively to his own conditions in the creation of specific technological program production lines. Therefore the description of the first four chapters contains a large number of simple examples and methodical recommendations and is consciously not formalized. It seemed to us that such a presentation of the material of those chapters will permit effectively using the new technology to solve not only tactical but also strategic questions regarding the creation of program systems.

In writing section 4.4 of the book the authors used materials presented by S. T. Rodionov. Chapter 9 on the STELZ metasystem was written with the use of working materials of the author of that metasystem, A. L. Kovalev. The authors wish to express their appreciation to those associates. The authors consider it their duty to express their appreciation and gratitude to V. P. Breyev, L. N. Korolev and L. B. Efros for valuable comments during the preparation of the book.

Chapter 1. General Information on the R-Technology of Program Production

The present chapter contains a description of the general conceptions of the new programming technology. That technology has been called two-dimensional programming technology or the R-technology of program production. The letter R in the name of the technology has no special significance and is merely a continuation of previously introduced working notation. The programming is called two-dimensional because representation of the program product in the form of a certain special graph forms its basis.

1.1. History of Its Development

In the second half of the 1960's, in the Institute of Cybernetics of the Ukrainian SSR Academy of Sciences, on the basis of the work of V. M. Glushkov on finite automata and automata with magazine memory [1,2], investigations were started on the formal description of programming languages. As a result a metalanguage was created for the formal description of the syntax of languages, called the R-metalanguage [3,6]. The grammar of the language in that metalanguage (the R-grammar) was presented in the form of a certain oriented graph, on the area of which were written symbols of the language and operations on any finite number of magazine memories. There were three such operations: recording, reading and empty. In the operation of recording, any symbol can be written in any magazine. The operation of reading was a certain condition of passage along the R-grammar arc. If the arguments of that operation coincided with the symbols on the peaks of the corresponding magazines, the transition along the R-grammar arc was accompanied by reading of the indicated symbols from the magazines. In the contrary case transition along the arc was forbidden and the reading was not done. It is understandable that the syntax of any programming language can be described in such a metalanguage.

In contrast with the metalanguage of Backus-Naur forms which was widespread at that time [7,8], the new metalanguage had considerably better characteristics with respect to the rate of syntactic recognition of texts in programming languages (10-50 instructions/symbol for the R-grammar and 1000-2000 instructions/symbol for the BNF grammar). This is natural, as the R-metalanguage has an essentially automatic nature. Two other features of the R-metalanguage were less evident: compactness of description and simplicity (technological effectiveness) of assignment of the syntax of existing programming languages. For example, the syntax of the ALGOL-60 language in the R-metalanguage occupies 5000 bits of computer storage, and in the metalanguage of Backus-Naur forms, 10,000 bits [9]. The technological effectiveness of the assignment of the syntax of programming languages has been checked and confirmed on a large number of languages of different levels: assemblers, procedure-oriented (ALGOL, FORTRAN, PASCAL) and problem-oriented.

In 1975 the concept of abstract memory was introduced into the R-metalanguage, and simultaneously four new types of it: counter, register, carload (a generalization of the magazine type of astract memory) and tabular [10-12]. The new types of abstract memory have permitted technologically describing in the R-metalanguage not only the syntax but also the semantics of programming languages. This means that for the recording of semantics it is sufficient, on the arcs of a graph giving the syntax of the language in the R-metalanguage, to record the corresponding operations on new types of abstract memory. The semantics of languages are given in a certain sense according to the same technology as the syntax. The syntax and semantics of the ALGOL-60 and PASCAL languages were formally described according to the proposed technology in the course of 1975-1976.

In 1971-1976 work was started on ASPP SINTERM (Avtomatizirovannaya sistema proizvodstvo programm real'nogo vremeni--Automated System for Production of Real-Time Programs) on the basis of the R-metalanguage. That system was intended for the production of specialized computer programs on the large BESM-6 universal computer, called an instrumental computer. In the course of 1971-1976 three generations of that system were constructed for three different specialized computers respectively. The total volume of software for an instrumental computer at the present time is 300,000 BESM-6 instructions. All three generations of the ASPP SINTERM system have been introduced into industry and are operated effectively at this time.

Generalization of experience in the construction of the ASPP SINTERM has also served as the basis for the formation of an R-technology for program production [15-17]. In the process of such formation the concept of the R-metalanguage and syntactically controlled translation has been gradually replaced by more general concepts of the R-machine and the programming language on it. In 1976 the R-technology was successfully transferred to a state commission, and its mass distribution started after that.

1.2. What the R-Technology Will Give

The new technology has been used effectively for the construction of translators from the FORTRAN and PASCAL languages [18-19], several assemblers, a structural expansion of ALGOL [20], the recoding of ALGOL-programs on the BESM-6 into ALGOL- and PL-programs on the YeS computer, an information retrieval system [21], an automated information system for counting empty and occupied individual apartments of Yerevan on the basis of the SUBD (Data Base Management System) "BANK" [23], the controlling part of a package of programs for designing shells of complex structural designs [24], programs for the formation of syntactically oriented listing of PL-program printing, programs for the automatic tracing of PL-programs for interactive debugging [22], systems for the automation of editorial and publishing work [25-26], program support of shared collective-use centers, etc.

Experience in the use of the new technology shows that it substantially simplifies and accelerates the four most labor-intensive stages in traditional programming: planning, debugging, documentation and operation, that is, the accompaniment and introduction of changes into a program system by other than its developers. In the new technology the stage of program encoding does not differ in labor-intensiveness from program encoding in existing high-level languages. The maximum labor productivity of a team of programmers working on the R-technology amounted to 100 instructions per man per day in the time interval from the technical task agreed upon with the purchaser to delivery of the finished and documented program product with a volume of 60,000 instructions. This includes instruction of the purchaser during delivery of the system. The labor productivity of programmers in doing similar work according to the traditional technology earlier did not exceed 5-10 instructions per day for the same workers.

Such high labor productivity for a programming team using the R-technology is not a maximum limit as widespread mastery of the R-technology is only a beginning.

The users who first mastered the R-technology noted its following merits:

--the team character of programmer's labor according to a clear top-down plan: all the work is spread over bands and all programmers work rhythmically according to a simple plan unified for all from the top down;

--independence of the technology from the qualifications of the performer and their mutual transpositions in the course of the work. There is positive experience in the use of the R-technology by non-professional programmers;

--stability of the technology toward errors in planning, ease of introduction of corrections into initially accepted architectural solutions;

--the possibility of effective control from above of the course of performance of the work, the possibility of effective application of network planning of work, the smooth connection of new actuators, the possibility of strong elimination of duplication of work and the organization of production relations in the team;

--the presence of means of automating work on the R-technology, means effective and convenient for work, a volume of documentation small and adaptable for organization (in comparison with that of YeS computer software).

One to two weeks to 1-2 months are required for the organization of R-technology.

1.4. General Description of the R-Technology

In analyzing Figure 1.1 still another important conclusion can be drawn. All existing methods of increasing the labor productivity of programmers are based on modification of the superstructure, whereas the basis, the principle of data processing in the computer itself, remains unchanged. The meaning of work on the creation of the R-technology consists in modification of the basis in such a manner as to simplify the process of programming, to draw it nearer to a production process more natural for man. For that purpose a certain provisional computer, called an R-machine or an RBM for short, realized for the programmer, is offered to him. Then a corresponding superstructure--a language (R-language) and a programming technology (R-technology)--is constructed for that machine.

Before describing distinctive features of data processing in the RBM machine for the programmer, we will dwell on shortcomings of existing computers. The main shortcoming in the organization of data processing on existing computers consists in the fact (Figure 1.5) that there is no direct connection between the data and the program which processes them (the connection between them is implicit, through man). For a program on a computer, if it was not constructed in a special manner (and, of course, artificially complicated in the process of planning), it does not matter which data are to be processed. In the construction of such programs all connections and correspondences between the program and data on the input and output must be kept in mind by the programmer. For the first computers, oriented toward computing tasks, the volume of such connections was not great, and so under the conditions of a primitive elementary base such a principle of data processing was very progressive. At present the volume and complexity of the data to be processed have increased so much that the volume of connections which must be kept in mind during the compilation of a contemporary program has surpassed the threshold natural for man, and this has led to a sharp complication of the process of programming for computers with a traditional structure.

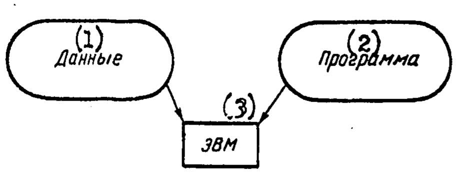


Figure 1.5. Principle of computer data processing.
1 -- Data 2 -- Program 3 -- Computer

In the RBM machine for the programmer it is proposed to include in the program itself an explicitly given connection with the data, for which the program flowchart is combined with the logical data structure and the functional part of the program is distinguished separately (Figure 1.6). In the R-technology, to record such programs it is proposed to use the most graphic of known languages--the language of loaded oriented graphs. In that language the logical data structure combined with a program flowchart is given by recording the corresponding symbols and predicates on the arcs of the graph, and the functions of processing the given structures are given on the same graphs by recording a series of only linear statements: assignment statements, statements of appeal to procedure-functions or statements of transfer between RBM memories. For such a program, in contrast with a computer program, it does not matter which data are to be processed. The processing of data not corresponding to the logical data structure is blocked--leads to the halting of the RBM, in contrast with the computer, and is similar to how

APPROVED FOR RELEASE: 2007/02/08: CIA-RDP82-00850R000300080023-5

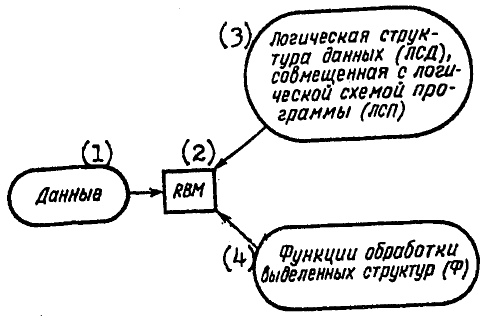


Figure 1.6. Principle of information processing on the RBM machine for the programmer.

1 -- Data
2 -- RBM
3 -- Logical data structure combined with the program flowchart
4 -- Processing functions of separated structures

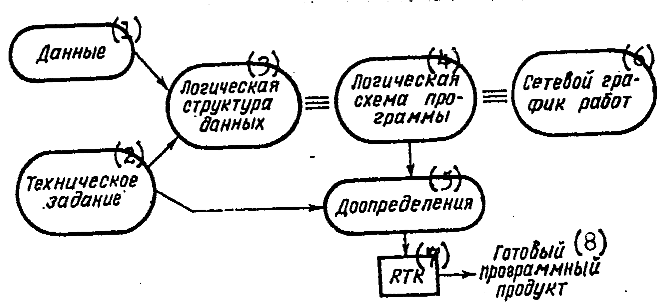


Figure 1.7. Programming technology for the RBM.

1 -- Data
2 -- Technical task
3 -- Logical data structure
4 -- Program flowchart
5 -- Predetermination
6 -- Network operation schedule
7 -- R-technological complex
8 -- Finished program product

man himself processes data. It is obvious that for the computer, by virtue of its universality, a program also can be constructed in which the processing of data not corresponding to the program flowchart is blocked and which also leads to halting of the computer in the case of the processing of incorrect information. In practice only such computer programs are constructed, as the others do not make sense. For the computer this is achieved through substantial complication of the program and the process of its construction, but for the RBM this flows from the very working principle of the R-machine and requires no efforts from the programmer.

The programming technology was changed for the RBM (Figure 1.7). It is clearly divided into two main stages. In the first stage the structure of the information is formally determined without connection with any sort of algorithm for its processing. In the second stage that structure is regarded as the logical diagram of the corresponding algorithm for its processing and as a certain diagram (network graph) of the work of the team of programmers on predetermination of the starting structure. The predetermination is accomplished only by linear operators. Used as such operators are either generally known (assignment statements) or statements obtained and standardized in the early stages of the development of work on the R-technology, or the user can introduce any of his own linear statements, recording them in advance in one of the programming languages of existing computers. At times the initial logical data structure obtained in the first stage is also modified in the process of predetermination. As a result of predetermination an R-program is obtained which by means of the RTK technological complex, is converted (by automatic generation) into a finished program product for work on existing computers.

1.5. A Very Simple Example

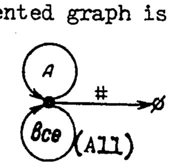
Let it be required to calculate the number of letter A's in a text bounded on the right by the symbol #. The recording of the corresponding algorithm in the PASCAL language--the most convenient of the programming languages in general use for the solution of such problems on computers, has the following form:

```
program  COUNTLETTERA (input, output);
var  SYMB: char; C : integer
begin  C: = 0;
repeat  read(SYMB);
if  SYMB = 'A' then C: = C + 1
until  SYMB = '#';
writein(C)
end
```

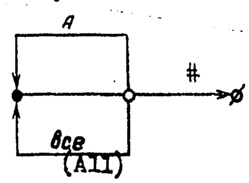
The first line of the program gives its name and the names of the standard input and output files. Such assignment of the input and output files in accordance with the syntax of the PASCAL language permits omitting their names and recordings of the operators read and writein in the body of the program itself.

In composing the program it was assumed that the initial text in which the number of letter A's is counted is found on a standard input file. The symbols of the initial text are subsequently (in the repeat cycle) read by an operator read (SYMB) and recorded in the SYMB working cell. The result of the work program--the value of the counter (C)--the number of letter A's is formed by the operator writein (C) on the standard output file and printed at the end of the program. The remaining recordings in the program are obvious. Also obvious from the presented recording of the program are the order of actions and the process of thinking (the working technology) of the programmer during construction of the required algorithm. They are traditional and generally known.

The process of designing the corresponding R-program starts with formal definition of the structure of the starting text in which the number of letter A's must be counted. The definition is made independently of any sort of algorithm for making the count. The following loaded oriented graph is used for the definition:



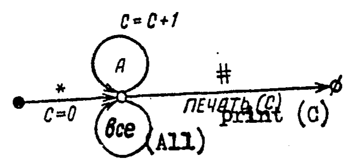
or more technologically with use of only horizontal and vertical lines:



where the arc without the arrow corresponds to identification of the peaks formed by it, ALL is the synterm or symbol of the R-machine's internal alphabet, designating in the given case that on the arc ALL can be any of the symbols used to record the initial text.

The presented graphs are R-machine programs or R-programs. Those programs still give only the logical structure of the initial text (the logical data structure). In the given case they give that structure as a sequence of any symbols, separating in it the two symbols A and #. The symbol A, in accordance with the given structure, can be encountered any number of times at any place in the series (in the initial text); the symbol # always is the last in the series. For correct reading of the R-program one should adhere to a fixed order in examining arcs: top-down and left to right around each peak. Arcs not containing recordings (sometimes such arcs are noted by the metasymbol *) are examined next. The first stage of R-program planning concludes with the construction of those graphs.

In the second and concluding stage the obtained R-program is determined more precisely for the algorithm for counting letter A's. For that purpose the arc of the initial installations (INITINST) is written in the above-presented graph and the two other graphs are determined with the following obvious linear operators:



Such a graph is a conclusive R-program. The recording of that program for input into the machine has the following form:

```
          R-program  COUNTLETTERA
          counter  C
INITINST  counter  C = 0      COUNT
   COUNT  A        C = C + 1  COUNT
          #        PRINT (C)  output
          all      *
          end
```

The first two lines of that program are auxiliary and give the name of the program and the RBM used by the counter memory (description of the RBM configuration). The recording of the R-program and each RBM instruction are put in order in four columns. In the first and last the marks INITINST, COUNT and output, which give a series of steps in implementation of the algorithm, are clearly noted. In the second column information is recorded on the structure of the initial text. That information is a certain condition for fulfilment of linear operators of the algorithm, which are recorded in the third column of the R-program. If the corresponding condition in the second column is true (that is, the current symbol on the RBM machine corresponds to the symbol in the second column) or is designated by the metasymbol * (which always corresponds to the true condition regardless of the current symbol on the RBM input), then the operators recorded on the line in the third column of the R-program are implemented, and then a transition is made according to the mark clearly indicated in the fourth column.

The work of the R-program starts with analysis of the first RBM instruction in the instruction complex called INSTINST. In the given case the complex contains one instruction which always can be accomplished, as it contains in the second column of the field of conditions the metasymbol *. Therefore the zero is assigned to the counter C and control is transmitted to the following complex of RBM instructions with the name COUNT. That complex consists of three RBM instructions. The first is analyzed first. It can be accomplished if the first (generally the current) symbol of the initial task is the letter A. In that case the unit (C = C + 1) is added to the counter, control is transmitted to the same instruction complex--COUNT, and the following is stated by the current symbol of the initial text.

If the first instruction of the complex must be accomplished, the following RBM instruction is analyzed, etc. If none of the instructions of the complex can be accomplished the R-machine halts, which corresponds to error in the initial text or in the R-program itself. In that case the error is corrected either manually by the programmer or automatically by means of the corresponding algorithm for error neutralization. Those algorithms are described in chapters 2 and 5.

In the absence of errors the work of the R-program concludes in the transition to a fixed mark output in the second instruction of the complex COUNT. A condition for the accomplishment of that instruction is the symbol # as a current one in the initial text. In that case with respect to the operator PRINT (C) the content of the counter (C) is printed and the R-machine is halted on the work output.

If the third instruction of the complex COUNT is analyzed, the current symbol of the initial text is compared with any symbol of the system all. In the comparison no effects are produced (since the metasymbol * is recorded in the third column of the instruction), control is transferred to the same complex of instructions COUNT, the following symbol of the initial text is the current one, etc.

1.6. A Very Well Known Example

It is known that for a better understanding of the new it is advisable to select a counterexample which would be nontypical or disadvantageous for it. The class of computational problems is such a counterexample for the R-technology. The R-technology is now used mainly for tasks in symbolic data processing. That technology has not been used for computational problems and it is considered (see subsection 1.4) that an existing computer with traditional programming is good for them. The example examined below in a certain sense puts the R-technology in conditions disadvantageous for it.

Let it be required to calculate the roots of the equation

$$ax^2+bx+c=0.$$

A known solution of that equation is the formula

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

where the discriminant $d = b^2 - 4ac$.

Since the coefficients of that equation a, b and c can assume any values, the algorithm for calculation of the presented quadratic equation in the general case appears as follows:

Step 1. If $a = 0$, then step 5 or else $x_1 = -b/2a$, $d = x_1^2 - c/a$,

Step 2, if $d > 0$, then (1); $d = \sqrt{d}$, $x_2 = x_1 - d$, $x_1 = x_1 + d$ output

Step 3, if $d < 0$, then (2); $x_2 = \sqrt{-d}$ output

Step 4, (3) output

Step 5, if $b \neq 0$, then (4), $x_1 = -c/b$ output

Step 6, if $c \neq 0$, then (5) output

Step 7, (6) output,

where the different branches of the algorithm are designated by marks which have the following meanings:

(1) two different real roots;
(2) two complex roots;
(3) two equal roots;
(4) one root;
(5) no roots;
(6) infinitely many roots.

(Translator's Note: These numerical designations also apply to Figures 1.8, 1.9, 1.10 and 1.11, and to the algorithms. Also add:

(7) input; (8) output

A recording of the presented algorithm in the language of generally accepted block diagrams is given on Figure 1.8. The recording of the algorithm in the ALGOL-60 language has the following form:

```
    АЛГОРИТМ: begin real a, b, c, d, x1, x2:
ALGORITHM:        input(a, b, c);
      НАЧАЛО:     if a≠0 then
   BEGIN:            begin x1:=-b/(2×a);
                        d:=x1↑2 - c/a;
                        if d > 0 then
      (1) ДРВК:            begin d:=sqrt(d);
                              x2:=x1 - d;
                              x1:=x1 + d;
                              goto ВЫХ(8)
                           end
                        else
                           if d < 0 then
      (2) ДКК:                begin x2:=sqrt(-d);
                                 goto ВЫХ(8)
                              end
                           else
```

```
(3) ДРК:                 goto ВЫХ (8)
                   end
                else
                   if b≠0 then
(4) КО:                  begin x1:=—c/b; goto ВЫХ end
                   else                           (8)
                         if c≠0 then
(5) КН:                     goto ВЫХ (8)
                         else
(6) КБМ:                    goto ВЫХ (8)
(8) ВЫХ: end
```

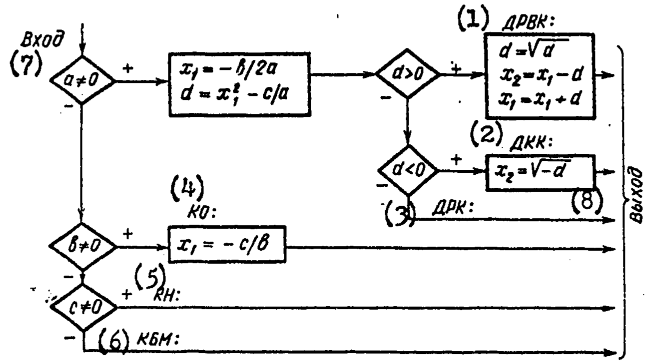


Figure 1.8. Block diagram for calculating roots of a quadratic algebraic equation.

Since we want to make a comprehensive comparison of the R-technology with known technologies, we will present a recording of the algorithm in accordance with the principles of structural programming. The corresponding block diagram is presented on Figure 1.9. A recording of the same block diagram by means of stylized structural programming diagrams is presented on Figure 1.10. The recording of the algorithm in the corresponding ALGOL structural modification has the following form:

```
 АЛГОРИТМ: begin real a, b, c, d, x1, x2;
ALGORITHM:     input(a, b, c);
  НАЧАЛО:      if a≠0 then
  BEGIN:         begin x1:=—b/(2×a);
                   d:=x1↑2 — c/a
                   if d > 0 then
(1) ДРВК:            begin d:=sqrt(d);
                       x2:=x1 — d;
                       x1:=x1 + d
                     end
                   else
                     if d < 0 then
(2) ДКК:               x2:=sqrt(—d)
                     else
(3) ДРК:             fi
                   fi
                 end
               else
                 if b≠0 then
(4) КО:              x2:=—c/b
                 else
                     if c≠0 then
(5) КН:              else
(6) КБМ:             fi
                 fi
               fi
           end
```

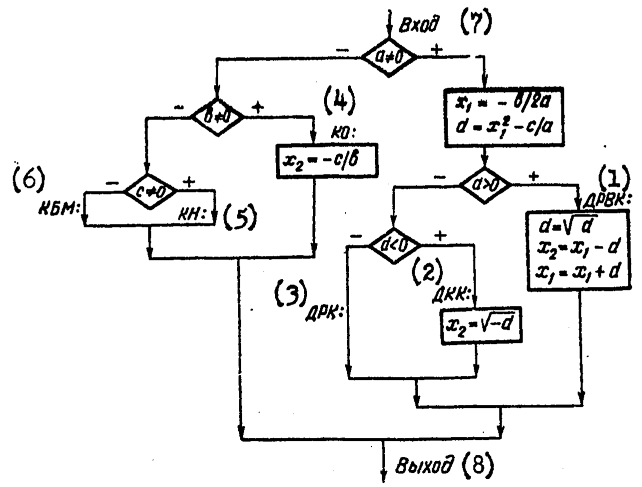


Figure 1.9. Structural block diagram for calculating roots of a quadratic algebraic equation.

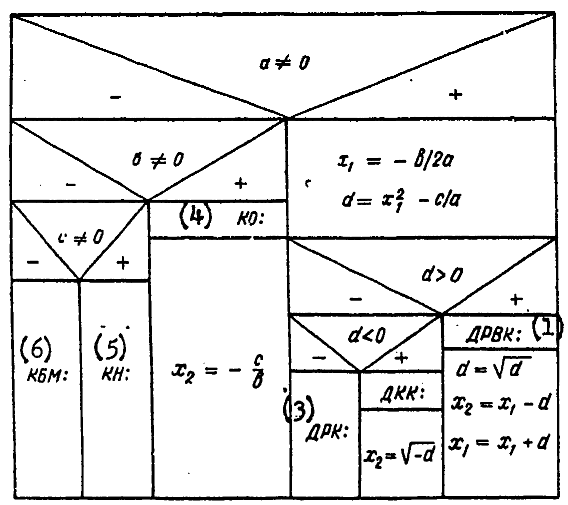


Figure 1.10. Recording of an algorithm for calculating roots of a quadratic algebraid equation by means of stylized structural programming diagrams.

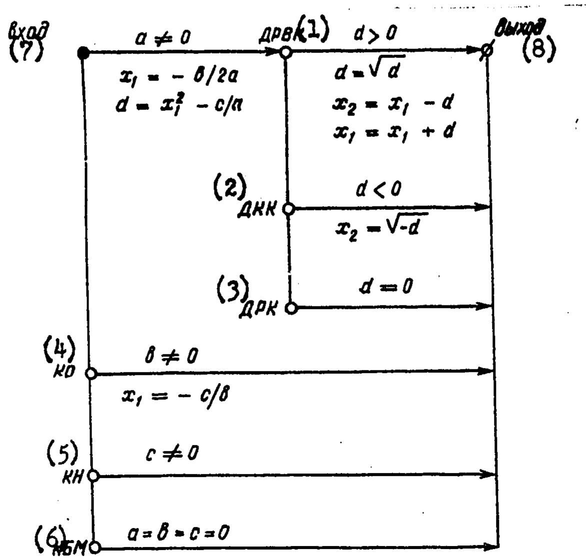


Figure 1.11. R-program for calculating roots of a quadratic algebraic equation.

The R-program of the same algorithm is presented on Figure 1.11. The recording of that algorithm in linear form for input into a computer has the following form:

```
                          R-program ALGORITHM
BEGIN                     counter a,b,c,d,x1,x2      ANALYSIS
НАЧАЛО      •    (7)  ВВОД(a, b, c)                   АНАЛИЗ
АНАЛИЗ      a≠0:      x1=—b/(2×a),
ANALYSIS              d=x1↑2—c/a                  (1) ДРВК
 (4) КО     b≠0:      x1=—c/b                     (8) выход
 (5) КН     c≠0:      •                               выход
(6) КБМ     •         •                               выход
(1) ДРВК    d > 0:    d=sqrt(d),
                      x2=x1—d,
                      x1=x1+d                         выход
(2) ДКК     d < 0:    x2=sqrt(—d)                     выход
(3) ДРК     d=0:      •                               выход
                      конец  end
```

1.7. General Conclusions

Analysis of the above-presented very simple examples permits drawing the following general conclusions.

Firstly, the R-program is a very clear method of algorithm recording. This applies to both the graphic and the linear forms of R-program recording.

In a linear R-program all the logical data structure predicates are separated into a single column (the second) and all the data structure processing operators into another (the third). All those predicates and operators are also present in recordings of the same algorithms in PASCAL or ALGOL, but there they are scattered chaotically over the program task, and in the R-program are distinctly characterized by recording place, by columns. This means that the traditional form of the recording of programs in programming languages is less technological (graphic, convenient for reading and understandable) than the form of R-program recording. Therefore R-programs are more effective in debugging, study and operation than programs recorded in traditional programming languages.

The positional method of R-program recording has still another advantage over the traditional. In R-programs there are no key-words of the type of if, for, while and others for the separation of predicates and the program logical scheme. Those words are unquestionably useful in the teaching of the programming language both for the recording of very simple small programs and in the first days of programming instruction. During industrial problem preparation those limitations clutter the texts of programs and make them non-technological in the work and unclear.

In the R-programs themselves there also is an element that was not present in traditional programs--the marks that clearly give the order of R-program implementation. Therefore it can appear that the R-programs also are cluttered. However, that is not so, as the R-program marks take an active part in their documentation designating separate algorithm sections, their implementers, etc. In addition, the R-program marks are brought out from the main field of the program to the edges and and are recorded strictly in the first and last columns. Therefore in reading the R-programs, if it is unnecessary, the marks can be ignored. This property of the new recording form is especially evident on the example of the ALGORITHM R-program (see section 1.6).

Now let us analyze the graphic form of the R-program recording. That recording form reminds one very much of program block diagrams. However, in contrast with block diagrams, the R-program can be debugged on an RBM in the ordinary way in all stages of its manufacture. After being coded in any programming language the block diagrams become inadequate for the corresponding programs. The divergence between them becomes larger in the process of debugging. The attempt to use block diagrams as documentation for program product leads to an extremely complex programming technology.

In contrast with block diagrams the graphic recording of R-programs is much more technological. It does not contain complex profiles: rectangles, rhombs, etc. A graphic R-program and an R-program in a machine always match one another precisely with respect to very simple recodings, which always can be done by unskilled personnel or automatically on a computer, with issuance to an alphanumeric printer or graph plotter.

Secondly, to describe an algorithm in an R-program only linear operators of the type of assignment statements, appeals to procedures, standard operations of recording, retrieval, etc, on various types of R-machine memory are used. Traditional cycle operators (of the type of for, while, repeat, etc), conditional (of the type of if, case, etc) and unconditional (of the type of goto, exit, etc) of

Table 1

Parameters for Comparison	Name of Program: COUNT LETTER A	Name of Program: ALGORITHM
1. Number of symbols in program in PASCAL language	159	-
2. Number of symbols in program in ALGOL-60 language	-	361 -
3. Number of symbols in program in structurized ALGOL-60 language	-	- 303
4. Number of symbols in program in R-language	96	193
5. Ratio of number of symbols in program in a known programming language to number of symbols in program in R-language	1.58	1.87 1.57
6. Integrated estimate of the advantage of writing cited programs in R-language	1.67	

transitions are absent in R-programs. There are no such operators in the third column of the R-program, where all operators of the corresponding algorithm are recorded. Therefore the R-technology by analogy and in contrast with structural programming is called the technology of programming on graphic data structures (it is given mainly by the first, second and fourth columns of the R-program) without the goto operators, without conditional operators and without loop operators. This feature makes R-programs transparent, unentangled and technological for general application. By this the R-technology is distinguished advantageously from known technologies.

Thirdly, the R-program is the most compact method of algorithm recording. Usually the R-program is shorter by one to two thirds (contains fewer symbols) than an analogous program in a high-level language. In Table 1 it is shown that the R-program has 39.6 percent fewer symbols (159/96) than the corresponding program in the PASCAL language, 46.5 percent fewer (361/193) than that in the corresponding modification of the ALGOL language and 36.3 percent fewer (303/193) than a program written in structural ALGOL. On the average for the two cited programs recording in R-language is shorted by a factor of 1.67 than a recording of the same programs in the widespread programming languages.

[69-2174]

2174
CSO: 1863

NEW BOOK PRESENTS FUNDAMENTALS OF PACKAGES OF APPLIED PROGRAMS

Moscow PRIKLADNYYE PROGRAMMY in Russian 1980 pp 2, 179-180

[Annotation and table of contents of book "Applied Prog ıms", by L.I. Gurova and S.S. Sakharov, Izdatel'stvo "Statistika", 1980, 20,000 copies, 180 pages]

[Excerpts] Annotation

This book considers one of the basic lines of development in software for current computers — packages of applied programs. It presents the basic concepts and the structure and general principles of their development. The book describes methods of designing particular parts of a package with application to third-generation computers equipped with YeS [Unified System] disk operations systems.

The book is intended for students at higher educational institutions who are studying the questions of the application and development of special software. It is a textbook that may also be useful to engineering-technical personnel who are interested in these matters.

Table of Contents

[79-11176]

11,176
CSO: 1863

APPLICATIONS

UDC 519.6

THE AUTOMATED SYSTEM FOR INFORMATION SUPPORT OF DEVELOPMENTAL WORK

Moscow AVTOMATIZIROVANNAYA SISTEMA INFORMATSIONNOGO OBESPECHNIYA RAZRABOTOK in Russian 1980 signed to press 20 Feb 80 pp 2-8, 18-19, 126-132, 175-179

[Excerpts from book by V.R. Khisamutdinov, V.S. Avramenko and V.I. Legon'kov, Izdatel'stvo "Nauka", main editorial office for physics and mathematics literature, 12,000 copies, 208 pages]

[Excerpt] Experience with the design, introduction and operation of the automated system for developmental work information support (ASIOR) is treated in this book. The set of system programs for the BESM-6 computer makes it possible to provide a wide circle of information services to subscribers. The programs permit the selective distribution of information and retrospective search in documental and factographic subfiles in batch and interactive modes using displays and teletypes.

Table of Contents

Foreword

Questions of improving the labor quality and efficiency of scientific workers, and the fastest possible practical implementation of the results of scientific work have become and important factor in increasing the nation's economic efficiency at the present time. In this regard, questions of information support for scientific research and developmental work have taken on special significance. This is a problem involving multiple plans. An important part in the resolution of this problem has been assigned to the program for designing the state system for scientific and technical information, being implemented under the supervision of the State Committee of the USSR Council of Ministers for Science and Engineering. Considerable attention is being devoted within the framework of this program to the development and introduction of automated scientific and technical information systems (ASNTI) and the network of automated centers for scientific and technical information (SATsNTI).

As a rule, large collectives of specialists in different fields of science and engineering participate in the development of models of new equipment and in the solution of complex scientific problems. Information support for comprehensive programs is an especially complicated problem, since it is necessary to take in a braod area of information sources at the input to the information system. It is clear that a well-tuned system for coordinating the work on processing the world-wide flow of published and unpublished information is needed for the solution of this problem. The constantly increasing volumes and cost of information processing have brought about the necessity of organizing a division of labor and interfacing information systems for the purpose of eliminating duplication of information efforts. Under these conditions, great hope is invested in the possibilities of interfacing automated information systems, both through communications channels and by means of information exchange via magnetic tapes.

With the existing ASNTI's in service, the success of information services is governed by the possibilities of adapting the system to the users, i.e., by how timely a system responds to a change in the information demands by the specialists being served.

Considering the fact that information support for real users, given the present state of the art of science and engineering, can be provided only where a network of interconnected systems is present, an important task is the design of an information system which is universal in terms of the interface capabilities and adaptable with respect to specific users.

The ASIOR automated information support system for developmental work was designed, placed in service and tested in trial operation in a number of organizations during 1969-1972, and an experimental cooperative of organizations was created, Koopinform, which operates on the principle of decentralized coordinated processing of documents and information interchange via magnetic tapes for the BESM-6 computer. The initial totals for Koopinform activity were obtained in 1974. The system provied to be viable, and Koopinform has been in service up to the present; more than 300,000 documents on physics, mathematics, radio engineering, electronics, automation, instrument construction and other sectors of science and engineering are contained in the aggregate file. Several thousand subscriber with more than 10,000 queries are serviced in various organizations, The "publishing" of information on magnetic tapes is becoming increasingly widespread in our country at the present time. Information began to be produced on magnetic tapes by the All-Union Institute for Scientific and Technical Informat[illegible] VINITI, the All-Union Scientific and Technical Information Center VNTITsentr, the International Center for Scientific and Technical Information MTsNTI, the Institute of Scientific Information on Social Sciences of the USSR Academy of Sciences, INION, and other large information centers.

Practical operational experience with ASIOR demonstrated the importance of the problems being solved by local ASNTI's within the framework of the overall state system. In this case, the tasks of teaching the users how to use the services of automated information systems are assigned to them, as well as learning to correctly assemble local data files, work out mechanisms for adapting systems to users, and also develop schemes and technology for the utilization of the magnetic tapes obtained in the interchange sector.

In the process of operating the ASIOR software, both its positive features and drawbacks were ascertained. A collective discussion of the prospects for the further expansion of the software and taking into account the state of the art and developmental trends in other domestic and foreign information systems have confronted developers with the necessity of selecting one of two possible directions: either modernize the software for the 'old' system, or design a new complex, a successor to the old one, but which takes into account all of the new requirements. As a result of the discussion, the second approach was taken and requirements placed on the new system were formulated. The new system was named the ASIOR-M (MODIS - software for dialog information systems). A special feature of the ASIOR-M software is the fact that it makes it possible to solve a practically unlimited field of information problems, taking over a number of functions previously performed by highly skilled personnel, and provides a high level of service both to the personnel and the users of the system. Despite the fact that individual functional capabilities of the programs are well-known from the descriptions of other systems, the aggregate of the functional capabilities and informational services of such a volume in one package of programs apparently has no equals.

The development of ideas, the design of algorithms and programs, as well as the introduction and operation of the large system are the results of many years of work by a comparatively small staff of specialists. Some of them are the coauthors of materials previously published on the system and the reader will find their names in the bibliography. Others have made a great contribution to debugging the system as well as installing and operating it in various organizations. The authors would like to express their sincere gratitude to all of them.

The authors are especially grateful to academician Andrey Nikolayevich Tikhonov for his constant attention, useful discussions and recommendations for the development of the system.

* * *

The problem of entering, correcting and editing the input information remains quite complex. Only very recently have displays appeared which make it possible to display up to 512 and more different characters on a screen. However, it is necessary to represent each symbol in the computer memory with not just one, but rather three to four bytes, and this complicates the operation of information input, both from punched tape/cards and the keyboards of the displays. These difficulties are of a common nature and their successful resolution would in many respects facilitate the search for more efficient forms of cooperation in questions of interfacing information systems.

This is especially important for international cooperation. Information in the Russian language, even written in the standard communications format, cannot be processed and printed out on the computers of the majority of European nations using the Latin alphabet. First of all, there are only Latin alphabet symbols on the printers of the information centers of these nations, and secondly, the program packages used in these countries do not permit operation with Cyrillic symbols or with characters represented by more than one byte.

There are several approaches to the solution of this problem.

--Purchase data output equipment (printers, displays) having both basic alphabets for the cooperating information centers or for some of them;

--Provide the interchange sector only with information translated into English, with the condition that this information be converted to a form suitable for processing using standard program packages (this approach is technically possible, but can hardly be realized economically in the foreseeable future, since it would be necessary to significantly increase the staff of translators in information centers);

--Provide information in the form of a transliteration of Russian words using letters of the Latin alphabet. Special routines are needed for this. However there is doubt as to the value of this information for the information users, since it is difficult in practice to read and translate it into other languages.

It is now understood what the problems are in the attempt to utilize information on magnetic tapes which is published in nations using the Cyrillic alphabet, or in other non-English speaking nations, for example, the Scandinavian countries. These countries are themselves in a more advantageous position, since the magnetic tapes of foreign systems can also be read and printed out on existing equipment. The information is also completely set up for the major European languages.

* * *

§ 4. Protecting Information. The System Administrator

The problem of protecting the information stored in a computer is a serious matter at present. The development of dialog systems and the utilization of large information files has led to the necessity of making provisions in a system for those measures which would provide access to the data for just those persons having this right.

Protective measures are provided in MODIS, which apply to the administrative, hardware and program portions of the system respectively. These measures guarantee the preservation of information, limit the access to the system and protect the existing data against unauthorized use.

4.1. Administrative and equipment protection. In order to protect any unit of a system, it is necessary to shield it against access by persons not having permission for this. In this regard, definite levels of access and passwords are introduced, corresponding to a definite hierarchy of users, specifically:

1) The system administrator;
2) System programmers;
3) Privileged (system) users;
4) Ordinary users.

The protection system includes procedures, by means of which access to the relevant information can be made available to a definite group of persons, as well as procedures which provide for this access. The responsibility for taking these actions is assigned to the system administrator.

The information system hardware consists of the computer and the auxiliary equipment which belongs directly to it. When solving information problems, the main (immediate-access) memory in which specific MODIS modules and the input/output information are located, is protected by means of the DIAPAK operating systems, which includes blocking of reading and writing in a given memory region. The volumes (magnetic tapes, magnetic disks) of the system on which the archive information (data base) is stored, can be protected against unauthorized intervention if they are incorporated in the DIAPAK ARKHIV operating system.

4.2. Program protection. Program protection tools have been designed into MODIS, which prohibit the disruption of normal operation of the system by means of chance or intentional distortions of the working programs or the service and archive information.

The means of program protection provide for the execution of the following major functions:

1) Monitoring the access to the system in interactive or batch modes;
2) Limiting access to the system to persons registered with the system administrator and the system;
3) Monitoring access to the data bases, the data sets and the service information. which consists in checking the user's right to access data in accordance with their access level;
4) Limiting access to the system, where this limitation permits servicing the queries of users only from definite terminals;
5) Monitoring the change and supplementation of the system program modules;
6) Providing for the preservation and restoration of information.

The most widespread method of prohibiting illicit access to the data is the use of passwords, user identification codes and access ciphers.

Three categories of access are related to each set of data (data base) in the system: For search, additions and modification. For example, if the list of rights of a user contains only the access category for search, then he has the right to use the corresponding data set during a search, but does not have the right to make any kind of changes in it. Access categories are governed by the appropriate access number, assigned to each user. The system administrator, when registering the next user, assigns a definite access number to him, i.e., includes him among the number of system programmers, privileged or ordinary users, with the corresponding authorizations.

A privileged user can be allowed access to definite data sets or bases in any category, corresponding to the functions of maintaining the data files: inspection, replenishment and modification.

A system programmer automatically has access to any service set of MODIS or data set of the system, as well as to definite data bases (at his or the system administrator's discretion) with all the functions of maintaining the data sets and bases.

An ordinary user can have access only to certain data bases without the right of making changes in them.

An important part of protecting information is limiting access to the system, which consists in "attaching" specific users to specific terminals. In the case, it is necessary for the user to not only know the system password, be registered in the system and authorized for definite data bases or sets, but also to work only from that terminal to which he is assigned.

Measures are also taken to protect the program software against unauthorized supplementation or changes in the system modules. A new module can be inserted, or one of the existing ones can be modified only by means of a special program. Attempts to change any module by other methods leads to its annihilation.

There always exists the possibility of mechanical failure (deformation) of the vehicle, equipment failure and program interference from without when information is stored on magnetic tapes or disks, something which can lead to the destruction or distortion of some part of the information. The best guarantee of information safety is the existence of copies of data bases or parts of them. In order to avoid frequent duplication, there is the possibility of storing information (before or after it is written into the data base) in auxiliary data sets in a system volume. A protocol is drawn up for all changes in the data bases or sets.

4.3. The Functions of the System Administrator. The responsibility for the performance of the major operations in the generation of a specific system configuration, the description of the structure of the data bases, as well as the protecttion of the software and data is assigned to a person (or group of persons) conventionally called the system administrator.

The following can be singled out from among the broad group of obligations of the system administrator:

--Assigning passwords to users (registration codes) and changing them at any time;
--Designating and changing the access passwords for the system;
--Monitoring the actual access of various persons to data sets by means of passwords and the appropriate access levels;
--Retrieving the status of service data and control tables;
--Checking successful or unsuccessful attempts to access information;
--Temporary termination of access to the system or to individual data bases (data sets);
--Monitoring all operating sessions with the system in all modes;

--Making changes in the "dossier" of users or the registration of new users;
--Describing the structure of the data bases;
--Providing for the preservation and restoration of information.

The majority of the functions enumerated here is performed by the administrator as a system user. For more flexible determination and modification of the main authorization and protection attributes available to the system administrator, there are means of storing limited use information and a special language: the system administrator language, which takes the form of a set of instructions which are executed only in an administrator mode, and if necessary, from a definite terminal. To initiate this mode, it is necessary to communicate the password of the system administrator to the system.

4.4. The System Administrator Instruction Language. Instructions which are performed only in the system administrator mode are described below.

1. A change in a system password:

ZPS (system password)

The efficacy of passwords can be reduced when working with them, as a result of long term use. To combat this phenomenon, a procedure is used to change the system password.

2. A change of the system administrator's password:

ZPA (administrator password)

It is desireable to change the system administrator's password only with a change in the status of its owner or when it is suspected that the password has been compromised.

3. The establishing of the highest access level for a user (DOP). When registering a new user, he can be assigned the access level of a privileged or ordinary user. The highest access level (system programmer) can be established only in the administrator mode in a dossier set up beforehand.

4. A change in the number of users simultaneously serviced by the system in a dialog mode:

POL (the number of users)

When generating a specific system configuration, the automatic and simultaneous servicing of two users is assured in a dialog mode. This instruction is used to increase the value of this parameter.

5. The output of the contents of a catalog of system sets:

KAT $\{{}^{P}_{E}\}$

where P (or E) is the sign for printout (or screen [E] display).

The printout (or screen display) capability in a form convenient for reading the contents of the overall system catalog (KATNAB) and the INFDOK data sets (INFZAP) is necessary for constant and operationally timely monitoring of the system parameters: The distribution of the physical memory in the system volume; the number of document data files entered and retrieved (queries) which belong to particular users; the distribution of the memory among the user data files.

6. The removal of the violation indicator in a user dossier:

SIN (user registration code)

A user is given only two attempts to present the system password, the registration code or name of a data base (or set). If he commits an error both times (has indicated nonexistent names), then the session is terminated with the given user. A note on the protection violation is entered in his dossier (the violation indicator is increased by a unit), and he cannot start operation until the administrator ascertains the reason for the violation and makes the corrections of the violation indicator.

7. The elimination of a user dossier:

VYB[ROSIT'] DOS['YE] (user registration code)

The instruction is intended to remove the dossier of a user who should not be serviced by the system for any reason. All of the statistical information accumulated in the process of working with the given user is transferred to a service dossier of the administrator. Additionally, all of the permanent queries of the user in the ZAPROS [QUERY] data set are liquidated.

8. The elimination of a data base:

VYB[ROSIT'] BAZ[U] (data base name)

Following the execution of the given instruction in the OPISBD data set, the removal indicator is assigned to the description of the corresponding data base, and all volumes assigned to the data base being dumped are simultaneously freed in the SPITOM data set. The information in corresponding volumes is not destroyed.

9. Output to a terminal of the information on users conducting a session:

KTO [WHO]

The names of the users, the names of the data bases (or sets) and the names of the volumes opened (established) at a given point in time by the users are fed out to the administrator.

10. The transmission of messages to the users:

PER[EDAT'] (administrator message)

The system administrator can transmit a message at any time to any of the users conducting the session. This can be a message of operational session termination, termination of operation with individual data bases (or sets), volumes, etc.

11. Closing off access to the system or to individual data bases (or sets):

```
            (system name)
ZAK[RYT']  {(data base name)}
            (data set name)
```

By indicating the system name or that of any of the data sets (or bases) in the operand of the given instruction, the administrator can temporarily stop all operations with the system, or the processing and utilization of the relevant data. This instruction is used for service purposes (restoration, reorganization of data, etc.).

12. Open up access to the system or to individual data bases (sets):

```
            (system name)
OTK[RYT']  {(data base name)}
            (data set name)
```

It is used as an instruction for renewing operation with the system or its individual components, previously cut off by means of the ZAK[RYT'] instruction.

Besides the special instructions, some other general instructions can also be used in the administrator mode: POM[OSHCH'], STAT[TIST], REG[IST], KON[ETS].

* * *

CHAPTER FIVE. A STUDY OF MODIS

§ 1. The Realization of Information Retrieval Systems Based On MODIS

The automated information retrieval system (AIPS), ASIOR-M, which has replaced ASIOR, was designed around MODIS. During the process of trial operation of ASIOR-M, all of the permanent user queries of ASIOR (4,000) were translated into the MODIS format, a reference information fund (SIF) was built up, which consists of 15,000 documents, and a number of comparative experiments were performed. The SIF documents were distributed among the INIS [International Nuclear Information System] and SNEG [not further defined] data bases. The INIS data base is filled with the documents of the INIS system in English, while the SNEG data base contains documents entered from journals which are processed in Koopinform [22].

At the present time, this system is in active operation both in the IRI [selective information distribution] mode using permanent queries of individual and collective subscribers, as well as in the retrospective search (RETRO) mode to provide users with information for one-time interrogations, and for patent engineering studies. All of the documents incoming to the system are subjected to lexical checking, the dictionaries are supplemented and the system capabilities and characteristics are studied.

In July of 1979, the ASIOR-M had the following characteristics:

The ASIOR-M reference information fund consisted of:

--300,000 documents in the ASIOR format;
--35,009 documents in the MODIS format.

The GOSFAP automated information retrieval system was designed around MODIS in the Computer Center of the USSR Academy of Sciences, where GOSFAP is intended for the needs of the State Library of Algorithms and Programs. The GOSFAP system is in trial operation and is being used in both research (to study the applications areas) and working service.

§ 2. Interfacing Information Retrieval Systems to Other Information Support Systems

The solution of large scientific and scientific-engineering problems, which are as a rule complex in terms of their content, is accomplished by large collectives of specialists in various fields of knowledge. The information requirements of such collectives are polythematic. The interpenetration of the sciences and the objectively existing laws governing the spread of information make it fruitless to attempt to artifically limit information support for the scientific and engineering and technical workers of scientific research institutes and design offices to narrow topical frameworks.

Experience with the functioning of information retrieval systems has shown that their major bottlenecks are the gathering, preliminary machine processing and input of the data into the system. In the case of a large number of information sources, these procedures have proved to be beyond the power of even large organizations. For this reason, the majority of specialists served by systems encompassing a small number of input sources quickly became disenchanted with such systems. The results of theoretical studies and practical experience of a number of foreign and domestic systems attest to the expediency of a division of labor in the processing of information and the exchange of processing results [1-3].

Priority attention was devoted to the question of interfacing information systems for the exchange of information via machine readable vehicles in the development of ASIOR.

The major question in producing cooperation among organizations of users of a single information system consists in resolving the problem of interfacing computers through magnetic tapes or disks [22]. In this case, a number of auxiliary problems confront the software: the organization of data files on magnetic tapes (or disks) in a form convenient for interchange, strict monitoring of input errors and simplification of the operation of the programs. The problem of interfacing various information systems, realized with different types of computers, is also important. This task includes the problem of interfacing different machines through plug-in vehicles (magnetic tapes, magnetic disks), the problem of standardizing the input information, i.e., coming up with some standard for the specification of the input data as well as the problem of language compatibility of the systems. This problem is apparently being solved only at the natural language level, which has been adopted as the basis for the construction of diverse service languages in many systems.

An experiment in decentralized processing of documents, the design of collated data files on magnetic tapes and providing the information from these data files to "Koopinform" users is discussed in the literature [22]. At the present time, the ASIOR-M, just as the ASIOR, is interfaced with the INIS International Nuclear Information System by means of reading IBM magnetic tapes on the BESM-6 and translating the information from the INIS format to the ASIOR-M format. A "null grammar" is assigned to all of the terms of the abstract (including the noninformational ones) and the descriptors, incorporated in the document in the indexing stage by means of the INIS thesaurus, when translating to the ASIOR format. Such an algorithm leads to large retrieval forms of the documents because of the large number of assigned descriptors and terms in the abstract, something which in turn produces large losses of time during a search. On the other hand, the lack of any links between the terms of the retrieval form of a document is a cause of the large outputs of irrelevant documents (word combinations which represent descriptors are broken down into independent elements).

The MODIS is completely compatible as a successor to the ASIOR information data files, regardless of their type of indexing (grammarless or with a grammar), because of the presence of a converter which converts the data files to "natural form", as well as a number of algorithms for automatic input data processing. The presence of a data description language makes it possible to accept any information written in an identification type format in a natural language; in this case, the role of the converter consists only in decoding the service information on a magnetic communications tape.

In turn, information retrieval systems based on MODIS can exchange information on magnetic tapes, selected in accordance with any of the retrieval or service attributes, both without preliminary syntactical and lexical monitoring (where data bases of the same structure are present) and with monitoring at the input and writing into any other (or its own) data base. In this case, the documents are automatically inventoried in both instances. Moreover, such information retrieval systems can transmit their own information files in the ASIOR format to systems which are different variants of the ASIOR system. There is a converter for these purposes in MODIS, which translates the information from natural form to the ASIOR format, automatically assigning grammatical categories to significant terms of chapters and abstracts.

The algorithm for translating to the ASIOR format takes the form of a simplified variant of an auto-indexing algorithm, the purpose of which is to assign two major grammatical categories (process, action, characteristics, properties; the appellation of matter, energy or information as a major message unit) and the relevant characteristics (qualitative, quantitative, unit of measurement, beginning of an interval quantitative characteristic; the end of an interval quantitative characteristic). The following experiment preceded this choice of grammatical categories. To estimate the distribution of grammatical categories among the terms and data files of ASIOR, experimental data files were picked out having an overall number of documents of about 42,000 (10,000 documents for each file) in accordance with the following topical headings (rubric designator of the VNTITsentr [All-Union Scientific and Technical Information Center]):

01 - mathematics; 02 - physics, mechanics; 12 - radio engineering, electronics and electrical communications; 07 - mining; 10 - machine building; 11 - power engineering; 09 - metallurgy; 15 - water transport; 17 - polygraphy and motion picture engineering; 23 - economics; 24 - the study of the sciences; 25 - scientific and technical information.

Using a program to generate a dictionary of stems, dictionaries were issued in a word selection mode in accordance with the following grammatical categories:

1. Process, action, characteristics, parameters, properties;
2. The appellation of matter, energy or information as a main message unit;
3. The appellation of the constituent parts, components, ingredients;
4. The appellation of the acting factors, agents;
5. The appellation of transmitters and recorders of an effect (equipment, instruments);
6. The appellation of the medium in which an object is present or a process takes place.

The results of the experiment are given in Table 19 [not reproduced], from which it can be seen that in almost all of the subject areas, two major grammatical categories predominate: 1 and 2. Among the remaining categories, 5 is singled out, which is a special case of 2. For this reason, 1 and 2 were adopted as the main categories when translating.

Considering the fact that the MODIS program software has a modular structure, which is "open" to the expansion of the functional capabilities of the system, the connection of any new converter is possible whenever this is dictated by changing circumstances. Thus, the utilization of data bases on magnetic tapes, prepared by the automated ASSISTENT system (VINITI) [(All-Union Institute of Scientific and Technical Information of the State Committee on Science and Technology, USSR Council of Ministers and the USSR Academy of Sciences)] in the ASIOR-M is planned for the immediate future.

[75-8225]

8225
CSO: 1863

SAVINGS FROM COMPUTER APPLICATIONS

Moscow EVM DLYA VSEKH in Russian 1980 signed to press 14 Nov 79 pp 73-74, 78-80, 87-89, 91-92, 107, 110-112, 123-125, 128, 134, 172-174, 181-182

[Excerpts from book by Vladimir Aleksandrovich Myasnikov, Sergey Aleksandrovich Mayorov and Gennadiy Ivanovich Novikov, Izdatel'stvo "Znaniya", 77,380 copies, 192 pages]

[Excerpts] Organization of the Design and Introduction of Automated Control Systems for Planning Calculations (ASPR's)

Work on the design and introduction of ASPR's in the republics is being carried out under the direction of the USSR Gosplan.

In 1976 the interdepartmental commission accepted the first phase of the Lithuanian SSR Gosplan ASPR. The system's structure includes 10 combined, 19 industrial and nine supporting subsystems.

The Lithuanian SSR Gosplan has been assigned responsibility for the development of six functional subsystems and for the overall design of a Gosplan ASPR for a Union republic without division into oblasts. These subsystems include "Capital Investment," "Planning and Research," "Labor and Personnel," "Cost and Profit," and the industrial subsystems "Local Industry" and "Construction and the Construction Industry." In addition, a "National Economic Balance" standard module is being developed for the functional subsystem "Combined National Economic Plan."

The total number of problems included in the first phase is 190, of which 20 come under the heading of longterm planning, 28 of medium-term planning and 141 of current planning.

One hundred and thirty-four problems have already been introduced (70 percent) and the remaining have passed experimental use tests and are ready to be introduced.

In October 1976 the first phase of the Ukrainian SSR Gosplan ASPR was surrendered to the interdepartmental commission. It is characterized by the fact that problems are solved by computer technology and their results are used for the purpose of developing indicators for the republic's national economic plans; the problems are mutually compatible in the technological and information sense, forming a system of methodologically compatible problems.

The first phase of the Ukrainian SSR Gosplan ASPR includes 21 functioning subsystems (including seven combined and 14 industrial) by which planning documentation is developed to the full extent.

The total number of problems solved is 574, of which 346, i.e., 60 percent, have been united into 75 groups.

In the annual planning mode are solved 357 problems, or more than 60 percent of the total number of problems, in the five-year planning mode 171 problems (30 percent) and in the longterm planning mode 46 problems, i.e., 10 percent of all problems.

The effectiveness of the introduction of the first phase of the Ukrainian SSR Gosplan ASPR was determined by the savings gained in the republic's national economy as the result of the optimization of planning decisions, as well as of the lowering of costs resulting from the automation of planning calculations. The annual savings from phase one of the Ukrainian SSR Gosplan ASPR equals 10.73 million rubles, and the expense recovery period equaled nine months.

Tasks Performed by the USSR Gosbank Central Computing Center

The Gosbank Central Computing Center (GVTs) processes the following types of data: intrabank reporting, data on statistics and credit planning, on cash transactions of the USSR Gosbank and cash transactions of the State budget, on overdue indebtedness with regard to Gosbank loans and on accounts not paid in time, on balance reporting, on longterm credit, on shortterm credit for kolkhozes and on the flow of kolkhoz capital toward capital investment, on the financing of capital investment and other measures relating to agriculture, and on the status of the expenditure of wage funds.

The compilation and creation of a credit plan for many ministries and departments of the country are also carried out.

Information relating to all these tasks (with the exception of the task of calculating wages) arrives at the Gosbank Central Computing Center through communications channels (a subscriber telegraph system) from all Gosbank oblast offices.

The total amount of information processed for these tasks equals, in terms of punched cards, about 200,000 per month.

Since 1968 the USSR Gosbank GVTs has managed an industrial solution to the problem "Receipting Interbranch Transactions" for all 34 Gosbank institutions in Moscow.

An experimental industrial solution is under way for the bank problem "Gosbank Operations Day" employing two combined computers (data are processed from 33 rayon institutions in Moscow).

The problem "Foreign Trade Bank (VTB) Operations Day" is being put into service. This problem is similar in terms of its structure and the kind of developments to be carried out to the problem "Gosbank Operations Day."

The entry into experimental industrial service of the "Bank" system with the formulation of a solution using a computer to the problems "Gosbank Operations Day" and

"Receipting Interbranch Transactions" have already made it possible to carry out the overall automation of the majority of bank processes. Furthermore, the quality, timeliness, completeness and number of developments presented have been improved considerably. In addition, the introduction of computers has made it possible to draw the conclusion that a slowing will occur in the growth of the number of the bank's accounting operations personnel as compared with the growth in the number of operations performed by the USSR Gosbank.

After the entry into experimental industrial service of the "Bank" system in Moscow the USSR Gosbank GVTs began to develop and prepare the organization of "Bank" computing complexes for 14 large industrial cities of the country.

At the present time the problems "Operations Day" and "Receipting" are being solved in Gosbank's oblast offices in Gor'kiy, Novosibirsk, Rostov, Chelyabinsk, Donetsk, Sverdlovsk, Kuybyshev, Khar'kov, Alma-Ata, Odessa, Voronezh, Irkutsk and Kiev.

The USSR Gosbank Central Computing Center has also carried out scientific research and experimental design developments relating to data preparation and transmission equipment (APD).

A technological algorithm has been developed for the preparation, transmission and receipt of data by the APD of the "Bank" system for the independent problem "Receipting MFO [Interbranch Transactions]"; a technological algorithm has been developed for the preparation, transmission and receipt of data directly from communications channels to computers of the "Bank" system.

For the purpose of transmitting data the central computing center uses a subscriber telegraph network making possible communications between the administration and peripheral institutions of the USSR Gosbank.

The central computing center is continuing to develop remote communications for the Gosbank for the country as a whole, which has made it possible to create the foundation of a communications network including cable line facilities and a direct-wire, subscriber line and station equipment network and to develop experience in the combined work of bookkeeping and technical personnel.

Employment of Computers in Power Engineering

The introduction of computer technology and the creation on its basis of ASU's [automated control systems] are being carried out in organizations and at enterprises of the USSR Minenergo [Ministry of Power and Electrification] for the following major levels:

An automated control system for a sector of industry (OASU).

An automated dispatcher control system (ASDU) for united power systems (OES's).

Automated systems for controlling (ASU's) power systems.

Automated systems for controlling construction and installation trusts and enterprises of the construction industry (ASUS's).

ASU's for electric power plants.

ASU's for technological processes of large-capacity power plants.

The USSR Minenergo Industrial Automated Control System (the "Energiya" [Power] OASU) is a complex system including the following:

An automated system for organizational and economic control (ASOU).

An automated system for dispatcher control (ASDU) of the USSR Unified Power System (YeES).

An automated system for controlling construction (ASUS).

Operations data on the technical and economic indicators of power systems and major electric power plants, fuel reserves, the presence of water resources in hydroelectric power station reservoirs and the state of key equipment are transmitted every 24 hours through a hierarchical system for gathering alphanumeric data. Information processed at the USSR YeES TsDU [Central Dispatching Administration] by means of computers and computer punching equipment arrives at the management of the USSR Minenergo and central administrations. Gathered and processed similarly every 10 days, every month and every quarter are statistical data on the flow of fuel, the generation and consumption of electric power, specific consumption of fuel, and on construction progress and deliveries of resources for the most important construction projects.

In the structure of the ASDU is solved one of the major problems of the operations and dispatcher control of the power industry--optimization of USSR YeES conditions in relation to active output, taking into account the economic characteristics of units and losses of electric power in networks, making it possible to achieve a considerable savings of fuel. These calculations are performed every 24 hours at 23 interacting VTs's [computing centers].

Plans for the entry of power capacities into service are designed in the structure of the ASUS.

The "Energiya" OASU services 14 functional subdivisions of the USSR Minenergo apparatus, for which as many as 1300 estimates are made per year, and 82 forms of operations estimates. In addition, 27 problems are solved and a total of 69,500 calculations per year are performed for the USSR YeES TsDU.

The entry into service of the just the first phase of the "Energiya" OASU in 1975 made it possible to improve the economic efficiency of the work of the Unified and of united power systems, taking into account the fulfillment of established norms for the quality of electric power and the reliability of power service to consumers; to reduce costs and to speed up power and subcontracting construction; and to reduce the cost of industrial products and to shorten the time required for preparing the planning and estimating and information materials required for management of the industry.

The annual savings from introducing the automated system has equaled 1.664 million rubles, and the recovery period 2.6 years.

By 1976 had been put into service the first phases of the "Energiya" OASU, 10 OES ASDU's, 18 power system ASU's, five electric power plant ASU's, 14 ASUS's in construction trusts and in construction industry trusts and 48 ASUTP's [automated control systems for technological processes] for large-capacity power plants.

In power system ASU's business management problems are solved, records are kept of operating costs for fuel at electric power plants, records are kept and an analysis is made of the utilization of materials and equipment resources in the power system's storehouses and bases, the presence and turnover of personnel at power system enterprises are monitored, etc.

Automated systems for the dispatcher control (ASDU's) of power systems are created at all upper control levels as independent systems for their technological control or as part of a combined organizational and technological ASU.

Control of Technological Processes in Power Engineering

Data services for operations personnel are made possible by means of ASUTP's. These include the retrieval of data for display, monitoring the key parameters of the operating conditions of a power plant, the documentation of operations data and the recording of operating condition parameters for the purpose of analyzing emergency situations.

ASU's based on small computers have been put into service at large-capacity thermal electric power plants, such as, for example, the Butyrskaya, Zmiyevskaya, Moldavian and others. General station problems are solved in these systems in addition to solving problems relating to controlling power plants.

At the Moldavian GRES output forms issued by a computer represent the only documents used for crediting personnel bonuses, summing up the results of socialist competition between watches and writing monthly reports.

The complexity of the technological process and the advanced monitoring system have determined the desirability of using computers for controlling a nuclear power plant (AES). Thus, new improved ASUTP's based on small computers have been created at the Beloyarsk and Leningrad AES's.

The number of monitored parameters increases drastically with an increase in the capacity of nuclear power plants. Computers installed at an AES perform the functions of gathering data, controlling signalling and recording parameters, and of calculating TEP's [technical and economic indicators].

At hydroelectric power plants and hydroelectric power plant series an ASUTP implemented on the basis of computer technology makes it possible on the basis of centralized data to control a GES under normal and emergency conditions.

ASUTP's have already been put into service at the present time at the Vatchinskaya and Saratov hydroelectric power plants. ASUTP's have been put into service at other GES's, including at the Krasnoyarsk GES, the largest in the world. An ASUTP is being developed for the Sayano-Shushenskaya GES and the Zagorsk and Kayshadar GES's.

The ASUTP's of large GES's have been called upon to solve the problems of the automatic control of standard and emergency conditions taking into account the effects of counteremergency and system automatic equipment.

The average cost of an automated system for controlling a technological process based on a third-generation computer equaled in 1971-1975 750,000 rubles, whereby approximately 65 percent of the cost was due to hardware (approximately 40 percent for computer equipment and 25 percent for automation devices and equipment).

The majority of ASUTP's pay for themselves in 1.5 to two years. The rapid recoverability of the cost of these systems is made possible on account of a two- to seven-percent increase in product output, a two- to four-percent reduction in the expenditure of material resources, a two- to five-percent increase in the output-capital ratio, etc.

The high effectiveness of these systems can be illustrated by other examples, too.

The employment of an ASUTP for the electrolysis of aluminum produces a savings equal to 1.5 percent of the electric power consumed by this process.

ASUTP's for reheating furnaces, making it possible to heat an ingot before rolling, make it possible to save 1.5 to two percent of the metal on account of the reduction of scale losses. With a furnace throughput of 0.5 to 1.5 million tons of ingots per year one system can produce a savings of 7,500 to 30,000 tons of metal.

Automation by means of computers of technological processes at the concentration plant of the Zyryanovsk Lead Combine has made it possible to increase the extraction of key elements from ore by 1.5 to four percent (including by 4.39 percent for copper, 1.97 for lead and 1.55 percent for zinc), to improve the quality of commodity concentrates, to increase the volume of ore processing by five percent, to reduce the consumption of auxiliary materials by three percent and to increase labor productivity by 66 percent, as the result of which the number of workers has been reduced by 100.

An automated system for controlling the primary oil refining plant at the Novo-Yaroslavskiy Oil Refining Plant has increased productivity by three percent and has reduced the consumption of fuel by four percent, of water vapor by 16, of circulating water by four and of electric power by four.

The "Karat" system for controlling mining transportation equipment in operation at the pit mine of the Yakutalmaz Association has made it possible to reduce the idle time of excavators and motor transport and to improve the employment of dump trucks by seven to 10 percent, which in 1975 produced a savings of 680,000 rubles. The cost of this system was paid back in less than one half year. A similar system introduced at the Tamusinskiy Open-Cast Coal Mine has made it possible to increase the output of coal with existing capacities by 4.5 percent.

An ASUP for the production of urea at the Severodonetsk Chemical Combine has made possible complete centralization of control of the technological process and on account of the optimization of operating modes of units has made it possible to increase product output by 3.5 percent, to reduce the consumption of raw materials by 1.5 percent, and to reduce harmful effluents eightfold. The number of operating personnel has been reduced by 7.5 percent.

At the Dzerzhinskiy Plastics Plant an ASUTP for the production of phenolic acetone has made it possible to produce an improved-quality phenol, and this has made it possible to award this type of product the State Emblem of Quality. In addition, the output of finished products has been increased by 6.6 percent, the consumption of raw materials has been reduced by 0.6 percent, the consumption of electric power by six percent, and the number of management personnel by eight percent. As the result of introducing this system management of the business was centralized and the number of sections was reduced from five to three.

With automation of the control of harmful and burdensome production processes in the chemical, oil refining and petrochemical industry as much as 30 percent of machine operators working at large units have been released, and the remaining have been removed from areas with harmful working conditions.

About 1300 ASUTP's will be put into operation in the 10th Five-Year Plan period (2.5 times more than in 1971-1975). The major portion of this program (more than 50 percent of the total work) is being carried out by heavy industries (the chemical, oil refining and petrochemical industry, ferrous and non-ferrous metallurgy and the like), which in the Ninth Five-Year Plan period gained great experience in creating automated systems. More than 100 systems were created in the machine building industry. Other industries (light industry and the food industry and the like) will gain experience in using computers for controlling technological processes and will create prototypes of systems.

A program of scientific and technical work has been devised for creating 75 ASUTP prototypes for key technological processes. This will make it possible to reduce considerably the cost of developing systems for similar projects.

In keeping with overall programs approved by the State Committee for Science and Technology, methodological materials are being developed, governing the procedure for creating ASUTP's, estimating their economic efficiency, calculating and norm-setting the reliability of systems, etc. The same plan has called for the development of collections of standard control algorithms and of software implementing them for various sectors of industry.

In conclusion it should be noted that ASUTP's should be created primarily for new or reconstructed technological systems, since the creation of systems at existing sites requires additional modifications of equipment.

For purposes of ensuring an improvement in the operating efficiency of complex technological equipment on the basis of automation, new complicated technological processes, units and production processes must be designed by employing automated control systems based on modern computer technology. The coordinated design of technological equipment and ASUTP's as an inseparable whole is today a necessary condition for the creation of new effective technological complexes.

Economic Efficiency of an ASU for Transportation

The development and introduction of an OASU in railway transportation has made it possible to reduce by 0.5 percent the empty mileage of cars. The annual savings has equaled 3.5 million rubles. The development and introduction of an ASU for a sorting yard has made it possible to automate current planning and to reduce the idle time of cars and to automate the process of selecting the optimal sequence for sorting trains at sorting yards. The calculation of full-scale sheets with a computer has substantially improved the quality of documents and has practically eliminated cases of the separation of documents and cars, which in the end result has reduced the number of incorrectly sent and lost cars. The annual savings from introducing these calculations has equaled 300,000 rubles. The introduction of the first phase of the "Ekspress" system at the Moscow center has made it possible to increase the number of occupied seats in a car by 2.5 to three percent on account of the more efficient sale of tickets. The annual savings from functioning of this system has equaled four million rubles.

In air transportation the development and introduction of an OASU has made it possible to improve the profitability of Aeroflot operations, to gain an additional profit as the result of an increase in flying hours, to increase profits on account of a reduction in the down time of aircraft equipment in waiting for modifications, to increase the pay load of aircraft, and to reduce the down time of aircraft resulting from a lack of spare parts. The annual savings has equaled nine million rubles.

The beginning of the 10th Five-Year Plan period was marked by the birth of the "Start" ASU for air traffic. Its industrial utilization began in Leningrad. Information on air traffic processed with computers helped the controller select the most optimal and safe course for an airplane and to determine its altitude and velocity. The "Start" system has not only improved the regularity of flights, but has also helped considerably to improve the traffic capacity of an airport.

Highly complicated electronic equipment has become a reliable assistant to takeoff and landing "conductors."

An ASU has made it possible to reduce the time required, to improve the quality and to lower the cost of repairs of the aircraft engine inventory at aviation plants. The annual savings from its introduction has equaled 400,000 rubles.

An airport ASU (for the Vnukovo airport, for example) has made it possible to improve the regularity of the fulfillment of flight schedules and to reduce the idle time of aircraft on account of the efficient presentation of data on the state of preparation for the completion of trips, on account of an increase in flying hours for an aircraft, as well as on account of an improvement in traffic safety because of optimal planning of the destination of aircraft for trips and for equipment servicing. The annual savings from carrying out this measure has equaled 300,000 rubles.

The introduction of the "Sirena" [Siren] system at the Moscow center has made it possible to increase the occupancy rate of seats in aircraft by three percent, which produced a savings of more than three million rubles for the first year of operation (1972). The labor productivity of cashiers and controllers increased three- to fourfold. The system made it possible to automate about 90 percent of all technological operations and procedures. It has made possible simultaneous access to central computers by 250 cashiers and controllers in Moscow, at its airports and in a number of large cities in the country. Cashiers and controllers have been supplied with displays and carry out their work with a computer by communicating with it in real time. About 500 more cities in the country have been linked with the "Sirena" by a subscriber telegraph network. During the time of its operation the "Sirena" has served more than 20 million passengers, including eight million people in 1975.

Computers in Public Health

In the 10th Five-Year Plan period the sector automated control system for public health will receive additional development--the number of subsystems is being increased and the number of treatment and scientific institutions serviced by the ASU is being expanded. During the five-year plan period 47 new automated computing and data processing systems and 29 computing centers will go into service.

The Statewide Automated Control System (OGAS)

At the present time more than 330 OASU's and 2000 ASUP's [automated enterprise management systems] are in operation. The average time for recovering their cost is about three years.

A few thousand general-purpose third-generation computers will be produced in 1976-1980 for OASU's, ASUP's, for scientific and planning organizations and for other purposes.

The course is being taken toward the creation in the country of large-capacity computers (with a speed of five million, 10 million and more than 100 million operations per second) and of control computers, which is the optimal course and corresponds to world trends.

Computer technology has become an integral part of the automation of complex technological processes in chemistry, petrochemistry, power engineering, ferrous metallurgy and other industries (ASUTP's). Whereas prior to 1971 there were 170 automation systems based on computers, 619 systems were created during the Ninth Five-Year Plan period. The time for recovering their cost is no longer than two years.

The intent is to create more than 1300 ASUTP's in various sectors of the national economy in the 10th Five-Year Plan period (1976-1980). Minpribor [Ministry of Instrument Making, Automation Equipment and Control Systems] will produce several thousand control computers in the 10th Five-Year Plan period for these and other purposes.

An analysis of the demand for computers has shown that it is not being satisfied completely by applicants with regard to control and general-purpose computers. Computing centers (and now there are about 3000 of them in the country) have still not been fully furnished with peripheral equipment (rapid printers, magnetic disk and tape storages, etc.), and there is an insufficiency of data transmission equipment and of operating materials (paper and magnetic data media, etc.).

These and other shortcomings (the unprepared state of tasks among users, the lack of readiness of space and the like) have hampered the effective utilization of computer technology (VT) and, in particular, have resulted in the fact that on average the computer load in the country equals about 12 hours instead of 16 to 18 hours a day.

In spite of these shortcomings, as the experience of the Ninth Five-Year Plan period has demonstrated, the effectiveness of capital investment in VT and ASU's is greater than for other lines of capital investment. A planned savings of 33 kopecks (an efficiency factor of 0.33) had been established in the Ninth Five-Year Plan period for each ruble of capital investment in a VT project; the efficiency factor actually equaled 0.4.

The OGSPD [Statewide Data Transmission System] should consist of communications lines with a carrying capacity of from 200 to 48,000 baud (bits per second), channel and message switching centers (TsKS's), data transmission equipment, etc.

A TsKS will contain in its structure special equipment for consolidating, coding, and, when necessary, encipherment of data. In the first phase they will be created in large cities in which it is planned to create communications networks, as well as in VTs KP's [collective-use computing centers].

In creating the OGSPD it is necessary to provide for the unification of separately operating communications networks of various fiscal agencies for the purpose of increasing their load. The reliability of data transmission by the OGSPD should be not worse than 10^{-7}.

For the purpose of creating the OGSPD it is possible in part to use the already existing communications system. However, many problems must be solved from scratch. It is necessary to construct a new system of channel and message switching centers and to increase the throughput of communications lines both by extending old and constructing new lines.

[73-8831]

8831
CSO: 1863

PUBLICATIONS

NEW BOOK REVIEWS MONITORING, TECHNICAL DIAGNOSIS OF COMPUTER SYSTEMS

Kiev TEKHNICHESKAYA DIAGNOSTIKA, EKSPLUATATSIYA UPRAVLYAYUSHCHIKH I VYCHISLITEL'NYKH MASHIN in Russian 1980 pp 2, 212-213

[Annotation and table of contents of book "Technical Diagnosis and Operation of Controlling and Computing Machines", Izdatel'stvo "Naukova dumka", 224 pages]

[Excerpts] Annotation

This collection of articles considers theoretical and applied questions of monitoring and technical diagnosis of computing and controlling systems and devices. It presents methods and results of practical work on building and analyzing the functioning of computing complexes and monitoring systems, techniques for evaluating the reliability of complex systems, and procedures for constructing test programs to check the working condition of the computing devices.

The book is intended for scientists, engineering-technical personnel, graduate students and undergraduates at higher educational institutions who are working on the problems of monitoring computing and controlling systems.

Table of Contents

[81-11176]

11,176
CSO: 1863

UDC 681.5.001.2

THE AUTOMATION OF THE DESIGN OF OPERATIONAL CONTROL SYSTEMS FOR TECHNOLOGICAL PROCESSES

Moscow AVTOMATIZATSIYA PROYEKTIROVANIYA SISTEM OPERATIVNOGO UPRAVLENIYA TEKHNOLOGICHESKIMI PROTSESSAMI in Russian 1980 signed to press 28 Dec 79 pp 4-5, 7, 287-288

[Annotation, table of contents and excerpts from the foreword in the book by G.G. Chogovadze, edited by Acadmicians V.A. Trapeznikov and A.A. Voronov, Doctors of Engineering Sciences A.G. Mamikonov and O.I. Aven, and Candidate of Engineering Sciences D.M. Berkovich, Izdatel'stvo "Energiya", 5,600 copies, 288 pages]

[Excerpts] Questions of automating the design of information and program complexes which realize the operational control of production processes are treated in the book. The basic principles of operational control system design theory are presented. Mathematical methods are widely represented, where these are used in the solution of operational control problems. Problems are analyzed which are related to the planning of data organization and the automated synthesis of programs.

The book is intended for specialists in the field of ASU design, directly involved in the solution of information and programming problems.

Foreword

An attempt is made for the first time in this book to treat the entire process of developing operational control systems for technological processes from unified viewpoints, starting with the substantiation of the necessity of control automation and concluding with the development of a working project plan for the information and program complex.

The materials of this book were the result of research carried out by the staff of the Department of Automated Control Systems of the Georgian Polytechnical imeni V.I. Lenin, directed by the author.

The discussion of this problem area in the scientific collectives of the Department of Cybernetics of the Moscow Engineering and Physics Institute, the Institute of Control Problems of the USSR Academy of Sciences and the Moscow Higher Engineering School imeni Bauman had a considerable impact on the quality of the material in the book.

Table of Contents

[75-8225]

8225
CSO: 1863

APPROVED FOR RELEASE: 2007/02/08: CIA-RDP82-00850R000300080023-5

NEW BOOK REVIEWS OPTIMIZATION OF DYNAMIC SYSTEMS OF RANDOM STRUCTURE

Moscow OPTIMIZATSIYA DINAMICHESKIKH SISTEM SLUCHAYNOY STRUKTURY in Russian 1980 pp 4-8

[Annotation and table of contents of book "Optimization of Dynamic Systems of Random Structure", by I. Ye. Kazakov and V.M. Artem'yev, Izdatel'stvo "Nauka", 3200 copies, 382 pages]

[Excerpt] Annotation

This book presents general and applied methods of solving the problems of signal filtering and statistical optimization of dynamic systems which have different structure at random time intervals. Such tasks arise in the control of mobile objects, industrial processes, robots, and other technical devices. They are a result of full automation of complex control processes in modern engineering.

These problems are considered on the basis of the theory of conditional Markov processes which leads to the need to determine the a posteriori density function of the probability of the vector of state. Approximate algorithms put in computers are also considered. They are illustrated with examples.

The book is intended for scientific researchers, engineers, and advanced college students who are specializing in automatic control. It has one table, 48 illustrations, and 63 bibliographic entries.

Table of Contents — Page

[79-11176]

11,176
CSO: 1863

END