JPRS-UCC-85-004

27 August 1985

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

FBIS FOREIGN BROADCAST INFORMATION SERVICE

JPRS-UCC-85-004

27 AUGUST 1985

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

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.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service (NTIS), Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semimonthly by the NTIS, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS-UCC-85-004

27 August 1985

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

CONTENTS

COMPUTER CENTER HEAD COMMENTS ON YES-1066 , YES-1046 MACHINES

Moscow SOVETSKAYA LITVA in Russian 22 Feb 85 p 4

[Excerpt] The new computers "YES-1066" and "YES-1046", which were developed in our country, operated for hundreds of hours continuously without a single malfunction. A state commission has recommended them for series production. The commission's chairman, academician A. Dorodnitsyn, director of the USSR Academy of Sciences' Computer Center, spoke about the merits and applications of these computers:

"The new machines open a new chapter in the development of the class of medium-capacity computers, which are used widely in industry. The first thing that is noteworthy about them is their high reliability. A malfunction which occurs in one of the computer's units will not put the whole machine out of order. Switching off the faulty unit, the computer readjusts itself to bypass this unit. Operation continues, at a somewhat slower pace, of course; the main thing is that the operational reliability of the new computer is not impaired.

"Another merit of the new machines is their higher capacity. In comparison with previous models, the speed of the 'YES-1066' has been increased from 2 to 5.5 million operations per second.

"Specialists of the Yerevan Scientific Research Institute of Mathematical Machines were very successful in developing these machines. According to the technical assignment, the 'YES-1046' was supposed to have a speed of one million operations per second. The developers were able to push this number up to 1.3 million operations per second. And even this number is not the limit. By introducing microprogram units, which perform frequently-encountered operations at an accelerated pace, the computer's speed can be raised to 3 million operations and more per second for many classes of tasks.

"The users of both machines are very satisfied with them. These users are large research institutes and design bureaus, plants, and industries, which use them in management information systems".

FTD/SNAP
CSO: 1863/218

RAISING THE RELIABILITY OF A REMOTE PROCESSING SYSTEM

Moscow AVTOMATIKA, TELEMEKHANIKA I SVYAZ' in Russian No 3, Mar 85 p 34

[Article by V. Ye. Beloshevskiy, chief, Electronics Department, Belorussian Railroad Computer Center]

[Text] In order to raise the reliability of the system created on the basis of the YeS1035 computer, the Electronics Department has developed and introduced some improvements in the devices and the procedures of working with them. First of all we replaced the Konsul-260.1 and Konsul-256 electromechanical typewriters with Videoton-340 and Videoton-52100.C video terminals. In order that the Videoton-340 could work with the YeS1035 computer, the circuit for recoding the typewriter adapter and Videoton -340 codes had to be developed, and the control signals had to be matched.

A special test bench was created for repeat testing and weeding out of failing 565RU1 memory microcircuits. The problem is that these microcircuits are very sensitive to temperature changes. When the processor fails, the system becomes uncontrollable. With introduction of redundancy, the consequences of failures may be minimized. But this immediately increases the system's cost, because all lines would have to be duplicated in the system, and an expensive processor would have to be placed in ready reserve.

There is another way possible--introducing a switching processor between the main and back-up computers; on detecting errors in the system, this processor would switch the system to the alternate processor or communicate presence of errors. This is the path we selected. In the first stage, after certain moments in time we transfer records of computer and channel errors contained in a SYS1.LOGREC data unit to magnetic tape for subsequent processing with an Elektronika-60 microprocessor. OBR, MCH and CCH recordings are selected from among recordings in the file. The data contained in these recordings are decoded using reference manuals and printed out or displayed in a prearranged form reflecting only mistakes. In this case data on the system's status at any moment in time may be obtained from the Elektronika-60 console, indicating the date and time.

In order to process data read from magnetic tape by the YeS1035 computer into the Elektronika-60 microprocessor, we had to write a recoding, editing and processing program. In terms of its volume the document obtained with an Elektronika-60 microprocessor is significantly lower than the print-outs of the SYS1.LOGREC unit obtained with a YeS computer using an EREP program.

However, it does provide exhaustive information on the status of the system to permit adoption of the necessary decisions.

The department is presently examining the possibility of connecting an Elektronika-60 microprocessor right into the system. By monitoring channel and computer errors in real time, we will make it possible to efficiently switch the remote processor to the alternate computer, since four YeS1035 computers are brought together into a single system by way of a shared memory field consisting of magnetic disks. Channel programs have been developed for this purpose, and circuits making it possible for an Elektronika-60 microprocessor to communicate with a YeS1035 computer were created. The technical possibilities of the Elektronika-60 microprocessor are being expanded through the connection of supplementary devices. A subprogram that will be connected into the core of an OS6.1 supervisor and which will make it possible to transmit recordings of channel and computer errors from the monitoring circuits directly to the Elektronika-60 microprocessor for processing and analysis of data on the system's performance is being written.

11004
CSO: 1863/221

CACHE MEMORY OF THE INPUT-OUTPUT REGISTER OF A YES8401 DATA TRANSMISSION MULTIPLEXOR

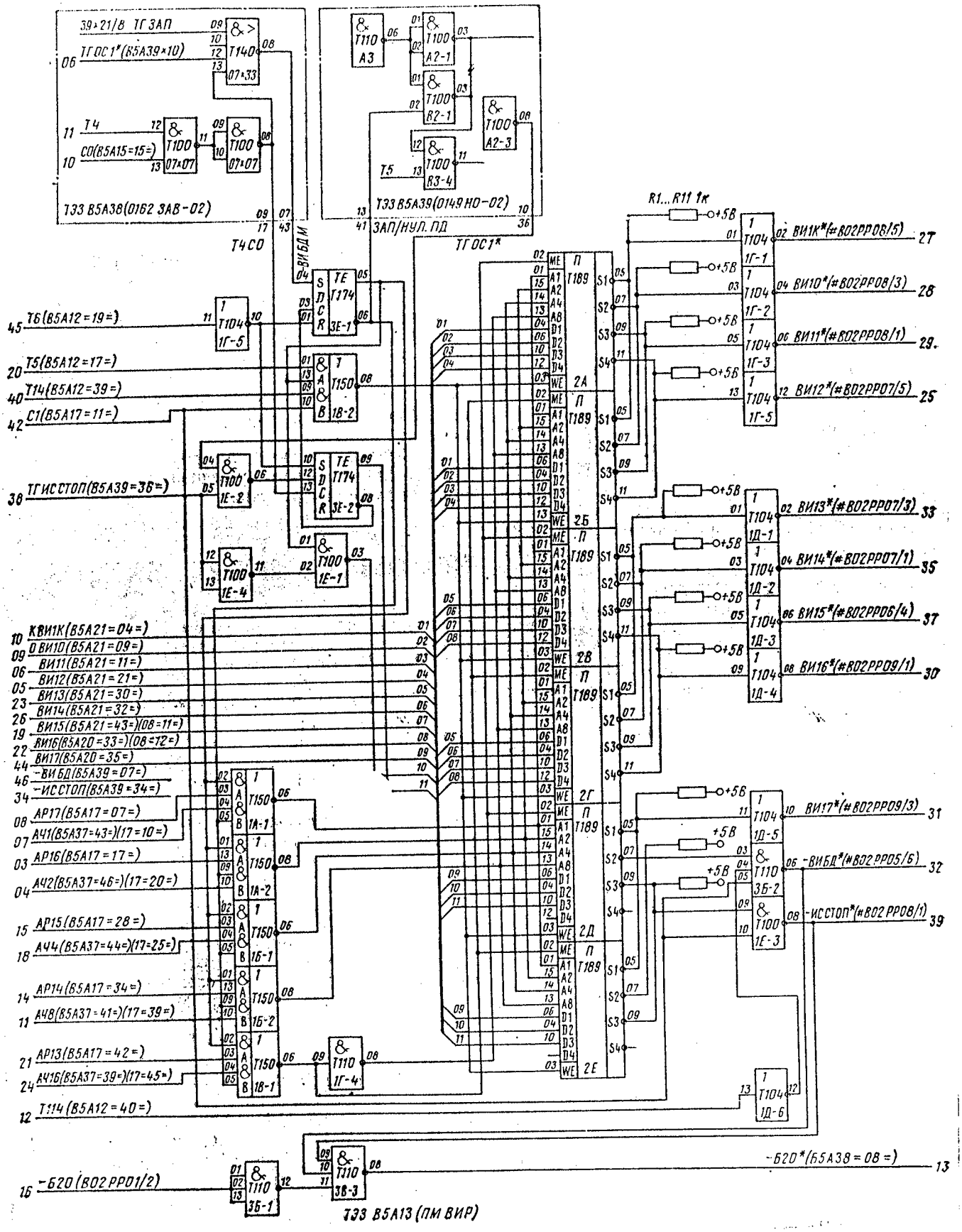
Moscow AVTOMATIKA, TELEMEKHANIKA I SVYAZ' in Russian No 3, Mar 85 pp 38-40

[Article by O. F. Aniskevich, department chief, Belorussian Railroad Computer Center, and I. B. Pakhomchik, technician]

[Text] To execute the WRITE command in a YeS8401 MPD [data transmission multiplexor], a TRB-A signal--a request for line service--is fed to the computer using cycle T15 after addressing the given line. The computer channel responds to this request after 10-15 μsec while the next line is being addressed. The symbol from the channel lines is recorded by the input-output register of the MPD in this case.

This register is shared by 32 lines. Each line frame A and C in the MPD has its own input-output register. A symbol is cleared and rewritten in the R6 data register of the line making the request for service only during the matching addressing time and during the next addressing of the given line after 768 μsec (the addressing cycle time).

As long as the symbol is in the input-output register, the TZAV0 or TZAV1 flip-flop is set at position 1. In this case all other lines (even of another line frame) are barred from requesting service. Thus when 13 or more lines work simultaneously with a WRITE command at a speed of 1,200 baud, the probability arises that other lines of the MPD operating at the same speed with a WRITE command may go into overflow--that is, they may be unable to satisfy a request for servicing a symbol prior to receipt of the next. These 13 lines may "entrap" service for a period of 768 μsec×13=9,974 μsec, which is longer than the time it takes to receive one symbol at a speed of 1,200 baud--9,240 μsec. As the quantity of lines increases, the probability of such "entrapment" grows. When a remote processing system is operating with 16 lines using the algorithm of an AP-64 user station and 3 lines with Akkord-1200 or TAP-3 apparatus, data input into the computer by way of a YeS8401 MPD at a speed of 1,200 baud is impossible. Moreover input of almost each data bloc (256 bytes) ends in short-duration failure with an OVERFLOW updated status byte. This situation causes no trouble to lines operating with Akkord-1200 and TAP-3 apparatus because there is a feedback at junction S3. Thus overflow is excluded, but work of these lines with a WRITE command significantly raises the probability of failures in other lines.


ТЗЗ В5А38(0162 ЗАВ-02)
ТЗЗ В5А39(0149 НО-02)
ЗАП/НУЛ. ПД
ТГОС1*
Т4СО
ВИБД
R1...R11 1к
+5В
ВИ1К*(#В02РР08/5)
ВИ10*(#В02РР08/3)
ВИ11*(#В02РР08/1)
ВИ12*(#В02РР07/5)
ВИ13*(#В02РР07/3)
ВИ14*(#В02РР07/1)
ВИ15*(#В02РР06/4)
ВИ16*(#В02РР09/1)
ВИ17*(#В02РР09/3)
-ВИБД*(#В02РР05/6)
-ИССТОП*(#В02РР08/1)
Т6(В5А12=19=)
Т5(В5А12=17=)
Т14(В5А12=39=)
С1(В5А17=11=)
ТГИССТОП(В5А39=36=)
КВИ1К(В5А21=04=)
ВИ10(В5А21=09=)
ВИ11(В5А21=11=)
ВИ12(В5А21=21=)
ВИ13(В5А21=30=)
ВИ14(В5А21=32=)
ВИ15(В5А21=43=)(08=11=)
ВИ16(В5А20=33=)(08=12=)
ВИ17(В5А20=35=)
-ВИБД(В5А39=07=)
-ИССТОП(В5А39=34=)
АР17(В5А17=07=)
АЧ1(В5А37=43=)(17=10=)
АР16(В5А17=17=)
АЧ2(В5А37=46=)(17=20=)
АР15(В5А17=28=)
АЧ4(В5А37=44=)(17=25=)
АР14(В5А17=34=)
АЧ8(В5А37=41=)(17=39=)
АР13(В5А17=42=)
АЧ16(В5А37=39=)(17=45=)
Т114(В5А12=40=)
-Б20(В02РР01/2)
-Б20*(Б5А38=08=)
ТЗЗ В5А13(ПМ ВИР)

To exclude them, a buffer memory for 32 lines of the input-output register of the MPD of a YeS8401 was developed and introduced (see figure). This system is connected to an input-output register operating with moderate-speed lines. When lines are working with a WRITE command, after the demand for service is satisfied the symbols in the input-output register are immediately reentered into the buffer memory as required by their address. Then the condition ZEROING during transmission is run. This symbol is reentered into data register R6 at the next addressing step in this line. Thus the input-output register is freed to work with other lines without waiting for a matching addressing step. The interface sequence signal INTERFACE STOP is transmitted by way of the buffer memory in similar fashion.

After a demand for service is satisfied by a VI BD M signal, flip-flop 3Ye-1 ("Record/Zero Transmission") is set in position 1. In this case the contents of the input-output register (signals VI1K-VI17) are reentered into the buffer memory (elements 2A, 2B, 2V, 2G, 2D, 2Ye) by cycle T5 through element 1V-2 and using the address of the line which had made the demand determined by signals AR17-AR13.

Concurrently flip-flop is set in its unit position in cycle T5 by way of element V3-4 in the "Zeroing Operation" (NO-02) TEZ [not further identified].

Flip-flop 3Ye-1 is set in its zero position in cycle T6, the current value of the address counter (signals A41-A416) appears at the input of the buffer memory, and the memory operates in reading mode. If a symbol had been written in the memory, then at the next addressing step the symbol is reentered into data register R6 at the zero level of signal -VI BD. After this the contents of the buffer memory at this address are erased by cycle T-14 by way of element 1V-2.

After the interface sequence INTERFACE STOP is outputed, the "Interface Stop" flip-flop (its output signal TG IS STOP is shown in the diagram) is set in its unit position. Next, after zeroing of the request flip-flop 4 "Await Coincidence" (its output signal TG OS 1* is shown on the diagram) is flipped to its unit position. When these two flip-flops are in their unit position, flip-flop 3Ye-2 is set in its unit position by way of element 1Ye-2. In this case the interface sequence signal -IS STOP is transmitted similarly as with transmission of data through the buffer memory. The signal of the main circuit -VI BD and the signal -IS STOP are not used in this circuit: They are required for work with a technological TEZ.

The TEZ of the input-output register memory (PM VIR) is located in a free position in frame V 5A13. The circuit is connected to breaks in the circuits of signals -VI BD, -IS STOP, VI1K-VI17.

Before installation of the system is begun, we recommend manufacturing a technological PCB with bridges at contacts 10-27, 09-28, 06-29, 05-25, 23-33, 26-35, 19-37, 22-30, 44-31, 46-32, 34-39. After the technological PCB is installed and tuned, the input-output register may be used without further adjustment--that is, this system can be introduced in stages, thus excluding interruptions in the work of the MPD.

11004
CSO: 1863/221

SWISS CIRCUIT-BOARD PRODUCTION EQUIPMENT EXHIBITED IN MOSCOW

Moscow VECHERNYAYA MOSKVA in Russian 25 Mar 85 p 1

[Excerpt] An exhibition of equipment produced by Switzerland's "Afig" firm opened today.

This equipment is intended for manufacturing printed circuit boards, electronic components, and special fixtures used in electroplating.

A symposium on the topic "New Methods for Producing Printed Circuit Boards" will open here on March 27.

The show is being held with the assistance of the USSR Chamber of Commerce's All-Union Exposition Center Association.

FTD/SNAP
CSO: 1863/218

MINICOMPUTER FOR MONITORING ASSEMBLY OF CIRCUIT BOARDS

Moscow SOTSIALISTICHESKAYA INDUSTRIYA in Russian 14 May 85 p 1

[Excerpt] Vil'nyus--An innovation proposed by scientists of the Lithuanian Academy of Sciences' Institute of Semiconductor Physics progressed from a scientific concept to the stage of introduction into industrial production several times as quickly as usual. The first minicomputers that are used for monitoring the assembly of printed circuit boards were put into operation yesterday at the Shyaulyay Television Plant.

The implementation of a program called "Institute--Plant" helped speed the introduction of the scientists' development. The Vil'nyus physicists opened a design bureau at the plant in Shyaulyay, and the plant opened a minicomputer production section.

"Shyaulyay has become the first city in Lithuania to conclude an agreement on scientific-technical cooperation with our academy," reported Yu. Pozhela, president of the Lithuanian academy.

FTD/SNAP
CSO: 1863/329

UDC 681.325.5

PROCEDURES AND GENERAL METHOD FOR CHECKING MICROPROCESSOR MALFUNCTIONS ON THE BASIS OF PSEUDORANDOM TESTING SEQUENCES

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (revised manuscript received 31 Jul 84) pp 44-46

[Article by Doctor of Technical Sciences K. G. Samofalov and engineers Yu. S. Vilinskiy and S. K. Ravnyago]

[Text] The method for checking malfunctions in BIS [large integrated circuit] microprocessors is based on using a determined flow of instructions containing pseudorandom operands representing a testing sequence. This testing sequence is formed by means of a tester based on a built-in pseudorandom code generator, which checks the microprocessor at its working frequency. This approach excludes the laborious stage of writing a test program and eliminates the need for storing microprocessor input units and output reactions in large memories. Moreover use of pseudorandom testing sequences of considerable length (on the order of 10^5-10^6 units) makes it possible to widen the class of malfunctions that are checked in a number of cases, and to avoid fault modeling, since probability methods can be used to assess the completeness of the check.

Checking malfunctions by means of the proposed principle of generation of check stimuli [1,2] is significantly more effective if the algorithm for formation of the testing influences accounts for the structural features of the object being checked. In this connection we propose decomposing the functional flowchart of the microprocessor into a number of subsystems, so that we can verify each of them, and conveying data by internal buses of the microprocessor using special check procedures.

The general structure of a check procedure is shown in Figure 1. The check procedure determines the set of elementary programs to be used in checking individual microprocessor subsystems. In some procedures the phases of initial setting and of result read-out can be combined with the phase of the check operations. The number of times each procedure is repeated is established in accordance with the required completeness with which malfunctions are checked. Because the microprocessor is set in its initial state at the beginning of each procedure, in the general case the functional subsystems of the microprocessor perform the transformation $\{X\}\rightarrow\{Y\}$ during the check operations, where $\{X\}$ is the set of m-dimensional binary vectors, and $\{Y\}$ is the set of n-dimensional binary vectors.

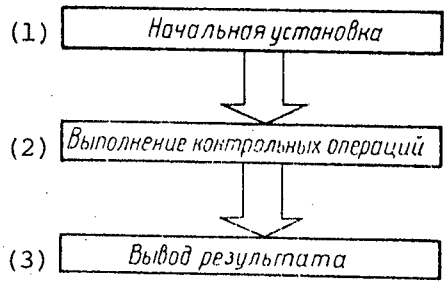


Figure 1. General Structure of the Check Procedure

Key:
1. Initial setting
2. Execution of check operations
3. Result read-out

Microprocessor subsystems can be divided into three basic classes:

subsystems which transfer data between memory registers (such subsystems perform the transformation $x_i \to y_i$, where x_i, y_i are the bits representing the initial and resultant machine words, $n=m$);

iterative circuits (for example of the accumulator and the decimal correction system);

general blocks (in particular, the operation code decoder). To make this method more understandable, its application is illustrated using the example of checking the widely used KR580 IK 80 microprocessor.

A random malfunction of the operation code decoder may be viewed as a failure in the performance of one or several microoperations, with failure associated with just one type of instructions being improbable. A minimal operation code decoder check test is synthesized using a table for cover of microprocessor instructions by microoperations [3].

The purpose of the initial check procedure is to check the subsystems that transfer and store data in general-purpose registers. This check procedure is carried out as follows (for reading convenience all check procedures are described at the level of KR580 IK 80 ASSEMBLER instructions and the computer codes corresponding to them:

ПК1 (1)	
MVI r	01 SSS 100
(2) <аргумент>	RRRRRRRR
MOV A, r	01111SSS
OUT	11010011
MOV r*, r	01S*S*S*SSS
MOV A, r*	01111S*S*S* (3)
OUT	11010011 (51 такт),

Key:
1. Check procedure 1
2. Argument
3. Cycles

where R--pseudorandom binary number; S, S*--bits formed by means of a counter modulo 8; r, r*--general-purpose register numbers. The number of computer cycles necessary to carry out a procedure is shown in parentheses at the end of the procedure. The result determination and read-out phases are separated in Check Procedure 1, and the states of outputs D0-D7 are the output reactions of the microprocessor undergoing testing.

To check a subsystem consisting of a program counter, address and data buses, and flag flip-flops, the flags, the branch address and the type of instructions are established in pseudorandom fashion. This is achieved by the following procedure:

ПК2 (1)	
RESET = 1	
MVI A	00111110
<байт операнда>(2)	RRRRRRRR
CMC	00111111
I <условие> (3)	11SSS010
< байты	11111111
адреса > (4)	11111111 (24 такта)(5).

Key:
1. Check procedure 2
2. Operand byte
3. Condition
4. Address bytes
5. Cycles

At the end of each cycle of this procedure, either a branch address or information on the state of the program counter is outputed on the address bus. Therefore the states of outputs A0-A15 at the end of the cycle would be the reactions of the microprocessor.

Checking an arithmetic-logic unit requires variation of arguments and instructions over a wide range, which is achieved by their following succession:

ПК3 (1)	
MVI A	00111110
(2) <аргумент>	RRRRRRRR
(3) <операция>	11SSS110
(4) <операнд>	RRRRRRRR
OUT	11010011 (24 такта)(5),

Key:
1. Check procedure 3
2. Argument
3. Operation
4. Operand
5. Cycles

where the instructions ADI, ACI, ANI, SUI, SBI, ORI, XRI, CPI are formed as the operations.

The circuits for decimal correction and formation of shifts in the accumulator are checked by the following check procedure:

ПК4 (1)	
MVI A	00111110
(2) <аргумент>	RRRRRRRR
ДАА	00100111
OUT	11010011 (24 такта(5));
ПК5 (3)	
MVI A	00111110
(2)<аргумент>	RRRRRRRR
<команда сдвига>(4)	000SS111
OUT	11010011 (24 такта(5)),

Key:
1. Check procedure 4
2. Argument
3. Check procedure 5
4. Shift instruction
5. Cycles

where RAL, RAR, RLC, RRC are formed as the shift instructions; in this case the link with the carry flag is checked during performance of the closed shift.

Check procedures 6-8 can be designed similarly to check the increment-decrement circuit, the stack pointer circuit and the indirect addressing mechanism circuit. At the same time these procedures, when used together with procedures 1-5 discussed above, provide for cover of microprocessor instructions by microoperations.

The minimum probabilities of malfunction detection are important indicators of the completeness with which malfunctions are checked in a microprocessor. Let us assume that a pseudorandom word is fed into the memory register, and that the probability that a unit would appear in each bit position equals p. Then the probabilities of detection $p_{\text{обн}}$ of malfunctions taking the form of unit and zero distortions in N read (write) cycles would be equal respectively to $1-(1-p)^N$ and $1-p^N$. In particular, when $N=20$, $p_{\text{обн}}=1-10^{-6}$.

It should be noted that malfunctions elicited by the mutual influence of neighboring bit positions in registers--so-called "nearest neighbor" malfunctions--are typical of circuits characterized by a high degree of integration. In this connection we need to determine the probability of recording all possible binary units in the presence of the given flow of pseudorandom operands into the given pair of neighboring bit positions of the register. Utilizing the classical probability of random distribution of marbles in a

set of pockets [4], we can calculate the probability of detecting malfunctions of this type using the formula

$$P_{\text{обн}} = \sum_{v=0}^{W} C_W^v (-1)^N \left(1 - \frac{v}{W}\right)^{N'},$$

where W--number of possible states of pairs of neighboring bit positions; N'--length of the testing sequence. In particular when $W=4$ and $N'=32$, we get $p_{\text{обн}}=0.997$.

The structure of an iterative circuit is represented in Figure 2. The following relationships are valid for such a circuit:

$$q_{i+1} = f_1(X_{i+1}, q_i); \; Y_{i+1} = f_2(X_{i+1}, q_i),$$

where f_1, f_2--Boolean functions; X_i, Y_i--correspondingly the input and output vectors of cell i of the iterative circuit, q_i--signal for transfer from cell i. Examining as an example a simple summator for which the following relationships are valid,

$$X_i = (x_{i1}, x_{i2}); \; y_i = x_{i1} \oplus x_{i2} \oplus q_i,$$
$$q_{i+1} = x_{i1}x_{i2}\bar{q}_i \vee x_{i1}\bar{x}_{i2}q_i \vee \bar{x}_{i1}x_{i2}q_i \vee \vee x_{i1}x_{i2}q_i,$$

it would not be difficult to show that when $p(x_{i1}=1)=p(x_{i2}=1)=0{,}5$, the probability of detection of an arbitrary constant malfunction that disturbs the function of one of the summator's bit positions is $p_{\text{обн}}=1-10^{-6}$ over a time period equal to 105 summation cycles. To calculate the probability of detection of a "nearest neighbor" malfunction, for example when using check procedure 3, we employ the asymptotic formula [4]

$$p_{\text{обн}} = e^{-\lambda}, \qquad (1)$$

where $\lambda = We^{-\frac{N'}{W}}$. In particular when $N'=520$ and $W=32$, $p_{\text{обн}}=1-10^{-6}$.

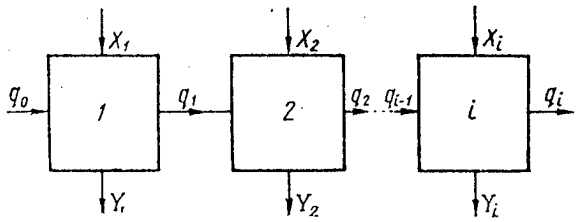


Figure 2. Structure of an Iterative Circuit

In contrast to the determined approach, where passage of all text units is guaranteed, in some cases when we use pseudorandom testing sequences, we need to estimate the length of the testing sequence that contains all possible input units, using the formula

$$N' = W'\left(\ln W' - \ln\ln\left(\frac{1}{p_n}\right)\right),$$

where W'--number of possible units, P_n--probability that each of the output units is contained in the testing sequence. Thus to check fulfillment of an instruction for a conditional branch on the basis of the parity index JPE, when p_n=1-10^{-6} and W=256, we get N'=4956.

The total time necessary to check a microprocessor can be obtained from the formula

$$T = \frac{1}{f}\sum_{i=1}^{k} n_i R_i t_i,$$

where f--operating frequency of the microprocessor, n_i--number of operations, each of which requires t_i computer cycles, in check procedure i, R_i--quantity of repetitions of check procedure i, k--number of microprocessor subsystems checked.

The microprocessor checking procedures are recorded in the tester's memory. Storage of each word of a check procedure requires 2 bytes in the memory--that is, the entire set of check procedures occupies less than 100 memory bytes.

The method proposed here makes it possible to detect malfunctions in microprocessors of the examined classes in not more than 1 second with a probability of 1-10^{-6} when the operating frequency of the microprocessor being checked is 2 MHz.

The Kiev Polytechnical Institute has developed a portable tester with which to check and diagnose malfunctions in digital blocks using the method proposed here. It is now being introduced in industry as a means for testing different blocks of microprocessors contained in control systems for machine tools equipped with digital program control, and for checking and restoring the function of local microcomputer networks. The method for checking malfunctions in microprocessors can also be used to test various devices containing microprocessor elements, and for microprocessor input control.

BIBLIOGRAPHY

1. Thevenod-Fosse, P. and David, R., "Random Testing of the Data Processing Section of a Microprocessor," FTCS-11, 1981, pp 275-280.

2. Berezov, B. P., "Some Features of Checking BIS Microprocessors," TR. INEUM, No 76, 1979, pp 29-32.

3. Bhujade, M. R. and Despande, S. R., "Microprocessor Testing Using Minimal Instruction Cover," IEEE TEST CONF., 1979, pp 182-186.

4. Feller, V., "Vvedeniye v teoriyu veroyatnostey i yeye prilozheniya" [Introduction to Probability Theory and Its Applications], Moscow, Mir, 1967, 493 pp.

11004
CSO: 1863/259

MICROELECTRONICS TECHNOLOGY

Moscow ZNANIYE-SILA in Russian No 4, Apr 85 pp 2-5

[Article by M. Sidorovyy, ZNANIYE-SILA correspondent, under the rubric: "Conversations on Technical Progress", based on a discussion with Cheslav Vasil'yevich Kopetskiy, corresponding member of the Academy of Sciences USSR and director of the Problems of Microelectronics and Ultrapure Materials Technology Institute, time and place not given]

> [Text] To develop at a leading pace the production of high-speed control and computational systems, peripheral equipment and software for these, electronic devices for control and telemechanics applications...
>
> Basic directions of economic and social development for the USSR during 1981-1985 and for the period until 1990.

The world of microelectronics created by man is one of fantastic possibilities and its horizons broaden daily. It is undergoing a process of interlocking growth and of introduction into our daily lives as well as into many fields of human endeavor.

Microelectronics is a catalyst for technical progress. It provides artificial intelligence for robots and systems of robots, for automated systems used in control, teaching, design and computing applications, spacecraft and production facilities. How did this amazing world of microelectronics start? What is the nature of its "products"?

Those who say that a modern state's power is not based on bars of gold or deposits of precious gems stored in bank vaults are right. A country's economic might and defensive capability are defined by the scale at which it produces pure and ultrapure substances and materials, including semiconductor and metallic crystals.

In the beginning there was silicon

The family of semiconductor devices is expanding year by year and the semiconductor family tree is growing apace. Nevertheless, ultrapure silicon will remain the basic material, at least until the end of the 20th century. The technology for growing and developing semiconductor "embryos" is consistent, continuous and harmonious.

First of all the raw crystals are cut into thin plates. Polishing and etching bring their surfaces to an ideal smoothness and purity. The diamond saws jangle, the polishing disks spin and the etching machines work without a sound. The cut crystal disks are washed in a cupel with extraordinarily pure water and then placed in electric ovens (either vacuum units or ones filled with hydrogen or argon). Here the most important operations--alloying, introducing necessary additives by diffusion, polishing crystal surface layers with these substances, metallization and the application of layers--take place at high temperature. Next comes assembly, a clever and intricate operation. Thermocompression welding ("the golden seal") is used to attach electrode conductors securely to the edges of the additive layer extending from the crystal's surface. A cap is added to the finished chip and sealing quality is tested. Caps with signs of leakage are cold-welded in a vacuum. Now all that remains is careful testing of the solid-state "embryo's"--the semiconductor device's--electronic characteristics.

Semiconductor mass production was established at the end of the 1950's and these devices became the basis for many new instruments, apparatuses and machines. This technology retains its significance to this date, as evidenced by the number of these devices hidden in our radio receivers and television sets.

So, it is a well established technology which will continue to serve us for some time producing semiconductors which have found and will be finding wide-scale application.

However, the problem of further device miniaturization and data processing speeds increases has arisen. The semiconductors themselves have become obstacles in this process. Semiconductor-based electronic devices have finally reached the dimensional limits of traditional (insertion or displacement) assembly. Assembly became so complex that it turned into an art which could be performed only by virtuosos in an operation not suitable for mechanization or automation. Labor productivity and quality fell sharply. Hand assembly of electronic circuits had to be replaced. But how and with what?

Thus arose the concept of a new technology: locating many components with different purposes on a single crystal, forming these elements by means of a certain distribution of additives connected by the thinnest of conductors arranged on the chip surface. The new addition to the semiconductor family, called the IC (integrated circuit), was developed by drawing on a surface--in this case the surface was that of the crystal itself.

Integral, or solid microelectronic circuits are not assembled. They are based on film, electronic lithography and ion technologies and on putting nearly

every new development in the natural sciences to work in the realm of microelectronics--including ultrapure substances, ion accelerators, low temperatures, laser beams and deep vacuums.

The integrated circuit gradually became so complex that it could independently perform logic operations on data signal streams--this algorithm required the execution of data "recognition" operations. Subsequent evolution of microelectronic components moved primarily in the direction of complicating and converting their functions while previous evolutionary trends were confined to changing electronic device production techniques and operational principles.

The genealogy of the "smart" chips, into which thousands of electronic components invisible to the naked eye are installed without human hands, is quite simple. They are classified by design complexity and of course by data processing "capacity" as integrated circuits--IC (with tens of components), medium-scale--MSIC (with thousands of components), large-scale--LSIC (with tens of thousands of components), and very large-scale integrated circuits--VLSIC (with hundreds of thousands of components).

Each level of IC complexity produced a new generation of electronic computers with new, more powerful memory operating at speeds not attainable by earlier computers. This came hand in hand with new degrees of microminiaturization. VLSICs are at the heart of personal computers which, instead of occupying rooms or massive cabinets, are the size of an encyclopedia and contain an incomparably larger volume of information. These are the single-chip computers.

Finished ICs in their miniature cases look like toys about the size of a kopeck or insects with their multiple contact pins. Take a comfortable handful of them and you will find it difficult to believe at first that you are holding the power of dozens of computers in your hand.

The internal construction of VLSICs is incredibly complex. It is hard to find an analogous example of such a system with millions of component operating in strict, well ordered interaction. Of course VLSICs cannot be designed without computers--no new developments in electronics can take place without the use of electronics itself. Maximum automation is required here. Programs had to be developed to allow computers to synthesize nearly a hundred different types of digital circuits in minutes or to plot the design of a thousand components, locate them in a given area and trace connection paths for them. Man is left with the creative part of this work: thinking out the structure and selecting optimum designs.

Purity--strength and weakness

Fantastic levels of crystal purity are required for the manufacture of integrated circuits. These in turn require sealed buildings, shops, laboratories, benches and equipment. This struggle for "semiconductor purity" (a level incomparable with operational purity) is neither cheap nor easy.

It appears that in the future super-complex integrated circuits will be synthesized on a monocrystal sheet in small air-tight cells using physicochemical technology.

Integrated circuits are based on the now familiar silicon sheet produced in a chemical/metallurgical plant. It is round, thin, silvery in color and weighs nearly nothing. Thanks, however, to its solid crystalline structure it can withstand the subsequent, multistep process of forming elaborate microelectronic circuits on its surface layer. The tools for this are ultraviolet light, electron beams, x-rays and laser beams acting as the "hands" of computers controlling the microelectronic production process. It is a significant phenomenon: electronics producing electronics in its own image.

A series of production processes starts the development of a dielectric oxide layer (film), composed of silicon dioxide, on the silicon sheet (substrate). Silicon dioxide is a nearly perfect insulator and forms a strong covering whose thickness is precisely controlled. The oxide layer protects the substrate from undesired additives.

This process takes place in a quartz tube, placed together with the semiconductor sheets in a high-temperature oven. Oxygen is blown into the tube to form the oxide layer on the sheets. Next the substrate is placed in a centrifuge and rotated. Photosensitive lacquer is added from above and it spreads to form a thin film over the oxide layer. This is the photolayer. Next it is covered with a glass template (mask) covered with a pattern of lines. One of the levels of the future IC is formed by light transmitted or rejected by various sections of the mask. Each level of the IC has its own mask.

The next step consists of photolithography. Light (including ultraviolet light) is shined on the mask-covered sheet. The photolayer undergoes a chemical transformation and is developed. Exposed portions are removed and washed. Analogous portions are removed by etching in oxygen.

Now a metered amount of additive elements is introduced at certain sites--"windows" (only where the oxide is etched)--by diffusion alloying. Usually alloying is effected while heating the sheet in phosphorus or boron vapors at a temperature of 1100-1200 degrees.

By repeating this process and forming several layers one after the other a conductor chip is created with infinitesimal transistor and diode structures distributed in an area significantly smaller than the diameter of a human hair.

Conditions demand further microminiaturization of components and integrated circuits themselves. Photolithography of IC layers is an obstacle in this process as it has reached its limits of application. It is not capable of forming circuit runs with thicknesses on the order of a thousandth of a millimeter. Light rays are beginning to give way to electron and x-ray beams. With electron beams, for example, IC runs only hundreds of times the size of an atom can be drawn.

Semiconductors from gas and vapor

Hundreds of techniques based on different operational principles and sequences have been developed. Production of the largest circuits requires not less than 10-12 groups of operations aimed at translating a circuit diagram into the surface structure of a crystal sheet.

In recent years molecular beam epitaxy has become one of the most basic methods in the production of semiconductor films. It is a remarkable example of a successful symbiosis of scientific and technological achievements, specifically the fields of solid-state physics and supervacuum technology.

Three fundamental methods are used to develop epitaxial semiconductor layers: chemical precipitation from the vapor phase, crystallization from liquid and evaporation in a vacuum. The latter, molecular beam epitaxy, is the most modern. A portion of the silicon slab (or other substance) is heated to the melting point by an electron beam. The drop only touches solid silicon and this maintains the purity of its vapors. Next the work area--an apparatus which also contains the substrate--is brought to a super-high vacuum. This assures the purity of the process. Silicon vapors settle uniformly on the substrate as atoms.

Their number is unlimited. Therefore, molecular beam epitaxy allows the formation of as many complex connections and heterostructures as desired. This method was used to produce the first, extremely promising gallium arsenide semiconductor material, used in laboratories to produce integrated circuits with switching times on the order of 10-11 second.

As we have discussed, solid-state structures are synthesized in a series of operations which apply two-dimensional diagram programs on a layer-by-layer basis. This is a basic principle in microelectronic technology. It is a constant and is changed or improved only in terms of the physical and chemical methods used in its implementation.

Here is one of the new physicochemical methods. The sheet of silicon or other semiconductor crystal material is bombarded with atoms (ions) from "beneficial" additives. The ions penetrate the semiconductor's lattice and change its electrical properties at the penetration sites.

Electron/ion facilities consisting of electron and ion accelerators, a high-resolution chamber and computer control equipment complete this process of "engraving" circuits on the crystal's body. The thin, machine-controlled ion beam forms the circuitry by applying systems of atomic films, with a thickness relative to the length of the light wave, at a given depth in the crystal's surface layer. Formed from microscopic zones with the use of additives, these circuits perform well as diodes, triodes, capacitors and resistors. Next the electron beam etches, fuses, vaporizes and separates the chemical connections.

The problem of maximum purity in microelectronic "construction materials" is solved with surprising elegance in this case. A magnet with a precisely determined field located at the curvature of the accelerator chamber forces the ion beam to bend, completing a turn. Only ions of a strictly defined mass

(a single element) can overcome this force, other ions pass by the target.

The production processes are executed automatically on commands from the control computer. In a short space of time the electron/ion facility can produce tens of silicon chips each containing tens of thousands of microelectronic circuits. Thus, once tools in the search for knowledge about the world, accelerators have become a means for changing it.

Necessary "contamination"

Future microminiaturization of all types of integrated circuits depends to a great extent on obtaining even purer metals and semiconductors, as well as deeper vacuums to provide conditions for the execution of the production processes needed to form circuits. Vacuum technology, including cryotechnics (freezing and subsequent isolation of gases), provides the most complete means of keeping gaseous substances, dust and microorganisms out of the work zone.

Year by year industry places stricter requirements for purity and perfection in the structure of silicon crystals. The use of vibration-free motors and pumps and noiseless transformers in production facilities allows an improvement in product quality. New or improved energy-saving equipment is being developed (silicon production consumes a great deal of power).

Together with the demand for ever greater purity, the requirements for "contaminants" (i.e. the alloying additives) are becoming stricter. Silicons with extremely high homogeneity are needed for microalloying with beneficial additives. A new alloying technique--carried out in a neutron flux--is now in use.

As long as 20 years ago it was known that when exposed to thermal (slow) neutrons the silicon isotope Si^{30} (found at levels barely in excess of 3 percent in normal silicon) is transformed into a phosphor donor, the most important element in alloying semiconductor silicon. This effect is the basis of a method developed in this country for neutron alloying of silicon monocrystals for semiconductor devices.

The authors of this invention--scientists from the Physical Chemistry SRI imeni L. Ya. Karpov and several other organizations--prepared an experimental batch of neutron-alloyed silicon in this country's nuclear plants. The organization of production of large batches of silicon semiconductors according to the new technique was planned at the Chernobyl AES. This material will be used to produce large power rectifiers--thyristors--capable of operation with several thousand volts, television imaging equipment, photodetectors and, primarily, microelectronic semiconductor devices and integrated circuits.

How to measure the nearly unmeasurable

For some time, the Achilles' heel of microelectronics technology was the determination of device quality and characteristics. A prime concern was the development of methods for non-destructive monitoring, the capacity to measure a device's principal characteristics without damaging its structure or

altering its physical and chemical properties. One example of a good "gauge" is the electron beam. Each atom of the alloy is visible under an electron microscope. One can thus observe microcircuit behavior under operational conditions--with power applied--and evaluate a device's suitability and reliability.

X-ray analysis and various types of spectroscopy are being used with increasing frequency for layer-by-layer examination of semiconductor materials in the vicinity of individual points, either on the surface or deep within the specimen. None of these methods damages the specimen, therefore they are used to monitor the correctness of production processes. As an example, spark mass spectrometry is used for local (especially layer-by-layer) analysis of epitaxial silicon and germanium films at a resolution level of up to one-tenth of a micron.

Electron and proton spectroscopy allows the analysis of surface layers only a few angstroms thick.

New methods are being developed for microelectronic device quality control monitoring during production. A group of specialists and scientists from the Crystallography Institute of the Academy of Sciences USSR and other organizations created and brought to production environment refinement a method for monitoring the thin-layer structure of microelectronic devices. The quality of these thin layers is the primary determinant of a device's operational reliability. This new method of monitoring integrated microcircuit film structures during the production process permits timely evaluation of device suitability and timely avoidance of wastage. Widespread applications of these types of methods will allow an increase in the production of ICs of sufficiently high quality and well in excess of the level reached abroad.

Still, in many cases it is significantly easier to achieve purity than to prove that the result is in accordance with the planned target. Thus, researchers need to solve the problem of analyzing small amounts of additives and evaluate the level of purity obtained.

A range of methods are used for this analysis. X-ray fluorescence spectroscopy "senses" and determines additive content to 10^{-4} percent, polarography to 10^{-7} percent and the spectral chemical method operates to 10^{-8} percent. The neutron activation method discovers additives at a level of 10^{-10} percent of the total ultrapure substance mass. The units used for this analysis are equipped with semiconductor detectors, multichannel gamma gauges and microcomputers.

The laser fluorescence method developed in this country at the Spectroscopy Institute of the Academy of Sciences USSR even permits the discovery of individual atoms in additives. This method is currently being perfected. Laser spectroscopy promises the possibility of isolating fantastically small quantities--tens of foreign atoms per cubic centimeter of a substance.

By choosing an appropriate laser radiation wavelength and intensity, ionization can be confined to a single type of atom (that of the additive) and the ions thus formed are relatively easy to count. This research on selective

laser analysis was carried out at the Spectroscopy Institute of the Academy of Sciences USSR and the Physics Institute imeni P. N. Lebedev of Moscow State University.

On the horizon: the photon computer

Researchers are now attempting to solve the problem of placing an increasing number of microelectronic components with large functional capabilities in a decreasing chip volume. One possibility for such a "volumetric" three-dimensional device is a layered structure with one layer making up the microprocessor (arithmetic-logic unit), another layer consisting of memory, etc. Even the connections between layers can be made to perform beneficial functions. Those working on three-dimensional VLSICs are faced with two main problems: developing a suitable production technology and dissipating the heat these devices generate when working.

Research extremely important to solid-state physics is underway on the surface properties of ultrapure metals, substances and films.

"Worth its weight in gold" is an expression which we use to define many material and spiritual values. This expression loses its meaning when seen from the point of view of surface physics, the newest field of science. In fact, only gold's thin upper layer "works." It generates a decorative attraction and has great anticorrosive properties. For this reason, gold should be valued by square centimeter of surface area rather than by the gram or carat. But of course this is the opinion of scientists--surface physicists--for whom gold is only an object of research; jewelers would hardly agree.

Currently, surface physics is enjoying a true boom. The Academy of Sciences has even produced a new scientific journal, entitled "Poverkhnost': fizika, khimiya i mekhanika" [The Surface: Physics, Chemistry and Mechanics]. The journal's chief editor is Academician Ye. P. Velikhov, a vice president of the Academy of Sciences USSR.

The development of microelectronics technology allows the hope that, by the year 2000, the minimum dimension of IC components will be reduced to one quarter of a micron, allowing one billion transistors to be located in a single microcircuit (this is a thousand times the current level).

The use of gallium arsenide semiconductors is promising in terms of increasing computer speeds. Devices of this type have switching times near 10^{-11} second. In theory there are a tremendous number of applications for VLSICs containing components made of ultrapure gallium arsenide and indium phosphide. Computers based on these VLSICs could apparently execute nearly 20 billion operations per second.

As before, the problem of improving semiconductor device operational characteristics is a critical one. Several approaches for resolving this complex task have been planned and are being implemented.

One of these is the formation of heterojunctions in semiconductor systems, essentially this is the joining in the monocrystal of two semiconductors which

have different chemical compositions but which are compatible in terms of their physical and other properties.

Research has confirmed that, in principle, a new means of controlling electron flow can be established at a heterojunction and that these elements can be placed in certain portions of a chip. This has even served as the basis for the creation at the Academy of Sciences USSR's Leningrad Physical Technical Institute of a new generation of semiconductors for application in various branches of science and industry, including optoelectronics--the basis for a future photon computer with amazing operating speeds, memory capacity and "quickness".

Improvements in integrated circuits mean improvements in all areas of electronic computer technology. Artificial crystals with a structure that is unimaginably complex, yet tightly organized at all levels (including the molecular and even atomic levels), will allow the intensification of many processes in industrial devices, in analytical equipment and in control systems.

Here are just a few examples.

In the chemical industry the time required for chemical reactions is usually many times less than the time required for process and substance analysis. Now an inverse relationship can be considered. This opens up fantastic possibilities for recording all of a substance's intermediate states and for developing "superoptimum" process control methods.

Microelectronics installed in motor vehicles will allow continuous monitoring and control of vehicle direction, speed and braking. Control optimization will allow a minimum fuel savings of 10 percent. When we imagine the fleet of vehicles in this country we are convinced that this savings can be achieved just through the introduction of semiconductor technology.

The development of flexible automated production and the introduction of robots in this technological environment are being widely discussed at this time, with the principal objective being machine construction. The introduction of microelectronics at all level of production--from machine tool control to divisional management--will allow the establishment of flexible automated systems even in metallurgy, and possibly in other types of production facilities as well. As a result there will be a fundamental change in man's entire technical environment.

12746
CSO: 1863/297

UDC 681.322.01

ORGANIZATION OF PROCESSOR ORIENTED TO MULTIPROCESSING

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 12, Dec 84 (manuscript received 7 May 84), pp 23-29

[Article by V.F. Guzik, V.E. Zolotovskiy, and V.N. Reshetnyak, Taganrog Radio Engineering Institute imeni V.D. Kalmykov: "Organization of Processor Oriented to Multiprocessing"]

[Text] This article proposes the structure of functionally distributed processor, permitting organization of source program multiprocessing. The foundation of this structure is the concept of basic functional processor elements in the form of multiservice units. The approach presented here makes it possible to reduce time expenditures for blocking computational processes and to increase processor speed.

The problem of increasing the capacity of modern computer systems is most urgent. Increasing the number of processors is not always expedient and justified. As evidenced in reference [1], the overall speed of system Q_s, here, can be defined by the expression

$$Q_s = Q_{pr}\log_2 M,$$

where M = number of processors in the system; Q_{pr} = speed of one processor. On the other hand, reserves in increasing Q_{pr} through increasing the speed of electronic processor components are, for all practical purposes, now exhausted.

Therefore, the outlook for increasing Q_{pr} (and subsequently Q_s) must be linked to architectural solutions, among which the following must be mentioned first of all.

1. Functional specialization of individual processor units and (or) a certain number of computer system processors to execute frequently encountered operations (task fragments).

2. Three-dimensional distribution of the computational process among different processor units -- multiprocessing.

3. Asynchronization of stages of the computational process.

Modern integrated technology supports such an approach, making the capability to build a processor based on LSI [large-scale integrated] circuits available to the developer. Here, the question concerns construction of a functionally distributed processor with command instrumentation.

Let us examine the general structure of such a processor (Figure 1). The following basic functional units can be distinguished.

1. ALB - arithmetic and logic unit [ALU], executing adding, computing, and logic operations.

2. BU - multiplication unit [MU].

3. BD - division unit [DU].

4. ARS - arithmetic expander [AE], executing conversion, binary, etc. operations.

5. BEF - elementary functions unit [EFU] (sin, cos, etc.).

6. OPP - program main storage [PMS].

7. OPD - data main storage [DMS].

8. BS - semaphore unit [SU], managing process interactions.

9. DISPETCHER - central control unit.

To more fully load each unit and reduce overall performance time, it is expedient to represent the program, handled by the processor, as a set of parallel arms. In this case, the processor with the structure described makes it possible to manage multiprocessing.

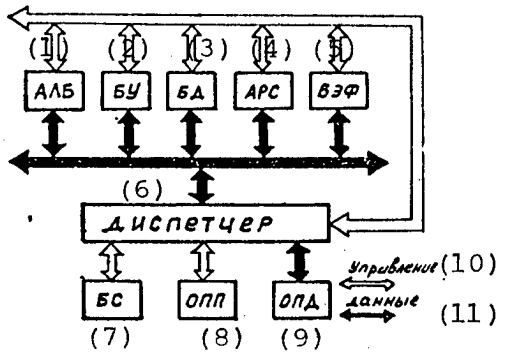


Figure 1. General structure of processor

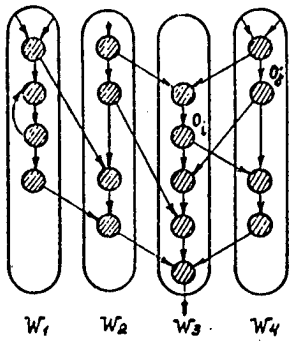


Figure 2. Representation of source program

Key:

1 - ALU	7 - SU
2 - MU	8 - PMS
3 - DU	9 - DMS
4 - AE	10 - Control
5 - EFU	11 - Data
6 - DISPATCHER	

Let us analyze the features of organization of such a computational process. Figure 2 shows a sample representation of program W as a set of parallel arms $W_1 \ldots W_4$ (filled-in circle = specific operation; arrow = linking of operations by data and logic). Here, the source computational process produced by the sequential program W is broken down into the set of processes produced by arms $W_1 \ldots W_4$, and interconnected at individual points in time. The following types of blocking, which reduce the effectiveness of organizing multiprocessing, are inherent in the emerging subprocesses.

1. Blocking of the First Type. Subprocess W_i requested and monopolized a certain resource R of the processor. The latter was also required by W_j, but in connection with the busy resource, the subprocess is blocked and shifts to the holding mode. If the operation implemented by resource R is encountered frequently enough in each arm, subprocess waiting time will be significant.

2. Blocking of the Second Type. Subprocess W_3 is blocked after execution of operation O_i if, by this point in time, subprocess W_4 has not prepared the result of operation O_j (Figure 2).

Analysis of the nature of these types of blocking makes it possible to draw the conclusion that the second type of blocking is internally inherent in the given approach, and the first type of blocking can be eliminated or considerably "moderated" if processor resources (functional units) are organized as multiservice units. The multiservice unit (MSU) represents a multichannel queueing system (MQS) with a common queue at the input. Each MSU channel (MSUC) is made as one or several LSI circuits, executing the operation of a given processor resource (multiplication, division, addition, etc.). Subprocess W_i obtains access to processor resource R by generating QUERYs, consisting of the following fields.

1. NB - number of branches; determines the applicability to the specific subprocess of the initiated operation.

2. IOC - initiated operation code.

3. OP1, OP2 - first and second operations, respectively.

4. RA - result address.

The result address and number of branches are attributes of the initiated operation which are input for unambiguous identification of the RESULT obtained. Field of the latter: NB, RA, and R, where R is the result of the

completed operation. The QUERY arriving at the MSU input is put in a queue, which must have a corresponding capacity to eliminate QUERY loss. Since only operands and IOCs directly participate in the initiated operation, a buffer should be introduced into the MSU structure to store attributes of those operands already selected from the queue and loaded into the corresponding MSUC.

Figure 3 shows the proposed MSU structure, containing the following components.

1. QUEUE - multiple buffer for receiving operands and attributes of the initiated operation.

2. GPR - general purpose register for operand attribute storage.

3. $MSUC_i$ - multiservice unit channel.

4. ARBITRATOR - unit for sending QUERYs and controlling RESULT unloading from the MSU.

5. PORT - unit for collecting the result of the executed operation with its attributes and transmitting the RESULT from the MSU.

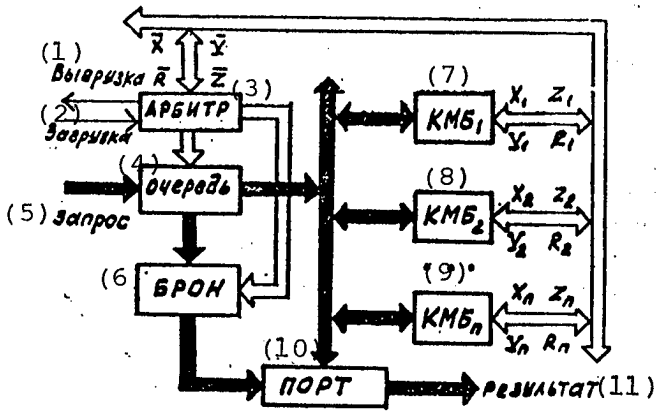


Figure 3. MSU structure

Key:

1. Unloading
2. Loading
3. ARBITRATOR
4. QUEUE
5. QUERY
6. GPR
7. $MSUC_1$
8. $MSUC_2$
9. $MSUC_n$
10. PORT
11. RESULT

Such a structure makes it possible to receive QUERYs asynchronously. A QUERY processed by a certain processor is immediately put in the QUEUE without preliminary analysis of the MSUC busy-state and condition. Operations in the MSUC are executed in parallel and independently. At each point in time the MSU can handle up to "n" different subprocesses, which makes it possible to reduce time expenditures for the first type of blocking.

The process of loading the QUERY is performed on the "LOAD" signal, handled by the processor's DISPATCHER. Here, the IOC determines the load point (MSU number). The ARBITRATOR develops the report signal "UNLOAD" for the DISPATCHER during preparation of the result of the operation if only by one MSUC, and permits unloading of the RESULT via the MSU PORT. Discipline of QUERY access is from the FIFO QUEUE. The MSUC forms vectors of condition $\overline{Z} = Z_1 \ldots Z_n$, $\overline{R} = R_1 \ldots R_n$ for the ARBITRATOR.

$$Z_i = \begin{cases} 1, \text{ if } MSUC_i \text{ is free,} \\ 0, \text{ otherwise.} \end{cases} \qquad R_i = \begin{cases} 1, \text{ if } MSUC_i \text{ prepared} \\ \text{results of the operation} \\ 0, \text{ otherwise.} \end{cases}$$

Modern LSI circuits, as a rule, make it possible to form the specified vectors. Based on $\overline{Z}$ and $\overline{R}$, the ARBITRATOR forms the load vector $\overline{Y} = Y_1 \ldots Y_n$ and unload vector $\overline{X} = X_1 \ldots X_n$.

$$Y_i = \begin{cases} 1, \text{ if } MSUC_i \text{ must be loaded,} \\ 0, \text{ otherwise.} \end{cases} \qquad X_i = \begin{cases} 1, \text{ if } MSUC_i \text{ must be un-} \\ \text{loaded,} \\ 0, \text{ otherwise.} \end{cases}$$

The load (unload) discipline is cyclical. If MSUC state vectors contain several units, then MSUC load (unload) is done in increasing order of their numbers.

During QUERY load into the $MSUC_i$, the QUERY is divided into two fields: operands (loaded into the $MSUC_i$) and attributes (loaded according to the i-th GPR address). The GPR address defines the ARBITRATOR directly from $\overline{Y}$ processing. During unloading, the ARBITRATOR processes $\overline{X}$ and the nubmer of the unloaded MSUC, which goes to the GPR as an address. The result of the operation and its attributes go to the PORT for further transmission from the MSU to the DISPATCHER.

It is likely that the proposed processor organization should increase its speed. Let us try, however, to estimate how the mean QUERY wait time will change during MSU organization, as compared to a single functional unit. Let us draw upon queueing theory for this.

Figure 4 shows an MSU model with n channels, which must handle arrivals of requests from m parallel branches ($n < m$). Each branch W_i generates a request flow to execute a given type of operation at an average rate of $\lambda = N/mT$, where N = number of requests for operation execution over the performance time of task W, T = total time to perform task W.

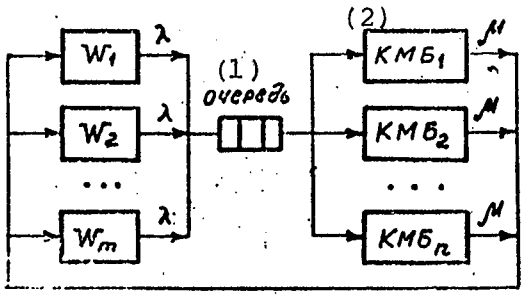


Figure 4. MSU model

Key:
1. QUEUE
2. MSU_1, MSU_2, MSU_n

Each MSUC handles incoming requests at an average rate of $\mu = 1/t_{op}$, where t_{op} = mean time for executing a given type of operation in the MSUC.

A closed model is used, since the queued request cannot be processed by the branch as long as the result of handling its previous request is not received. The maximum number of places in the queue when all m branches requested the resources of a given MSU can be estimated. Here, n requests will be in operation in the MSU and (m - n) will be in the queue. The random process occurring in the MQS under examination is Markovian. Such an assumption is, for all practical purposes, acceptable since the Poisson arrivals of requests creates the most difficult operating conditions for the MQS [2]. In the case of a difference in actual arrivals (input and handling), the characteristics limiting MQS functioning can be obtained from the Poisson arrivals.

The MQS being examined can be characterized by the following conditions: S_0 = all MSUCs available; S_1 = one MSUC busy with request handling, the remaining MSUCs available; ...S_n = all n MSUCs busy with handling requests; ...S_{n+1} = (n + 1) requests entered the MSU, of which n are handled in the MSUC and i requests remain in the queue; ...S_m = all m requests admitted to the MSU, of which n are handled in the MSUC and (m - n) remain in the queue.

Figure 5 shows a graph of MQS conditions and represents a classic system of destruction and reproduction. Let us designate the probability of condition S_i by P_i. If the set of Kolmogorov's equations [3] is formulated for this MQS and solved for the established regime, then the probability of system conditions is determined as follows:

$$P_0 = \left[1 + \sum_{i=1}^{n} \frac{m!}{i!\,(m-i)!}\rho^i + \sum_{i=n+1}^{m} \frac{m!}{(m-i)!\,n!\,n^{i-n}}\rho^i\right]^{-1},$$

$$P_e = \frac{m!}{e!\,(m-e)!}\rho^e P_0, \quad (e \leqslant n),$$

$$P_{n+i} = \frac{m!}{n!\,n^i\,(m-n-i)!}\rho^{n+i}P_0, \quad \text{where } \rho = \lambda/\mu.$$

Let us designate r the average number of requests arriving at the MQS, and l the average number of requests in the queue. Then, in the active state (response not expected) there will be an average of (m - r) program branches, and the overall arrival of requests produced by them will have a density of $(m - r)\lambda$. For the established MQS functional mode, the equation

$$(m - r)\lambda = k\mu$$

is true, where $k = P_1 + 2P_2 + \ldots + (n - 1)P_{n-1} + n(1 - P_0 - P_1 - \ldots - P_{n-1})$ is the average number of MSUCs directly occupied with handling requests.

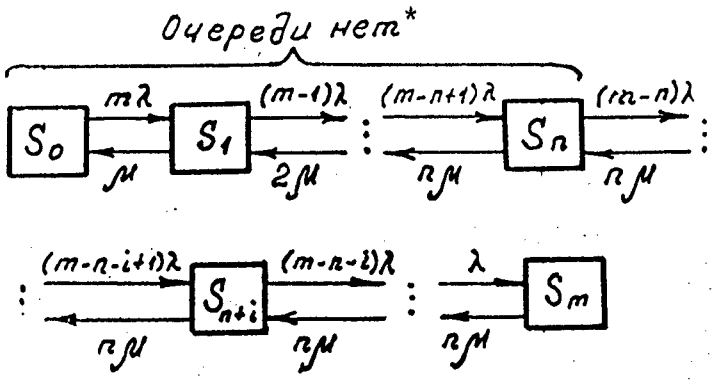


Figure 5. Graph of MSU model conditions

* No QUEUE

Considering the relationship r = 1 + k, from the equation presented [above] we obtain

$$l = \frac{m\rho - k(1 + \rho)}{\rho}. \qquad (1)$$

Then, the average waiting time t_{wait} of requests in the queue can be determined as follows:

$$t_{wait} = \frac{l}{(m - r)\lambda} = \frac{m\rho - k(1 + \rho)}{\lambda k}. \qquad (2)$$

Using the same reasoning we determine l and t_{wait} for the single functional unit,

$$l = \frac{m\rho - (1 - P_0)(1 + \rho)}{\rho}, \quad (3)$$

$$t_{wait} = \frac{m\rho - (1 - P_0)(1 + \rho)}{\lambda(1 - P_0)}, \quad (4)$$

where $P_0=[1+m\rho+m(m-1)\rho^2+\ldots+m!\rho^m]^{-1}$. The expressions (1) - (4) obtained make it possible to estimate the efficiency of MSU organization compared to a single functional unit. The table shows some results of such estimates.

m	n	ρ	λ, 1/c	l	$t_{ом}$, мкс *
10	1	0,5	10^6	7	3,5
10	4	0,5	10^6	0,55	0,09
10	1	0,8	10^6	7,75	6,2
10	4	0,8	10^6	1,67	0,36

* t_{wait}, μs

Analysis down to the basic circuit arrangements of a processor with the proposed organization (for m = 16, n = 5) showed its practical expediency in equipment costs. However, it should be noted that in MSU construction it is not recommended to take n > 10, since, in this case, expenditures for subprocesses grow.

Processors whose principles of construction are presented in this article can be used successfully in computer systems for modeling and real-time control.

BIBLIOGRAPHY

1. Wirschig, J.E. "Computer of the 1980's. Is it Network of Microcomputers" [sic], EXPRESS INFORMATSIYA. VYCHISLITEL'NAYA TEKHNIKA, 1976, No 13, pp 14-16.

2. "Osnovy teorii vychislitel'nykh sistem" [Fundamentals of Computer Network Theory], Ed.: S.A. Mayorov, Moscow, Vysshaya Shkola, 1978, 408 pp.

3. Venttsel', Ye.S. "Issledovaniye operatsiy" [Operations Research], Moscow, Sovetskoye Radio, 1972, 552 pp.

12304

CSO: 1863/172

UDC 681.32.06

MODELING PROBLEM-ORIENTED MULTIPROCESSOR SYSTEMS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (manuscript received 11 Nov 83) pp 37-39

[Article by engineer V. A. Smol'nikov]

[Text] The problem orientation of computer systems is associated as a rule with expansion and complication of the basic functions of general-purpose computers. The architectural features of problem-oriented computer complexes must maximally reflect the unique features of the algorithms of the class of tasks carried out by the given complex, and insure satisfaction of certain requirements imposed on its use [1].

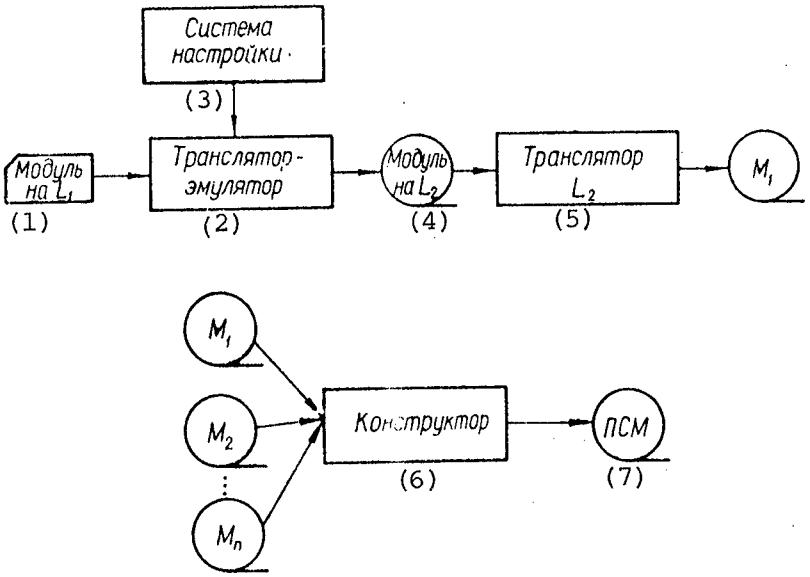
Typical of this class of problem-oriented computer complexes are the following features: use of multiprocessors; presence of a wide assortment of nonstandard input-output devices; orientation of the structure of the problem-oriented computer complex on processing homogeneous information and on a pipeline system of organizing calculations both at the micro- and at the macrolevel; presence of the necessary hardware for multiprogramming; the possibility of using the principal processor time for complex data processing with minimum system outlays but with sufficiently complex and diverse software components; presence of standard software developed specially for the given problem-oriented computer complex for the purposes of achieving the greatest effectiveness of the complex's operation.

There are three basic levels of modeling: systemic, at the level of register transmissions, and circuit. These levels of modeling are used in different stages of designing a problem-oriented computer complex [2]. Modeling at the instruction level, which makes it possible to combine the first two levels of modeling, is examined below.

During the systemic and logical stages of designing a problem-oriented computer complex it is important to define different versions of architectural decisions for the problem-oriented computer complex with the required speed, economy, reliability and other characteristics.

It is for this purpose that the GENMOD system was developed. It permits generation of different versions of program modeling systems and analysis and debugging subsystems depending on the user's requirements [3], to verify the

logical correctness of the proposed hardware at the level of instruction fulexecution and to evaluate the operational qualities of this hardware. Descending and ascending design methods are used in this case. The flowchart showing how the GENMOD system functions can be seen below.



Flowchart Showing How the GENMOD System Functions: $M_1,\ldots,M_n$--working modules

Key:

1. L_1 module
2. Translator-emulator
3. Tuning system
4. L_2 module
5. L_2 translator
6. Designer
7. Modeling program system

The problem of translating the description language L_1 into one of the high-level languages $\{L_{2i}\}$ was solved by using the machinery of the theory of formal grammar and substitution theory:

$$\varphi_i : L_1 \longrightarrow \{L_{2i}\}.$$

When modeling is conducted at the instruction level, the architectural features of a proposed problem-oriented computer complex may be described with varying degrees of detail; moreover there is a possibility for determining the static and dynamic characteristics of the system in different operating conditions, to reveal bottlenecks and to achieve component-by-component analysis and sufficiently simple reorganization of the program modeling system.

An emulating translator compiles the initial text from GENL into an intermediate language L_2. Prior to this, a system for tuning the translator to a

fixed language L_2 and the structure monitor to control the program modeling process is foreseen. Interaction between individual subsystems and the user is achieved by modules of the system's interface--clusters, which are generated automatically with regard for the requirements of the program modeling system. The subsystem statistically gathers statistics pertaining to instructions and branches in the user's program, it calculates the loading characteristics of individual modules of the problem-oriented computer complex being modeled, and it plots time diagrams of the function of components of the problem-oriented computer complex. The debugging system makes it possible to obtain complete information on processes occurring in the problem-oriented computer complex being modeled. The modeling process is carried out in batch or in interactive mode.

To permit modeling of parallel asynchronous processes, the GENMOD system contains a structure monitor, at each control level of which time is reckoned separately by timers synchronized by an upper-level monitor. When homogeneous multiprocessor systems are modeled, only one copy of the program model exists at each moment in time, and in this case the structure model provides for a reduction and expansion of the principal data region. Modeling proceeds phase by phase. In order to insure effective operation of the program modeling system in time, the upper-level monitor automatically selects the modeling step, which makes it possible to avoid reduction of a process if no data transfer occurs in the given time interval. The size of a modeling step is determined by the following formula:

$$t_{i+1} = k\, t_i,$$

where k=1.5-2 and changes sign at points t_i=0 and t_i=t_k; t_k--critical interval (selected equal to the maximum duration of the instruction used most frequently).

Peripheral devices are modeled as blocks containing their own structural components, which are reflected in particular areas of the memory in the program modeling system. The logical input-output system permits dynamic buffering and access on the basis of the index under the control of the structure monitor.

The GENMOD system has been implemented in the operating system of a YeS computer, and it was used to plan the specialized Etalon computer complex and a problem-oriented subsystem for a macro-pipeline computer. The total volume of the development is about 5,000 commands in PL/1 language.

BIBLIOGRAPHY

1. Smolov, V. B., (editor), "Spetsializirovannyye TsVM" [Specialized Digital Computers], Moscow, Vysshaya shkola, 1981, 278 pp.

2. Glushkov, V. M, Kapitonova, Yu. V. and Letichevskiy, A. A., "Avtomatizatsiya proyektirovaniya vychislitel'nykh mashin" [Automation of Computer Design], Kiev, Naukova dumka, 1975, 230 pp.

3. Smol'nikov, V. A., "Generation of Cross Program Resources for Modeling Computer Systems," USIM, No 5, 1982, pp 63-65.

11004
CSO: 1863/259

UDC 681.3.01

MULTIPROCESSOR SYSTEMS WITH DYNAMIC REDISTRIBUTION OF REQUESTS THROUGH A COMMON BUS

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 3, Mar 85 (manuscript received 10 May 84) pp 33-38

[Article by V. A. Bogatyrev, Leningrad]

[Text] The author proposes an approach to the evaluation of the reliability of multiprocessor systems for dynamic distribution of requests with the use of partially efficient states of the processors and the limited capacity of the common bus.

Modern modules of computing equipment and, particularly, microprocessors are multifunctional devices. Multifunctional modules, due to incomplete coincidence of circuits activated during the performance of various functions, can preserve their ability of realizing part of the functions after failures. Such states of multifunctional modules are called partially efficient. The use of partially efficient states of processors with dynamic distribution of requests opens additional possibilities of increasing the realibility of process control computer complexes (UVK) [1, 2].

This article examines the use of the principle of dynamic distribution of requests in UVK with each of the processors controlling an object requiring the performance of the functions (programs) f_1, f_2 ..., f_n for its servicing. Information exchange among the processors is accomplished through a common system bus (OM). In such UVK, when requests arrive from the j-th object for performing the function f_i, the equipment checks the ability of the j-th processor (Pr_j) to perform this function. If Prj cannot fulfill f_i, then the request is redistributed through the common bus to one of the processors which preserved the ability to fulfill it. This realizes dynamic distribution of requests from the j-th object with the priority of loading the j-th processor which accepts all requests for the functions perserved by it. The possibilities of load redistribution are limited by the carrying capacity of OM and by the overloading of the processors due to the additional handling of the redistributed requests. The carrying capacity of OM has the fundamental influence on the limitation of the potentialities of the system when the time of the redistribution of requests through OM is commensurable with the time of their fulfillment by the processors. For real UVK, such cases take place, for example, when there are large data files necessary for handling requests and their transmission through one common line at great distances.

This article gives an evaluation of the carrying capacity of a common bus on the reliability level of UVK with dynamic distribution of requests. This level is compared with the reliability level achieved in load redistribution systems when processors are cut off after their first failure.

The carrying capacity of OM is determined from the condition of the existence in it of a steady-state exchange mode $R < 1$ [3], where R is the OM load. If the object and processors are uniform,

$$R = \Lambda_0 \vartheta_0 + \sum_{i=1}^{n} \vartheta_i \Lambda_i k_i,$$

k_i -- total number of f_i functions lost in all processors; Λ_i -- intensity of requests from one object for the performance of the f_i function; ϑ_i -- average time of redistribution of the request through OM for the performance of f_i; Λ_0 -- total intensity of requests for interprocessor exchange through OM from all processors; ϑ_0 -- average length of interprocessor exchange.

If $\vartheta_i \approx \vartheta_j \approx \vartheta$ $(i, j = \overline{1, n})$, then $R = \Lambda_0 \vartheta_0 + \vartheta \Lambda$, where $\Lambda = \sum_{i=1}^{n} \Lambda_i k_i$.

When, moreover, $\Lambda_i = \Lambda_j = \Lambda_f$, then $R = \Lambda_0 \vartheta_0 + k \vartheta \Lambda_f$, $k = \sum_{i=1}^{n} k_i$. From the condition of steady-state exchange through OM $R < 1$, we find the limit Θ of the number of requests redistributed through OM as the nearest integer not greater than $(1 - \Lambda_0 \vartheta_0)/\Lambda_f \vartheta$.

Without considering the carrying capacity of OM for the most unfavorable case when all of the processors lose functions of the same kind, UVK can withstand m failures, and n(m-1) failures for the most favorable case. In the latter case, UVK retains its ability to perform all n kinds of functions. Thus, $m \leqslant \Theta \leqslant n(m-1)$, m -- number of processors in UVK.

The probability of faultless operation of UVK whose OM makes it possible to redistribute the load until the appearance of Θ failures is defined as

$$P_{\Gamma} = \sum_{k=0}^{\Theta} P_k,$$

where P_k -- probability of the efficient state of UVK with the total loss of $k = \sum_{i=1}^{n} k_i$ functions by the processors.

Let us use Φ_i to denote the equipment of the processor activated during the performance of function f_i in the general case of $\Phi_i \cap \Phi_j = \varnothing$. The probability of the performance of function f_i by the processor is defined as $p_{f_i} = \exp(-\lambda_{f_i} t)$, where λ_{f_i} is the total intensity of the failures of the equipment Φ_i. Let us represent the state of the system by the matrix $\|\varphi_{ij}\|_{n \times m}$ whose element $\varphi_{ij} = 1$, if the j-th processor is capable of performing function f_i, otherwise $\varphi_{ij} = 0$. UVK retains its efficiency when each function $f_i \in F$, $F = \{f_1, f_2, \ldots, f_n\}$ is performed by at least one processor, i.e., if the matrix $\|\varphi_{ij}\|$ does not contain rows with all zero elements.

If $p_{f_i} = p_{f_j} = p_j$ and $\Phi_i \cap \Phi_j = \emptyset$, then UVK retains its efficiency after k failures with the probability of

$$P_k = \left[C_{mn}^k - \sum_{b=1}^{n} (-1)^b C_n^b C_{m(n-b)}^{k-bn} \right] p_f^{mn-k} (1-p_f)^k.$$

The expression in the square brackets corresponds to the number of all kinds of matrices $\|\varphi_{ij}\|$ containing k zeros in the absence of zero rows.

Let us assume $(\forall i)(\forall j)[(i \neq j) \rightarrow \Phi_i \cap \Phi_j = \Omega]$, i.e., some common equipment Ω is used in the processor in fulfilling all functions. Failure of the Ω equipment leads to simultaneous loss of all functions of the processor, while failure of the equipment $A_i = \Phi_i \setminus \Omega$ is connected with the loss of one function f_i, therefore, if $p_{f_i} = p_{f_j}$, then

$$P_k = \sum_{a=0}^{r} \left\{ C_m^a p_\Omega^{m-a} (1-p_\Omega)^a \sum_{l=0}^{k-na} \left[C_{(m-a)n}^l + \sum_{b=1}^{n} (-1)^b C_n^b C_{(m-a)(n-b)}^{l-b(m-a)} \right] \times \right.$$
$$\left. \times p_A^{(m-a)n-l} (1-p_A)^l \right\},$$

r -- the nearest integer not greater than k/n.

$$p_\Omega = \exp(-t\lambda_\Omega), \quad p_A = \exp(-t\lambda_A),$$

λ_Ω, λ_A -- failure rate of the processor equipment Ω and $A = \Phi_i \setminus \Omega$. Usually, the processor equipment Ω includes an arithmetical-logical block used in performing all programs, and the equipment $A_i = \Phi_i \setminus \Omega$ includes zones of storage memory of f_i program.

if $p_{f_i} \neq p_{f_j}$ and $\Omega = \emptyset$, then

$$P_k = \sum_{j=1}^{d} \left(\prod_{l=1}^{n} C_m^{k_{lj}} p_{f_l}^{m-k_{lj}} (1-p_{f_l})^{k_{lj}} \right),$$

where d is the number of various expansions of k into n addends

$$k = k_{1j} + k_{2j} + \ldots + k_{nj}, \quad j = \overline{1,d}, \quad 1 < k_{ij} < m,$$
$$d = C_{k-1}^{n-1} + \sum_{\varepsilon} (-1)^\varepsilon C_n^\varepsilon C_{n+k-\varepsilon m}^{n-1} - C_{n+r-1}^k,$$

here C_{k-1}^{n-1} defines the number of divisions of k into n nonzero addends, the second term -- the number of divisions of k into n addends assuming the value 0.1, ..., m (summation is done as long as the inequality $k - \varepsilon n+1 \geqslant 0$ is fulfilled), C_{n+r-1}^k -- total number of divisions of k into n addends [5].

If $\Omega \neq \emptyset$ and $p_{f_i} \neq p_{f_j}$, then

$$P_k = \sum_{a=0}^{r} \left\{ C_m^a p_{\text{в}}^{m-a} (1-p_{\text{в}})^a \sum_{j=1}^{d} \prod_{l=1}^{n} C_n^{k_{lj}} p_{A_l}^{m-k_{lj}} (1-p_{A_l})^{k_{lj}} \right\}.$$

Here, d -- the number of divisions of the number (k-an) into n addends assuming the value $1 \leq k_{ij} \leq m-a$.

If $\theta = n(m-1)$, i.e., the carrying capacity of the common bus does not limit the reliability level of UVK, than, at $(\forall_i)(\forall_j)(i \neq j) \rightarrow \Phi_i \cap \Phi_j = \Omega$), the probability of faultless operation of UVK is defined as [1,2]:

$$P_F = \left\{ \prod_{l=1}^{n} [1-(1-p_{f_l})^m] \right\} / \{1-(1-p_{\text{в}})^m\}^{n-1}. \qquad (1)$$

Realization of the proposed method requires certain complication of the integration modules (MK) ensuring the redistribution of the load and organization of information exchange through OM with respect to MK systems without the use of partially efficient states. Complication is connected with the consideration of the ability of the processors to perform each type of requested functions with load distribution. For this purpose, in the case of decentralized control, each MK has a memory microcircuit storing the matrix $\|\varphi_{ij}\|$ nxm to whose address input of the row the code of the requested function is delivered, and the output of the searching meter of the processor capable of performing the requested function is connected to the address input of the row. The buildup of content of the meter determining the address of the processor intended for loading through OM is realized by the equipment until one is delivered from the microcircuit storing the matrix $\|\varphi_{ij}\|$. After that the information necessary for the performance of the requested function starts through OM. The circuits of information transmission through OM for the MK variants being compared are similar. The realization of the proposed method of increasing reliability makes it necessary to complicate MK by 5-10%. These expenditures are small in comparison with the equipment of complex processor modules and, therefore, will not be henceforth taken into consideration. The proposed method of ensuring reliability makes it necessary also to have more complicated software due to the necessity of forming $\|\varphi_{ij}\|$ in each MK after failures of the processor.

Let us compare the reliability level of an UVK with dynamic distribution of requests using partially efficient states and the reliability level of an UVK without the use of partially efficient states:

$$P_c = \sum_{l=0}^{g} C_m^l p_{\text{м}}^{m-l} (1-p_{\text{м}})^l, \quad p_{\text{м}} = p_{\text{в}} \prod_{l=1}^{n} p_{A_l},$$

g -- maximum number of failed processors at which the carrying capacity of the common bus makes it possible to redistribute requests from processors disconnected after the first failure; g is equal to the nearest integer smaller than $(1-\Lambda_0\vartheta_0)/n\Lambda_f\vartheta$.

Let us examine some examples of the evaluation of the efficiency of using partially efficient states on the basis of dynamic distribution of requests.

Let us assume that $m = n = 5$, $\theta = 4$, $p_{f_i} = p_{f_j}$, $p_\Omega = p_M^\gamma$, $p_A = p_M^{\frac{1-\gamma}{n}}$, $P_F = p_\Omega^m \sum_{k=0}^{4} \times$ $\times C_{mn}^k p_A^{mn-k} (1-p_A)^k$ and $P_c = p_M^m$. . Computation results of $\delta(\gamma, p_M) = \frac{P_F}{P_c}$ are shown in Figure 1.

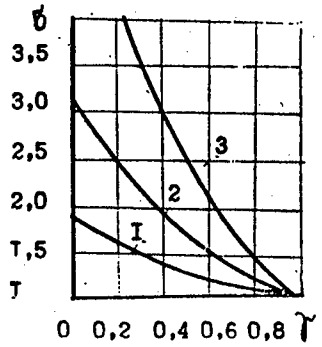


Figure 1. Relative gain in UVK reliability from using partially efficient states of processors: m=n=5, θ=4; 1 -- p_M=0.9; 2 -- p_M=0.8; 3 -- p_M= 0.7.

Let us examine a case when the carrying capacity of the common bus does not affect the UVK reliability level, i.e., if θ = (m-1)n, then P_F is determined by formula (1), and $P_c = 1-(1-p_M)^m$. The evaluation results of $P_F(\gamma, p_M, n)$ and $P_c(\gamma, p_M)$ are shown in Figure 2. Figures 1 and 2 show a high effectiveness of using partially efficient states of processors on the basis of dynamic distribution of requests. The effect increases as the number of functions performed by the system (n) increases, as well as when processors and their operating conditions become more complicated (decrease of p_M). The effectiveness increases as the degree of the interdependence of failures of the performance of various functions by the processor decreases (decrease of γ). It is natural that at $\gamma = 1$, i.e., when the same equipment is used for performing all functions, $P_F = P_c$.

An approach to the evaluation of the reliability of multiprocessor systems of dynamic distribution of requests with consideration of the carrying capacity of the common bus is proposed. A high effectiveness of increasing reliability on the basis of the use of partially efficient states of processors is shown. The obtained results can be used in designing failure-free multiprocessor systems.

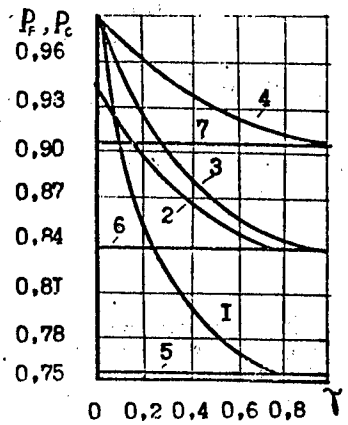


Figure 2. UVK reliability (at m=2):
UVK with the use of partially efficient states:
1 -- p_M=05, n=100; 2 -- f_M=0.6, n=4; 3 -- p_M=0.6, n=100; 4 -- p_M=0.7, n=100.
UVK without the use of partially efficient states;
5 -- p_M=0.5; 6 -- p_M=0.6; 7 -- p_M=0.7.

Bibliography

1. Bogatyrev, V. A. "Toward Increasing the Reliability of Computing Systems on the Basis of Dynamic Distribution of Functions", IZVESTIYA VUZOV SSSR -- PRIBOROSTROYENIYE [News of Vuzes of the USSR -- Instrument Making], 1981, No 8, pp 62-65.

2. Bogatyrev, V. A. "Toward Increasing the Reliability of Multimicroprocessor Systems", IZVESTIYA VUZOV SSSR -- PRIBOROSTROYENIYE, 1982, No 8, pp 93-96.

3. Mayorov, S. A., et al. "Osnovy teorii vychistilel'nykh sistem" [Fundamentals of the Theory of Computing Systems], Moscow, Vysshaya shkola [Higher School], 1978.

4. Bogatyrev, V. A. "Toward Increasing the Reliability of Data Processing Systems using the Multifunctional Characteristics of Modules", in the book "Sistemy sbora i obrabotki izmeritel'noy informatsii" [Systems for Acquisition and Processing Measuring Information], Taganrog, 1981, No 3, pp 52-55.

5. Kofman, A. "Vvedeniye v prikladnuyu kombinatoriku" [Introduction to Applied Combinatorics], Moscow, Nauka, 1975.

10233
CSO: 1863/280

UDC 681.3.06:51

USING DIGITAL SIMULATION IN THE DESIGN OF CAMAC SYSTEMS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (manuscript received 11 Nov 83) pp 46-48

[Article by engineers P. V. Gamin, S. V. Morgunov and M. A. Shamashov, and Candidate of Technical Sciences M. A. Korablin]

[Text] Today most information-measuring systems use program-controlled CAMAC standard bus-module systems as linking devices when designing automated scientific research systems (ASNI) and when performing bench tests [1]. These systems do not provide developed resources for debugging the programs, which sharply reduces the effectiveness of the process of design and debugging programs directly by means of the control microcomputer contained within the composition of the information-measuring system. On the other hand the diversity of the experiments that are carried out generates the need for creating new specialized CAMAC standard devices. The problems that arise in this case include those of determining the internal logical organization of the devices, time synchronization, studying reactions to external controlling influences and acquiring reliable programs for servicing such devices. The methods of simulational modeling are enjoying wide use as a way to solve these problems [2]. In particular, a simulational model of an information-measuring system consisting of an emulator with a full configuration based on a YeS computer has been proposed as a tool for design the hardware and software base of an information-measuring system [3]. This will make it possible to automate the work of testing and improving CAMAC hardware; to expand the possibilities for debugging programs for information-measuring systems owing to utilization of the rich arsenal of debugging resources available in the instrumental computer; to carry out parallel design of new CAMAC modules and debugging of their service programs, as a result of which the time required to design and place information-measuring systems into operation will decrease by a factor of about 1.5-2; to collect statistics on the work of individual components of the information-measuring system with the purpose of studying the load on the CAMAC bus and the CAMAC modules, and using this information to improve or modify both individual components and the information-measuring system as a whole.

The complete emulator includes the program emulator of the minicomputer [4] and a simulation model of the CAMAC, the function of which is based on an identification-interpretation algorithm [3]. The model of a crate bus is adopted as the basis for the model of the CAMAC system. The model of the

groups of lines of the bus does not differ in any way from the model of the registers and flip-flops of CAMAC modules, and it is simulated by memory elements and registers of the instrumental computer. Simulation of the processing of CAMAC instructions boils down to changing the content of the model of the internal registers and signals, or to monitoring their state over time. Thus a CAMAC operation interpreter creates the models of CAMAC systems. The structure of the interpretation block is based on the fact that the processing of any CAMAC instruction or controlling signal (I, C, Z) may be represented as the processing of a sequence of elementary actions (micro-operations):

blocking-deblocking of the input of devices *LAM* state bits and so on;

input-output of data at the "bus-device" level;

change of the contents of internal registers (enlargement of a value by 1, nullification and so on);

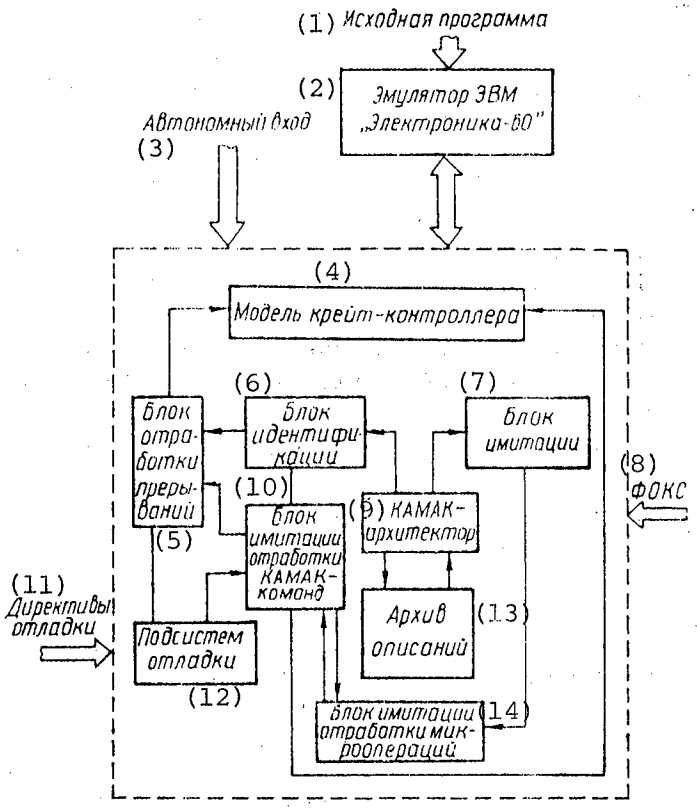
analysis of the state bits of a module and formation of its corresponding output signals;

setting signals X and Q, and so on.

Analysis of the function of a number of CAMAC modules showed that 14 types of such microoperations are typical of most of them. Creation of programs to interpret isolated microoperations made it possible to model a wide spectrum of CAMAC instructions and general control signals by superimposing references to these programs. Use of this approach insured universality of the CAMAC emulator and a possibility for modeling CAMAC systems of various configurations without rebuilding the simulation block, as well as a significant savings of the memory of the instrumental computer, without reducing emulator speed.

The CAMAC emulator includes a debugging subsystem which operates in batch and interactive mode. The functions of this subsystem are, in particular, recording and displaying information on the state of lines, flip-flops and registers of the modeled modules and buses, which provides a possibility for studying the functional dynamics of the CAMAC system and using the CAMAC simulation model to select and analyze the logical organization of different sets of instructions for the hardware being designed, to test models of CAMAC devices and to study the reactions of both an individual module and the system as a whole to certain controlling influences (a flow of instructions and/or general control signals, interrupt signals and so on). The emulator is started up in this mode by way of the "independent input."

Programs servicing CAMAC hardware are debugged by an emulator with a complete configuration. A program written in ASSEMBLER language for the Elektronika-60 microcomputer and debugging commands are fed to the input of the emulator. A cross system [4] performs translations, editing and layout and transfers control to the microcomputer's emulator to simulate running of the program. When program errors or errors in the algorithms for interaction with CAMAC hardware are detected, the full-configuration emulator switches to waiting



Structure of the Model of a CAMAC System

Key:

1. Initial program
2. Elektronika-60 computer emulator
3. Independent input
4. Model of crate-controller
5. Interrupt processing block
6. Identification block
7. Simulation block
8. FOKS [CAMAC system functional description]
9. CAMAC architect
10. CAMAC instruction processing simulation block
11. Debugging directives
12. Debugging subsystems
13. Description archive
14. Microoperation processing simulation block

mode, giving the user the possibility for eliminating the error (for correcting the flow of instructions, data, addresses and so on) and restarting the corrected program at the necessary point.

The debugging directives include complete and selective tracing, information input and statistics collection operators. The tracing operators are used to read out information on the state of the instruction counter, addresses, operands and their values, the state of buses and devices in the CAMAC standard and so on. This provides a possibility for visualizing the entire calculation process both in the Elektronika-60 and in CAMAC hardware. Analysis

of the displayed information makes it possible to locate and eliminate bottlenecks in programs undergoing debugging. The values of registers and flip-flops in the CAMAC devices being modeled can be changed with the help of information input operators, use of which during debugging provides a possibility for modeling the dynamics of the object of analysis in the information-measuring system, and thus achieving integrated program debugging.

Typical features of information-measuring systems based on CAMAC hardware include flexibility, the capability for adaptation in the course of the most diverse experiments, and ease of changing the configuration of the CAMAC system, owing to presence of a descriptions archive and CAMAC system model architect in the emulator. The service programs of these systems use the CAMAC device control and description operators, written in CAMAC system functional description (FOKS) language developed by the authors; they are responsible for forming the archive, correcting it and generating programs to simulate CAMAC instructions and signals. The obtained programs are translated, edited and packaged for the purposes of forming, together with the core of the CAMAC emulator, a simulation model of the information-measuring system.

The modeling system described here was created in the operating environment of a YeS DOS. It was introduced into a number of organizations, and it is being operated successfully in the design of firmware for physicotechnical automated scientific research systems. The system is presently being translated for work in the operating environment of a YeS OS.

BIBLIOGRAPHY

1. Vukolikov, V. M., Vystavkin, A. N., Oleynikov, A. Ya. et al., "Maintenance of Input and Output Characteristics of Experiment Automation Systems Using Small Electronic Computers and CAMAC Hardware," PRIBORY I TEKHNIKA EKSPERIMENTA, No 1, 1982, pp 7-13.

2. Boltyanskiy, A. A., Vittikh, V. A., Korablin, M. A. et al., "Tsifrovaya imitatsiya avtomatizirovannykh sistem" [Digital Simulation of Automated Systems], Moscow, Nauka, 1983, 262 pp.

3. Kovartsev, A. N., Korablin, M. A. and Shamashov, M. A., "Simulation Modeling of Experiment Automation Systems Using Full-Configuration Emulators," USIM, No 4, 1979, pp 124-127.

4. Kalechits, V. Ye., Lobodin, N. I. and Shchukin, A. Ya., "A Cross System of the Elektronika-60 Computer for the YeS Computer," in "Metody translyatsii" [Translation Methods], Rostov-on-Don, 1981, pp 52-59.

11004
CSO: 1863/259

UDC 621.317.725

WIDE-BAND SINGLE-BOARD ANALOG-DIGITAL CONVERTER

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript received 31 January 1984), pp 108-112

[Article by A.N. Kasperovich and Yu. V. Shalaginov, Novosibirsk]

[Text] The use of fast domestically built series 1107 large-scale integrated circuits (LSI) as parallel 6- and 8-bit analog-digital converters (ADC) has considerably broadened the possibilities for creating digital signal processing systems. However the practical use of such devices requires the development and assembly on the circuit board of a buffer amplifier for the input signal (because of the need for rapid conversion of the large input volume by the ADC microcircuit), a reference voltage source (-2V) and a trigger pulse formation unit. We note that the TRW company which produces this type of microcircuits also produces functionally complete ADC which are complete on a single board where, along with the LSI ADC there is also the required auxiliary equipment [1].

Experimental testing of the dynamic characteristics of the domestically made LSI parallel ADC (by the method described below) showed that these ADC, although based on the parallel conversion method do not produce the necessary precision when working with high-frequency signals. Thus, for the 8-bit ADC with 20 MHz sampling for a harmonic signal with an amplitude swing over the entire range and a frequency of 7 MHz, the conversion error amounts to around 8 quantizing steps (for a signal with a frequency of 10 MHz it is up to 16 quantizing steps) which does not agree with the requirements of the count theorem. In order to improve the dynamic characteristics of the ADC it is necessary to use a signal sampling and storage unit (SSU). We note that similar recommendations are given in [2]. We describe below the LSI parallel 8-bit single-board wide-band ADC with SSU. A method is given for testing the dynamic parameters of the ADC.

A block diagram of the ADC is given in Fig. 1. Along with the LSI ADC it contains the SSU, reference voltage source and trigger pulse shaper. The most complex and important of these circuits is the SSU for whose precision and response speed there are strict requirements. It is important that the SSU not only carry out qualitative conversion of signals in the 0-10 MHz range but that they have short sampling and storage times (approx. 20 ns) and also low impedance. This is necessary so that use of the SSU at the ADC input does not lead to a lowering of the maximum permissible digitizing frequency of the ADC. The development of such SSU is a complex technical problem.

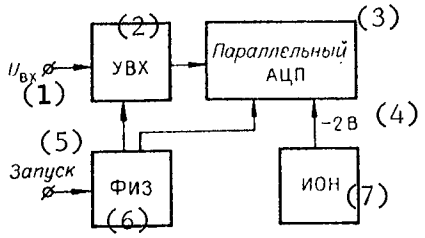


Figure 1

Key:
1. U_{input}
2. Signal sampling and storage unit
3. Parallel ADC
4. -2V
5. Trigger
6. Trigger pulse former
7. Reference voltage source

The SSU is designed on the basis of an open circuit according to the operating principle for such SSU described in [3] but is unlike this design in that it is realized using a conventional rather than differential circuit. Lowering of the effect of commutation noise is attained by reducing the transient time for commutation (through the use of elements with faster response times).

The SSU schematic diagram is shown in Fig. 2. The input signal passes through two controlled emitters of repeaters (T3 and T5), loads a capacitor C_{zap}, goes through repeater transistors with a different type of conductance (T8-T11) and passes to the output. The input repeater emitters are controlled by the current switch (T6, T7). In tracking mode the current switch feeds the emitter circuit of transistor T7 without affecting the potential of its base. In transition to storage mode, the collector current of transistor T6 creates a negative shift in the base potential of transistor T5 and emitter potential of T3 which blocks these transistors. The change in the potential of this circuit is limited by diode D4 to the level of -2.5 V. Simultaneously, the emitter current of transistor T5 is cut off. The use of this kind of switch is the second advantage of the SSU over that described in [3] and is the best protection of the voltage setting of the storage capacitor against the action of the input signal (through parasite capacitance).

The signals for control of the current switch (T6, T7) are formed by means of the differential pair T1 and T2 and the E1 microcircuit. Here the necessary shift of signal levels is carried out by means of diodes D1-D3 in the feed circuit of the E1 microcircuit and diodes D5, D6 which are connected to the output of the microcircuit.

The SSU can be used for signal conversion in two ranges: $\pm$ 1V and 0 to -2V. In the first case, a shift of the signal is required to make it agree with the LSI range of o to -2V. This shift is made by passage through the resistor R1 of the stable collector current of transistor T4. For operation in the second range, the collector of transistor T4 is connected directly to the output of repeater T3.

The reference current source utilizes an operational amplifier with 100% negative feedback.

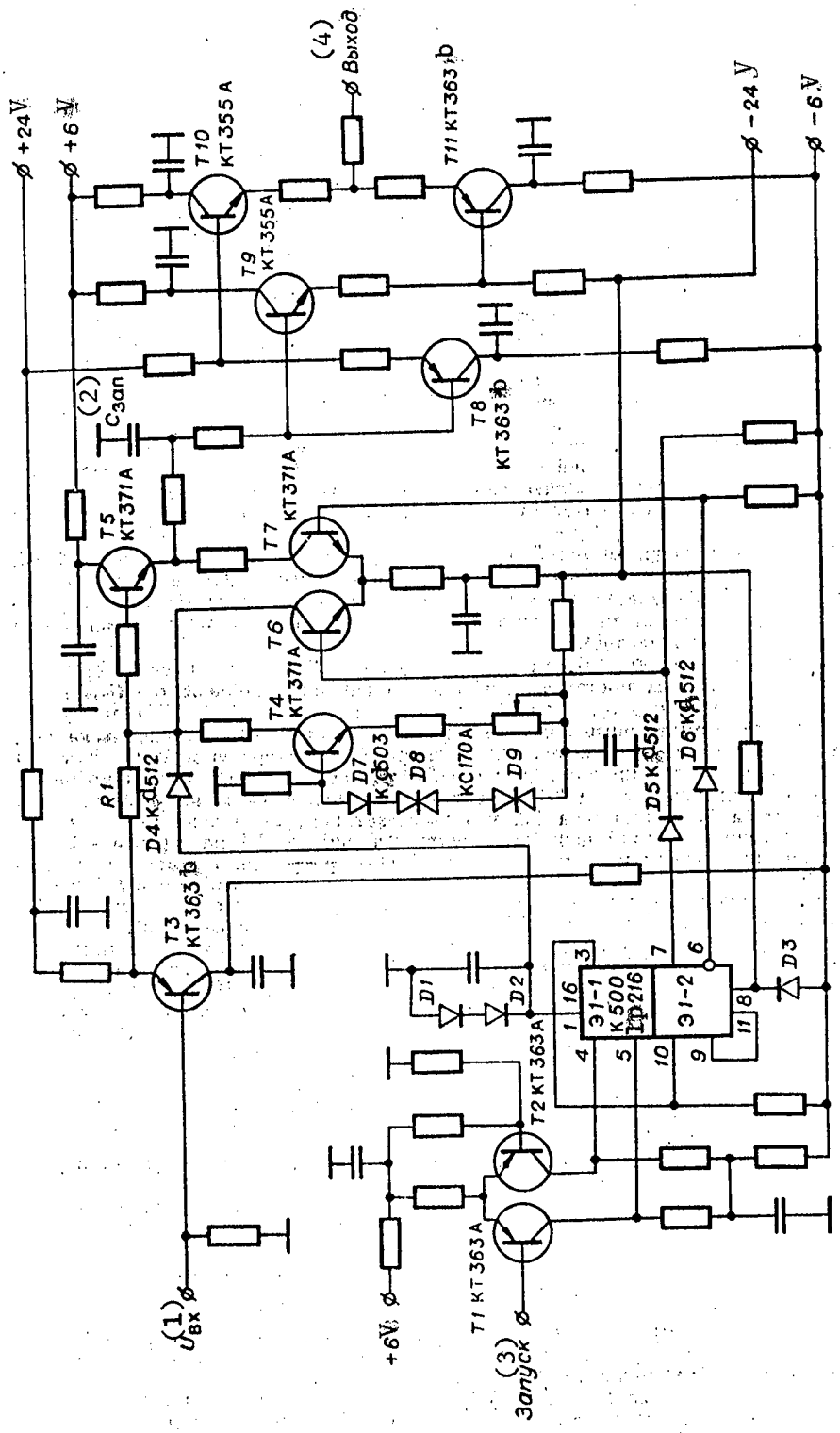


Figure 2

Key:

1. U_{input}
2. Storage capacitor
3. Trigger
4. Output

The trigger pulse shaper generates two sequential pulses with positive polarity at the transistor-transistor logic (TTL) levels with a length of 20 ns. The pulses which gate the ADC lag relative to the sample pulses of the SSU by 10-12 ns. The trigger pulses are formed by series 531 elements.

The dynamic parameters of the ADC were studied by a specially developed method which is a further development of the stroboscopic method described in [4]. We note that a very similar method (beat method) also makes it possible to observe periodic signals regenerated by a DAC at the output of the ADC for all points of the scale. This method is considered in [5].

The main drawback of method [4] is the relatively low quantizing frequency determined by the trigger frequency of the strobe oscillator. The drawback is also, to a certain extent, characteristic of the beat method [5] where the quantizing frequency of the ADC is selected at approximately the frequency of the harmonic test signal.

In order to monitor the parameter of the ADC at the working quantizing frequencies a pulse burst generator was introduced into the trigger channel of the ADC and is triggered by the strobe pulse of the stroboscopic oscillator (Fig. 3). The frequency of this generator can be regulated within significant limits and this makes it possible to test the ADC at different quantization frequencies including the limiting frequency.

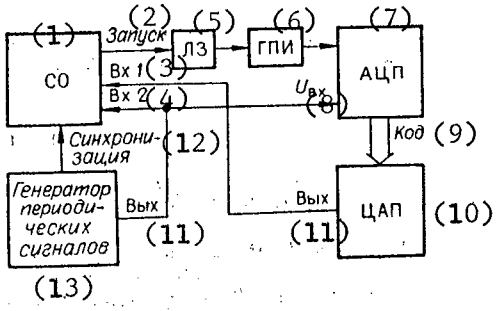


Figure 3

Key:

1. Stroboscopic oscillator
2. Trigger
3. Input 1
4. Input 2
5. Delay line
6. Pulse burst generator
7. ADC
8. U_{input}
9. Code
10. DAC
11. Output
12. Synchronization
13. Periodic signal generator

It was proposed to monitor the dynamic errors of the ADC by means of signal difference. In this case, the oscillograph is connected in signal subtraction mode (conversion and regeneration) and the phase shift on the oscillograph shows the signal difference corresponding to the converter error. The phase shift on the screen of the signals is produced by adjusting the number of pulses in the burst (roughly) and by means of the adjustment of the wide-band delay line (fine). In signal difference mode it is possible to evaluate the different components of ADC error. Integral nonlinearity is evaluated by the degree of distortion of the signal difference line and differential nonlinearity is evaluated by the

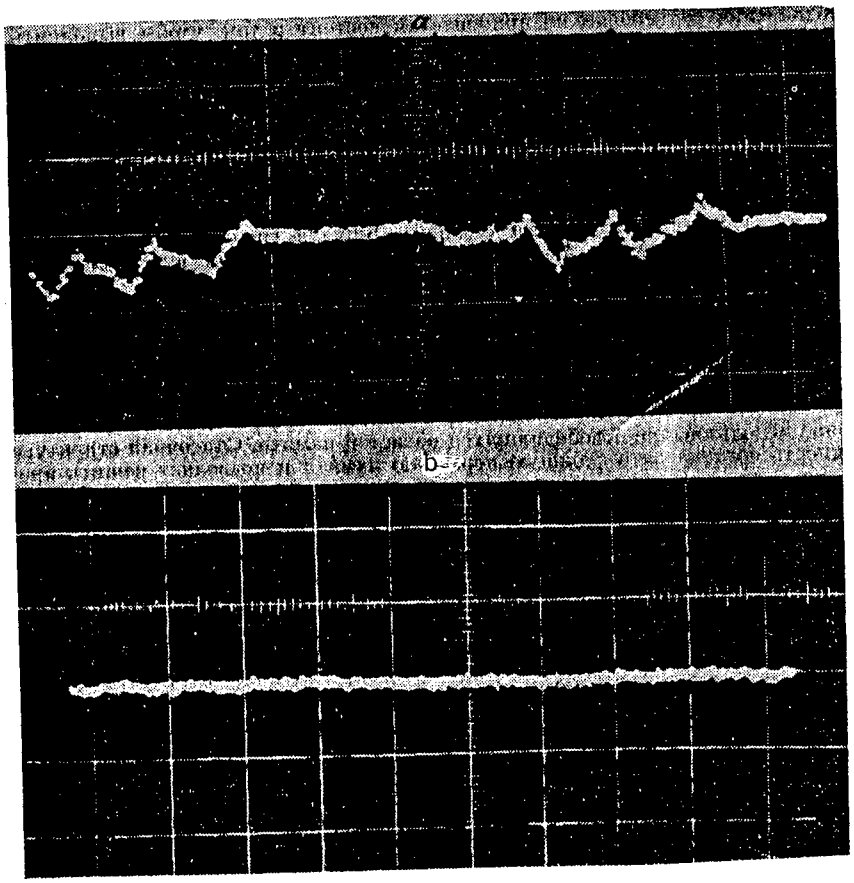


Figure 4

value of the spread of error amplitudes at different points of the ADC scale (this form of error can be conveniently observed in large extensions of the signal along the time axis. Aperture error takes the form of jitter of the differential signal amplitudes. Here it should be kept in mind that in evaluating aperture errors, errors are involved which arise from the time indetermination of the pulses in the trigger channel of the ADC and from the synchronization of the oscillator. There are therfore strict requirements for speed of response and noise protection for the trigger channel elements.

Fig. 4, a, b show oscillograms of difference signals obtained by subtraction of the regenerated signal from the converted harmonic signal with a frequency of 7 MHz and an amplitude sweep equal to the range of the ADC. Fig. 4, a shows the case of the use of the ADC without SSU and Fig 4, b shows use of the SSU. In Fig. 4, a the integral nonlinearity of the ADC is clearly visible and consists of several quantizing steps (the width of the line corresponds approximately to 2 quantizing steps).

The ADC was realized by means of a single-width CAMAC module and has the following parameters: number of bits: 8; response speed up to 20 million conversions/s. Testing of the dynamic characteristics of the ADC by the described method showed the high degree of effectiveness of the use of the SSU with the LSI parallel-type ADC. The dynamic error of the ADC for frequencies to 7 MHz is $\pm$ 1 quantizing step.

Bibliography

1. "Plata dlya bystrogo analogo-tsifrogo preobrazhaniya" [Printed Circuit Board for High-Speed Analog-Digital Converter], ELEKTRONIKA, 1981, NO 16.

2. Michaels, Stuart R. "Mend Flash-Converter Flaws with Track/Hold Core", END, 1981, No 30.

3. Yefremov A.I., A.N. Kasperovich, N.V. Litvinov and Yu. V. Shalaginov. "Shirokopolosnyy analogo-tsifrovoy preobrazovatel'" [Wide-Band Analog-Digital Converter], AVTOMETRIYA, 1981, No 6.

4. Belomestnykh V.A., V.N. V'yukhin and A.N. Kasperovich. "Ob odnom sposobe eksperimental'nogo opredeleniya dinamicheskikh svoystv bystrodeystvuyushchikh ATsP" [A Method for Experimental Determination of the Dynamic Characteristics of a High-Speed ADC], AVTOMETRIYA, 1976, No 5.

5. Nil M. and A. M'yuto. "Dinamicheskiy kontrol' analogo-tsifrovykh preobrazovateley" [Dynamic Monitoring of Analog-Digital Converters], ELEKTRONIKA, 1982, No 4.

12497
CSO: 1863/146

CYBERNETICS INSTITUTE DEVELOPS PARALLEL-PROCESSING COMPUTER

Moscow PRAVDA in Russian 22 Feb 85 p 3

[Abstract] The lengthy article reports on the development of a multi-processor computer that operates on the parallel-processing principle, at the Ukrainian Academy of Sciences' Cybernetics Institute imeni Glushkov,

It is recalled that several years ago, the institute's director, academician Viktor Mikhaylovich Glushkov, who is now deceased, proposed the so-called 'macro-assembly-line principle' of organizing computations as one way of realizing recursive computers. The concept uses the analogy of a number of plants making assemblies and parts that go into an end product. Under Glushkov's direction, specialists of the institute began elaborating and substantiating in theory the new principle of organizing computations. It is said that the control system which ensures effective execution of operations under a task-sharing system is a 'macro-assembly-line computer.' The work reportedly led to a success which has demonstrated the possibility of building computing systems which ensure growth in capacity practically directly proportional to the number of processors that are linked into the system.

It is said that the computer which has been developed for this occupies a room only 14-15 square meters in area, and is contained in several cabinets. For communicating with the computer, a language called MAYAK was developed; the letters are an acrynym which stands for 'macro-assembly-line language of computer programming.' Together with a system of parallel programming and an on-line system, it makes it possible for extremely complex tasks which would take a year to accomplish by conventional means to be done in a matter of weeks or days. The technology is said to have no counter-parts in world mathematical machine building, and it has been recommended for series production.

Professor Yu. Kapitonov, who took part in the development, is quoted, saying that the new computer complex was developed and made operational in a comparatively short time thanks to good cooperation among organizations of the ministries of the radio and electronics industries, and to the support of the USSR State Committee for Science and Technology (GKNT), and the presidium of the Ukrainian Academy of Sciences. The project demonstrated

that through coordination of efforts among academy and industry institutes and a plant, and with cooperation from industry leaders, the State Planning Committees of the USSR and of the Ukrainian SSR, and GKNT, lead time for the most advanced technology was reduced by five to six times.

It is said that among the important remaining tasks for the institute's staff is to train specialists capable of quickly solving problems of applying the new technology in various industries.

Academician V. Mikhalevich, the institute's current director, is mentioned.

FTD/SNAP
CSO: 1863/218

UDC 681.3.06

PROGRAM BASE FOR RASTER DISPLAY

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript received 10 May 1984) pp 104-107

[Article by E. A. Talnykin, Novosibirsk]

[Text] A previous paper* described the graphic display developed at the Automation and Electrometry Institute of the Siberian Department Academy of Sciences USSR. The object of the present paper is to give a more detailed idea of its programming and, more precisely, the programming for the graphic station based on the "Elektronika-60" microcomputer is discussed. The microcomputer controls the graphic controller and can have a communication link to another more fully equipped host computer. The only variants considered are those in which both computers have compatible architecture.

Organization of Programming

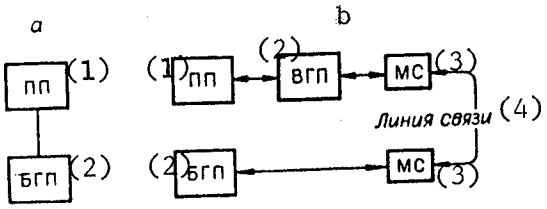
The basis for the organization of the programming is the primitive graphic base module which generates graphic display for other program components. The module is always located in the microcomputer. If the application program which requires graphic possibilities is also located in the microprocessor then all the problems of intermodular action are solved at the editing level of the external links (Fig. a).

For access to display for the application program from the host computer the diagram shown in Fig. b is used. The virtual graphic primitive module creates an interface for the application program which is identical to the base program while the real work consists in formulating a request in batch form and its transmission through the link module to the microcomputer. After reliable reception of the batch the link module breaks it up and organizes a request to the real module of the base graphic primitive module.

Base Graphic Primitives

The graphic primitives are realized in the form of functional requests with possible parameters. We will not structure the various primitives according to groups although it is possible to divide them into graphic station control equipment,

* Kovalev A.M. and E.A. Talnykin. "Graficheskiy displey rastrovo tipa dlya sistem dvukhkoordinatnogo proyektirovaniya" [Raster-Type Graphic Display for Two-Coordinate Design Systems], AVTOMETRIYA, 1984, No 4.



Key:

1. Application program
2. Vitual graphic primitives
3. Link module
4. Communication link

construction of representation elements, representation mode control and others.

DINIT

Put display equipment in working order. It suffices to give the the indicated request once during application of the resident application program.

TMODE/GMODE/OFFMODE

Set up the text, graphic or autonomous work modes respectively for graphic station operation.

FAST/SLOW

Set up high-speed protocol for exchange on communication link or a slower link which is more reliable.

FIELD (n,m)

Set up dimensions of virtual field so that the string will hold a minimum of n representation elements. Set up representation layering on m maps.

MAP(m)

Set up number of current maps equal to m.

PENSET (p)

Set up "pen" value. Drawing on display amounts to filling memory with units or zeroes. The value $p \neq 0$ sets a drawing mode by units while $p = 0$ sets a mode with zeroes.

PENFF

Inventory the state of the pen.

CLMEN

Delete representation. If the pen value is in units then the memory is filled with zeroes and in the reverse case it fills with units.

CLMAR

Delete representations from current memory map.

POINT (x,y)

Draw point at coordinates xy of the current memory map. This means that one bit of memory is set at unit or zero depending on the state of the pen.

LINE (x1, y1, x2, y2)

Join two points of the representation by a straight line.

BOX (x1, y1, x2, y2)
Draw on current map a rectangle with sides parallel to the axes and two assigned points as opposite vertices.

CIRCLE (x,y,d)
Draw on current map a circle with center at point x, y and with diameter d.

CBODY (x,y,d)
Draw on current map a circle with center at point x, y and diameter d.

CONNCT (x1, y1, x2, y2, d)
Join two points of current representation map with a straight line with width d.

GRID (x, y, Cx, Cy, s)
Draw on current map a network which locates a point at each node of the network. Here x, y are the coordinates of the lower left node of the network. Cx is the number of columns, Cy is the number of lines and s is the network step.

GETFUN (ch, x, y)
Introduce graphic data. After satisfying the request, the function button code is applied for variable ch while the variables x, y are the coordinates of the locator on the representation field. Input of information is possible only in graphic mode. The indicated request transfers the application program to the waiting state for operator input.

LOCXP/LOCXM/LOCYR/LOCYM
Move the locator to the right (left, up, down) to one element of the representation.

LOCSET (x,y)
Set the locator in position with coordinates x,y.

LOCOFF/LOCON
Turn locator on (off). The locator may not be represented on the screen. For example, the locator may be turned on only before the request for the introduction of graphic information.

SCRXP/SCRXM/SCRYP/SCRYM
Shift screen to the right (left, up, down) for the minimum possible shift. The screen shift step on the Y axis is one representation element and on the X axis it is 16 elements.

SCRSET (x,y)
Set the screen so that the point of the virtual field with coordinates x,y is at the upper left corner of the screen.

ZOOM (p)
Set the screen scale to p.

ZOOMP/ZOOMM
Increase (reduce) the screen scale if this is possible.

PRIOR
Change the priority of the memory maps for the representation. Sequential satisfaction of the request leads to a cyclical change in priorities.

MIXER
Set the color mix mode for representation intersections with various memory maps.

DISABLE (m)/ENABLE (m)
Turn off (on) the representation of memory map m. The representation of the switched-off map is not on the screen but is not deleted from the memory and, most important, it is possible to draw on the invisible map.

COLOR
Carry out changes in the types of map coloring. Sequential satisfaction of the request carries out cyclic color change.

SETPRI (p)
Set value of priority register to p. All requests for assignment of priorities in color mixes as well as the turning on and off of maps and changes in color types are effected by setting the value of the priority register.

MIRROR/NORMAL
Set the mirror (normal) system of coordinates in which the origin is at the upper left corner, the x axis is directed to the right and the y axis is directed down.

Communication Link Module

The communication link module of the host computer gets a packet from the virtual graphic primitives module and its task is reliable transmission to the recipient. The communication line for the program consists of the terminal itself and the link protocol is realized at the level of read/write string requests.

The main function of the microcomputer link module is transparent switching of the terminal. Symbols introduced by the keyboard are transmitted on the line (to the host computer) and the input from the line appears on the alphanumeric display screen. If, in the flow of symbols from the host computer, a correctly formed batch is detected then it is processed by the graphic primitives module. This is called the text mode since only text data comes from the terminal.

In graphic mode, amongst the symbols incoming from the keyboard, local control symbols are isolated which are processed by the base primitives. The input of any other symbol leads to transmission to the host computer of the packet containing the symbol code and the current coordinates of the locator on the representation field. Graphic information is thus transmitted from the terminal.

Off-line mode differs from graphic mode in that there is no transmission on the line towards the host computer. All symbols introduced from the keyboard and which are not local control symbols are simply represented on the screen of the alphanumeric display.

Local Control

Local control symbols act like function buttons and initiate execution of the following functions: shift of locator on screen, shift of screen in relation to representation, control of priorities and coloring, screen scaling.

Shift (of locator or screen) is controled for position or speed. Position control consists in shifting the object by discrete units in response to pushing the buttons (with directional arrows). In speed control, pressing the buttons with directional arrows leads to discrete increases in speed which may be zero or reverse (movement in reverse).

Conclusion

We will determine certain directions for the development of the base system under consideration. Adaptation of the virtual primitive module can create an interface for any graphic or standard batch. By changing the protocol section of the microcomputer it is possible to simulate a certain type of graphic terminal in order to implement application programming. It is quite easy to adapt the base graphic interface to various high-level languages.

12497
CSO: 1863/146

UDC 681.3.06

SHADOW GENERATION IN THREE-DIMENSIONAL VISUAL SCENE SYNTHESIS SYSTEM

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript received 23 January 1984), pp 107-108

[Article by B. Kh. Zinger, Novosibirsk]

[Text] 1. One of the main problems in the creation of systems for synthesizing visual scenes is improving the realism and clarity of the imagery. The main effect is attained by the removal of invisible surfaces. However the introduction of shadows in the synthesized scene makes it possible to sharply increase the illusion of depth of the imagery. The shadows supply significant information on the position of the observer and a more detailed understanding of the environmental situation.

In the present paper a method is proposed for generating shadows in three-dimensional space utilizing as initial information a data base consisting of polygons defined by an ordered set of coordinates of vertices and by the normal. The result of the work of the generation algorithm is a data base supplemented by polygons describing shadows. The method is based on the clipping of three-dimensional space by pyramids based on the shadows cast by the faces.

It is necessary to note that the shadow generator is a preprocessor in the real time imagery system with programming and hardware for the removal of invisible surfaces based on the priority mechanism [1,2]. The mechanism is that the data base to be represented is previously subjected to static sorting by a priority program and invisible surfaces are removed by the equipment during visualization by means of access of objects to the frame in order of priority.

It should be noted that for economic use of computer resources for representing visual scenes, the computation of shadows and the addition of the faces producing shadows must be previously entered into the initial data base. Since the shadow is a polygon coinciding with a certain polygon (or part of it) in the initial data base and differing only by color, then the shadow generator does not change the organization of the initial base and does not affect the subsequent processing algorithms. In the case in which shadows are generated on the basis of data sorted by priority, the faces producing shadows do not disturb the order of priority, i.e., additional sorting is not required.

Use of the shadow generator makes it possible to significantly increase the number of faces in the data base. However during the time of construction of the imagery

the expenditure for processing additional faces is insignificant and cannot in any way be compared with the expenditure necessary for computing shadows.

2. The next part of the discussion concerns the situation in which the scene is illuminated by a point source which is a finite or infinite (parallel beam) distance away while the location of the light source is considered fixed and invariable. For the case of an extended source, the applicable discussion is similar to that given below. However the laboriousness of the computations which is linked to the need to compute half-shadows is significantly greater.

The problem of shadow generation is closely linked to the problem of removing invisibl surfaces in three-dimensional space. Actually, if the position of the observer is fixed at a point where the light source is located and an orientation is assigned which coincides with the direction of the light beam then only the illuminated faces will be visible and all the faces which are invisible from this point will be inside the shadow. The shadows become visible as the observer moves away from the point at which the light source is located. Thus, in order to determine all shadows it is necessary to set the observer point at the light source and select an observer orientation which coincides with the direction of the light beam. Then, for this setting of the position and orientation of the observer, it is necessary to exclude from consideration the "rear" faces (i.e., the faces for which the normal coincides with the direction of the light beam) and determine all invisible faces. In addition, it is necessary that the surfaces which are invisible from this point be colored corresponding to the colors of the shadow and that they be addd to the initial data base.

3. The method presented below for shadow generation is based on the analysis of visibility pyramids constructed from planes passing through the light sources and polygonal sides included in the data base [3,4].

In the first step, "rear" faces are excluded from consideration [4]. After completion of the first step, let the data base be a set A consisting of M faces: $A = \{\Gamma_i\}$, $i = 1, \ldots, m$.

The second stage consists of the construction for each face Γ_i from the set A of the clipper K_i. The clipper K_i is an infinite truncated pyramid bounded by planes passing through the observation point and the sides of the face Γ_i and also by the plane passing through the face. We note that the face Γ_i partially or completely covers the face Γ_j if the face partially or completely falls in the clipper K_i. It is clear that if the face Γ_j lies completely within the clipper K_i then the clipper K_j lies completely within the clipper K_i. In this case the clipper K_j can be excluded from the set of clippers $K = \{K_i\}$ $i = 1, \ldots, m$, since K_i clearly duplicates all invisible faces in relation to K_j.

In the third stage, for each face Γ_i from the set of faces A the following procedures are carried out: 1) from the set of clippers the clipper K_i is temporarily excluded which corresponds to the face Γ_i ; 2) the face Γ_i is clipped after all the rest.

If at any stage it appears that the face Γ_i is invisible then we remove it from the set of faces A, color it according to its color in the shadow and add it to the initial data base. We remove the corresponding clipper K_i from the set of clippers K and go on to the processing of the next face of set A. In the opposite

case, the face Γ_i is partially or wholly visible and it is necessary to clip it so as to determine the face bounding the shadow. It is then added to the initial data base.

It is easy to see that in the general case it is necessary to subject every face to the clipper procedure M - 1 times. Then the maximum number of clippings will be M(M - 1).

4. In the case in which several sources of illumination are modelled, shadow generation for each source is carried out independently from the rest. Thus the data base is subjected to a shadow generation procedure as many times as there are light sources which must be modelled. When an additional light source is added, repeated shadow generation for previous sources is not necessary.

The proposed algorithm for shadow generation requires a large computation expenditure and is intended for use as a preprocessor in real time systems.

In our case, the removal of invisible surfaces is realized by the programming for real time imagery generation. For graphic systems in which the real time condition is not obligatory, the procedure described above can be used for removing invisible surfaces.

It should be noted that it would be possible to economize a large number of computations if the shadow generation algorithm took into account the silhouettes of objects which throw shadows instead of each face separately.

Bibliography

1. Zinger B.Kh. and E.A. Talnykin. "Predvaritel'naya prostranstvennaya sortirovka-osnova algoritma udaleniya nevidimykh poverkhnostey dlya sistem prioritetnogo tipa" [Preliminary Space Sorting-The Main Algorithm for Removal of Invisible Surfaces for Priority-Type Systems], AVTOMETRIYA, 1983, NO 6.

2. Kovalev A.M. et al. "Ustroystvo dlya vyvoda polutonovykh trekhmernykh ob'yektov ha ekran televiionnogo priyemnika" [Unit for Output of Half-Tone Three-Dimensional Objects on a Television Screen], (Licence NO 834692), BI, 1981, No 20.

3. N'yumen U. and R. Sprull. "Osnovy interaktivnoy mashinoy grafiki" [Principles of Interactive Computer Technology], Moscow, Mir, 1976.

4. Kovalev A.M. and E.A. Talnykin. "Mashinnyy sintez vizual'noy obstanovki" [Computer Synthesis of a Visual Scene], AVTOMETRIYA, 1984, No 4.

12497
CSO: 1863/146

UDC 681.3.06

REALIZATION OF SPATIAL SORTING BY PRIORITIES

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript received 19 May 1984), pp 112-114

[Article by B. Kh. Zinger, Novosibirsk]

[Text] The present paper gives a short description of the realization of an algorithm for preliminary spatial sorting proposed in [1].

Priority sorting is intended to carry out many computations (before display) which are linked to the removal of invisible surfaces. These are carried out independently of the position and orientation of the observer, i.e., it is done once for all possible images of the scene. This type of approach makes it possible to sharply reduce the volume of computations during the image generation time. There is such a significant reduction in computer time expenditure that it becomes possible to generate three-dimensional visual scenes in real time.

As input information for the algorithm under consideration is the data base containing the description of the model of the visual scene. Any object is approximated by a set of plane polygons (faces). Each face is represented by an ordered set of vertices with corresponding coordinates in three-dimensional space.

The process of sorting faces in order to obtain acceptable response speeds and the possibility of processing large volumes of data must have a flexible structural organization and a set of adequate processing methods.

In the general case, for realization of the sorting algorithm the required memory volume may exceed the resources assigned for the task since in order to obtain the required program capacity (possibly in a less effective way) it is necessary to repeatedly use the freed memory regions. List structure makes it possible to organize a complex hierarchical memeory and to solve the problem of processing and representing data with nonstandard organization [2]. Memory is dynamically allocated for each element of the list so that it can be redistributed if necessary.

Thus, in order to effectively process data with complex organization it is necessary, firstly, to use lists, secondly, to make repeated use of freed memory areas and, thirdly, to store data which is temporarily not being used adequately in direct access units [3].

The figure shows a linear list of discriptors of faces each of which represents a structure describing determined attributes of the faces and containing a face indicator. The face, in turn, is a structure including other parameters of the face and an indicator to the list of vertices. The vertices are organized in a linear list.

The face descriptor consists of four fields: counter (Count) which is an integer showing the number of faces enclosing the given face; key field (Ind) which is an integer (number of line in relation matrix); indicator field (Next) which is the indicator the for next element of the list of descriptors; indicator field (Face) which is the face indicator.

Face: this is the memeory structure consisting of five fields- normal (N) is the perpendicular of the face; tag (Tag) is data on processing; Col_2 is the color of the face illuminated by the light source; Col_1 is the color in shadow; Vert is the indicator for the first vertex.

The vertex is described by a structure of four types. In all variants, the indicator field (Link) is used which indicates the next vertex. The description of the type of vertex can vary (for example, the vertex for approximation of a curvilinear object, for the description of a limited texture, etc.). Each vertex is determined by coordinates (x,y,z) and by the texture. The texture of the vertex is assigned by the normal vector (pseudonormal) and by two three-component scalars: color during illumination and color in shadow. Texture information does not participate in the processing but during the dissection of the faces new vertices must be supplied with the correct texture which is linearly distributed on the edges.

In order to describe the relations between faces, a two-dimensional matrix is used M [i,j] (relation matrix). If the face Γ_i covers the face Γ_j ($\Gamma_i > \Gamma_j$ [1]), then M[i,j] = 1.

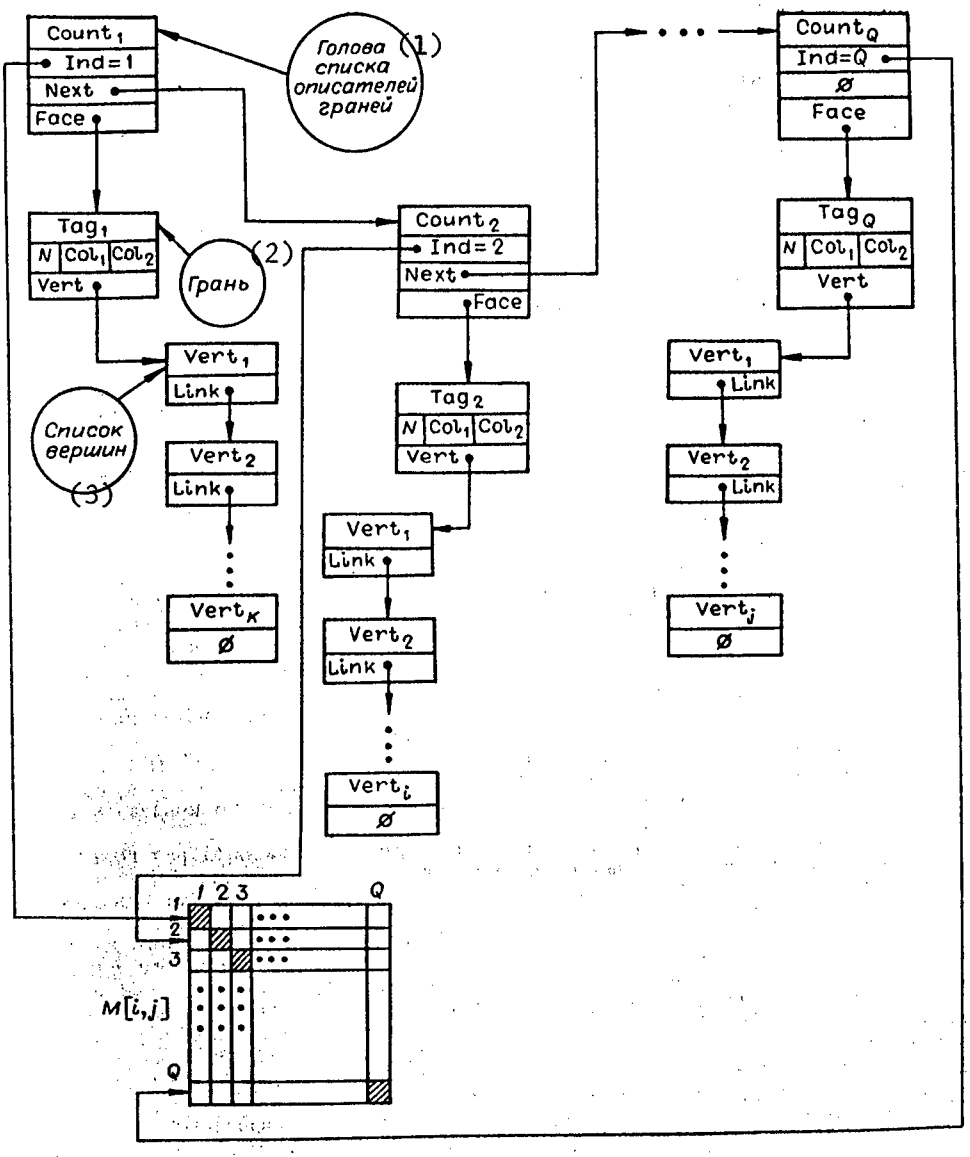
During the sorting of the face list, the main list is broken into sublists which are processed recursively. In order to effectively use the main memory, records which are temporarily not participating in the processing are stored on external carriers. Magnetic discs are used as auxiliary memory carriers and are organized in direct access files [4]. Records are processed in an order which is the reverse of the input order. Below we will call this type of data structure a stack file.

We note that the method being described excludes the formation of "holes", i.e., unused sections of the disc memory.

Sorting Algorithm

The basis of the sorting algorithm is the recursive procedure SORT. In the present case, the recursive processing method is the most natural one since it reflects the special characteristics of the data structure.

The problem consists in the construction of an order of faces such that for any observer position after exclusion of background faces the set of remaining faces can be sorted according to an order of "remoteness" from the observation point (we will suppose that one face is more remote in comparison with another if it covers a second face from the point of view of the observer). We will call a set



Key:

1. Header of face descriptor list
2. Face
3. Vertex list

of faces of this type an ordered set.

From the set of faces $\{\Gamma_1, \ldots, \Gamma_Q\}$, by means of the relations introduced in [1], a subset can be distinguished. If such a subset can be found then all its elements will be written in the file of sorted faces and are excluded from the initial set. After exclusion of the ordered subset the remaining subset does not contain the maximum element.

We will describe the procedure for distinguishing ordered subsets of faces. Its essence is as follows. Each face is compared with all the others, i.e., the face Γ_i, $i = 1, \ldots, n$ is compared with all faces Γ_j, $j = i + 1, \ldots, n$. In order

to do this, two loops are organized (the outer for i and the inner for j). In each step after the comparison of the faces the following procedure is performed: the elements M[i,j] of the i-th line and M[i,j] of the i-th column of the relation matrix are computed. The Count field increases by one in the descriptor of face D_i or D_j (for $\Gamma_i > \Gamma_j$ or $\Gamma_j > \Gamma_i$ respectively).

After completion of the inner loop, according to the Count value in the D_i descriptor field it is possible to determine whether the face Γ_i is a maximum (i.e., can face Γ_i be represented before all other faces). In this case: a) the face Γ_i is written in the file of sorted faces and is removed from the initial list of faces; b) the memory structures corresponding to the face Γ_i which were previously prepared for repeated use are added to the list of free memory sections; c) at the i-th column of the relation matrix all faces Γ_j where j = 1, ..., n, i ≠ j are determined which covered the face Γ_i ; d) for each such face Γ_j ($\Gamma_i > \Gamma_j$) the Count value for the descriptor D_j falls by one and the bit M[j,i] in the relation matrix becomes one.

We note that after the removal of face Γ_i in the face subset $\{\Gamma_1, \ldots, \Gamma_{i-1}\}$ it is possible that maximum elements will appear, i.e., faces for whose descriptors the Count value is zero. Then, for such faces, it is necessary to carry out the same procedures as for face Γ_i .

In other words, at the completion of the inner loop there is, in each case, a recursion procedure which determines over face subset $\{\Gamma_1, \ldots, \Gamma_i\}$ all faces which can be ordered. The outer loop then continues as long as i has not reached the value n.

In order to sort a subset not containing a maximum element, we pass a separation plane through the face which is first in order. This plane divides the entire space (list of faces) into two semispaces (two lists) and the face through which the separation plane passes falls into the semispace where the maximum will be considered to be found. Thus every face lies either entirely in a certain space or is represented in each space. The stack files are then organized according to the division of faces into semispaces. Both lists are prepared for repeated use and are added to the list of free memory sections. A procedure for discriminating ordered subsets is applied to each semispace. All the operations described above are repeated until all the lists have been exhausted and as a result of the operation of the algorithm there are ordered subsets of faces divided by separating planes in both semispaces of which there are ordered subsets.

For a face subset which does not have loops, the determination of priorities reduces to "topological sorting" [5]. In order to separate the ordered set of faces, an algorithm is used which is similar to that described in [5] but with considerable improvements. The algorithm described above for static sorting of faces by priority was realized at the Automation and Electrometry Institute, Academy of Sciences USSR in 1980.

Bibliography

1. Zinger B.Kh. and E.A. Talnykin. "Predvaritel'naya prostranstvennaya sortirovka-osnova algoritma udaleniya nevidimykh poverkhnostey dlya sistem prioritetnogo tipa" [Preliminary Spatial Sorting- Principle of Algorithm for

Removal of Invisible Surfaces for Priority-Type Systems], AVTOMETRIYA, 1983, No 6.

2. Foster Dzh. "Obrabotka spiskov" [List Processing], Moscow, Mir, 1974.

3. Kattsan G. "Operatsionniye sistemy" [Operating Systems], Moscow, Mir, 1976.

4. Dzhadd D.R. " Rabota c faylami" [Working with Files], Moscow, Mir, 1975.

5. Knut D. "Iskusstvo programmirovaniya dlya EVM" [Art of Computer Programming], Moscow, Mir, 1976, Vol. 1.

12497
CSO: 1863/146

UDC 681.327.11

IMPROVING RESOURCES FOR GENERATING AND CONVERTING DYNAMIC SYMBOLIC VIDEO INFORMATION USING MICROPROCESSOR TECHNOLOGY

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (revised manuscript received 18 Apr 84) pp 50-51

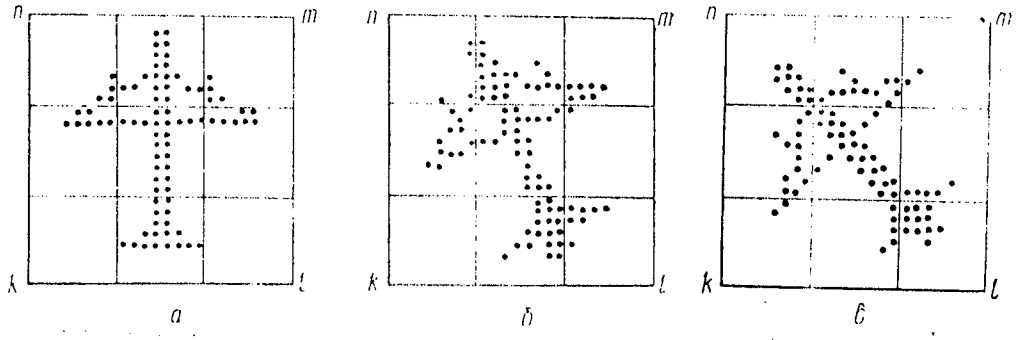
[Article by engineer G. A. Bochenko and Candidate of Technical Sciences M. I. Vasyukhin]

[Text] Designing programs and hardware for generating and converting large volumes of video information is now becoming an extremely important problem. Production of complex dynamic color half-tone images requires high data processing speeds. Development of computer technology in recent years out of high-speed microprocessor elements and inexpensive random-access semiconductor memories and advances in programming have made it possible to create a number of effective video terminal complexes characterized by a high level of intelligence.

But resources produced by Soviet and foreign industry for generating and converting video information are limited in their possibilities for producing complex color dynamic images. Thus a number of modified color video terminal complexes produced in our country are basically capable of producing graphical symbols with shape and dimensions limited to matrixes of 6×8, 8×8 and 8×12 dots. But many applied tasks--for example representing the situation at a marine port and the aerial situation in the vicinity of airports--require the use of symbols of significantly greater complexity and dimensions. Moreover to reduce the probability of operator error and to achieve the complete visual effect of the real motion of an object, symbols must be able to turn about their axes in any direction while concurrently moving across the screen of the video terminal in a two-coordinate plane.

An example of such a symbol containing about 100 points of different colors arranged in a 24×24 matrix can be seen in Figure *a*. An analysis of a number of real applied problems established that 16 different azimuth positions of the image with an angle interval of 22.5° (of which some are shown in the figure) are enough to permit visual perception of change in the direction of motion.

Several methods can be used to produce any of 16 positions different from that shown in Figure *a*.



Representation of symbols: *a*--straight; *b*--turned 22.5°; *c*--turned 45°

The first method requires preliminary storage of all 16 images of the symbol in the memory. In this case (see figure) when a programmable signal generator with an 8×8 dot znakomesto [translation unknown] is used to record a symbol with dimensions of 24×24 dots, 3×3 znakomesto are required. Memorization of all 16 positions requires 16×9=144 znakomesto, which is uneconomical in most cases. This method is characterized by high speed and convenience of data access, but it is often impossible to use owing to the limited volume of the imaging memory.

In the second method the memory retains the image of only one position of the symbol, and the 15 others are generated by a program [1]--that is, all points of the image are recorded. This method is also uneconomical due to the large volume of the working memory occupied by the recoding program, and due to the significant time outlays on fulfilling a large number of operations, which makes it difficult to write the supporting microprograms for industrially produced microprocessors (series K589 for example). Another major shortcoming of this method is that when the symbol turns, its geometric dimensions change (when it turns 45°, its size changes by a factor of $\sqrt{2}$). This circumstance results in a significant increase in the number of wrong reactions by the operator when the screen is saturated by moving symbols.

The method proposed here for generating and converting symbol images consists of the following.

Only three basic images of the symbol are stored in the memory: straight, and turned 22.5 and 45°. The rest of the 13 images are obtained by conversion of these three images, which is done by using different procedures to read the matrixes of video information recorded in the memory. For example to obtain an image turned 90, 180 and 270°, the initial matrix $\begin{bmatrix} n & m \\ k & l \end{bmatrix}$ must be read such that the lines and columns would switch places so that the matrix would appear as $\begin{bmatrix} k & n \\ l & m \end{bmatrix}$, $\begin{bmatrix} k & l \\ n & m \end{bmatrix}$ and $\begin{bmatrix} m & l \\ n & k \end{bmatrix}$. Converting the three basic images of the symbol in this fashion, we obtain nine other images.

The remaining four images of the symbol, turned 67.5, 157.5, 247.5 and 337.5°, can be obtained by converting the matrix shown in Figure *b* into the following matrixes respectively:

$$\begin{bmatrix} n & k \\ m & l \end{bmatrix}, \begin{bmatrix} k & l \\ n & m \end{bmatrix}, \begin{bmatrix} l & m \\ k & n \end{bmatrix} \text{ и } \begin{bmatrix} m & n \\ l & k \end{bmatrix}.$$

The method described here is sufficiently simple to achieve with series K589 microprocessor elements. Expansion of the functional possibilities of a K589IK01 programmed control block by using multiplexers makes it possible to significantly simplify the microprograms and accelerate the process of forming one of 16 images on the basis of the appropriate characteristic [2]. For example when series K589 microprocessor elements are used, it takes about $25 \cdot 10^3$ nanoseconds to form one symbol with the required turning angle, while a programmed turn of a symbol using MIKROKOD language and existing video terminals requires $3 \cdot 10^5$ nanoseconds.

In contrast to the known methods of turning images of complex symbols, this method insures a significant savings in video terminal memory, acceleration of the process of forming dynamic images, reduction of program volume and, in addition, fuller geometric identity of turned images, which in turn promotes a decrease in the number of wrong reactions by the operator.

The proposed method can be used to represent the situation in the vicinity of airports and sea ports on displays and in automated transportation and passenger traffic control systems.

BIBLIOGRAPHY

1. Ahuja, D. V., "Geometry for Screen Design and Display," VTsP, Kiyevskaya redaktsiya, No KV-2229, 29 pages; IBM SYSTEMS JOURNAL, Vol 4, No 3, 1968, pp 188-205.

2. Berezenko, A. N., Koryagin, L. N. and Nazar'yan, A. P., "Mikroprotsessornyye komplekty povyshennogo bystrodeystviya" [Higher-Speed Microprocessor Outfits], Moscow, Radio i svyaz', 1981, 168 pages.

11004
CSO: 1863/259

UDC 681.3.069

MICROPROCESSOR REALIZATION OF A DENDRITIC RECOGNITION ALGORITHM

Leningrad IZVESTIYA VYSSHIKH UCHEBNYKH ZAVEDENIY: PRIBOROSTROYENIYE in Russian No 3, Mar 85 (manuscript received 23 Nov 81) pp 38-41

[Article by I. F. Klistorin, Yu. A. Pushnyak and G. Ya. Shevchenko, Kishinev Polytechnical Institute imeni S. Lazo, recommended by the Department of Design and Production of Electronic Equipment]

[Text] The authors examine the possibility of microprocessor realization of algorithm with a dendritic structure which can be used in problems of diagnostics, pattern recognition and control. Basic stages of the development of a specialized recognition device are described. Realization of a recognition algorithm is explained on an example.

Algorithms with a dendritic structure are used widely in problems of pattern recognition, diagnostics, decision-making and control [1]. As a rule, their construction requires considerable computing resources which can be reached only with the use of large computers, while the use of large computers in the operating conditions of the obtained algorithm is not always possible technically and is not expedient economically. Evidently, it is possible to find a way out by using microprocessors (MP) at the stage of operation programmed for solving specific problems in accordance with algorithms worked out in advance on large computers. In view of the generally known advantages of MP (small dimensions, low costs, possibility of flexible readjustment of the functioning algorithm, real-time operation, etc), this approach has obvious advantages and makes it possible to create small-size systems with parameters unattainable before [2].

This work examines microprocessor realization of a dendritic-structure algorithm for pattern recognition and diagnostics worked out in advance on a large computer. Solution of a recognition problem consists in constructing a solution rule (RP), i.e., a rule which establishes correspondence between one of the possible solutions and the values of the characteristics describing the object or situation being recognized [3].

Very often an RP is constructed on the basis of information presented in the form of a learning sampling (OV). One of the effective methods of RP representation is a graphic method, such as recognition tree (RD). In RD, finite and nonfinite nodes are distinguished. In nonfinite nodes, characteristics from among those presented in OV are recorded, and in finite nodes -- numbers

of classes or values of functions F(X), where $X = X_1, X_2, \ldots, X_n(1)$ -- the object (set of values of characteristics), and X_i -- the i-th characteristic (parameter, index) assuming discrete values.

The entire process of work with RD in constructing a specialized recognition device is divided into four stages.

1. Learning, during which the construction of the RD structure takes place. It is possible to construct RD in different ways, and in the majority of cases the number of variables (characteristics) operating in this tree is substantially smaller than the initial number of variables in OV, which is a result of the selection of the most informative characteristics in RD. In the course of the learning stage, an appropriate characteristic (X_i) is determined at each next step with the aid of particular criteria and placed in the next nonfinite node of the RD path being examined. RD is constructed until the numbers of classes represented in the given OV are found at a certain step, after which they are placed at finite RD nodes.

The result of the learning stage is a certain dendritic structure corresponding to the particular problem being examined.

2. Checking, in the course of which the correspondence of the RD structure to the examined problem is checked with the aid of control sampling (KV). If the user is satisfied with the recognition reliability of KV, then the process of constructing and checking RD is considered completed; if he is not satisfied, then it is necessary to return to the first stage and make appropriate corrections in the RD structure.

3. Permanent memory (PZU) programming. As has already been mentioned above, the first and second stages are done with the use of the resources of large computers, when a particular structure corresponding to a particular problem is obtained and worked out in a final form. It is realized in the form of a program on an Assembler in application to a specific MP. After that, the microprocessor PZU is programmed by one of the possible methods and the system is ready for operation, i.e., for classification and diagnostics.

4. Operation. This is a "working" stage during which the microprocessor classifies the samplings arriving for recognition in accordance with the algorithm stored in its PZU. Below is a verbal description of this algorithm:

a) reception of sampling $X=X_1, X_2, \ldots, X_n$ arriving for recognition;

b) extraction of the characteristic X_i corresponding to the next RD node;

c) analysis of the value of the X_i characteristic;

d) depending on the value of X_i (0, 1, ..., k), movement along one of the verges (0, 1, ..., k) to the next node;

e) reaching the next node. Analysis of the node for its finiteness. If the node is nonfinite, then the number of the characteristic

is marked and jump to (b) is performed; if the node is finite -- jump to (f);

f) readout of the value of F_j corresponding to the finite node reached. The object (sampling X) belongs to the class with the number j. Stop.

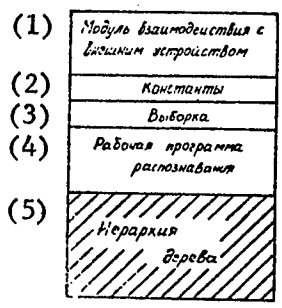


Figure 1. Memory chart

Key: 1. Module of interaction with a peripheral unit
2. Constants
3. Sampling
4. Running recognition program
5. Tree hierarchy

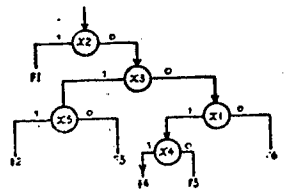


Figure 2. Graphic example of the hierarchic structure of RD

As can be seen, the pattern recognition algorithm consists in moving along RD to a particular finite node proceeding from successive analysis of characteristics (indexes) of the sampling. Such an algorithm is universal, only the data on the hierarchic structure of RD change from one case to another. This is reflected in the memory chart (Figure 1), where the changing fragment of the program is shaded. It is practical to place the fragment "Constants" and "Sampling" in working registers of MP because of their small dimensions and frequent use, and to place the remaining fragments in OZU or PZU of the system. To illustrate this, let us examine a simple example. Let us assume that, as a result of the learning stage, an RD with a binary structure was obtained (Figure 2). As can be seen, RD has six finite nodes (F1...F6) and 5 nonfinite nodes (X1...X5). Each nonfinite node has only one incoming verge and two outgoing verges: verge "0" and verge "1". The RD input can receive samplings of the type of (1), where X_i is a binary variable, while variables with $i > 5$ are unessential (as can be seen from the RD structure) and will be omitted hereafter.

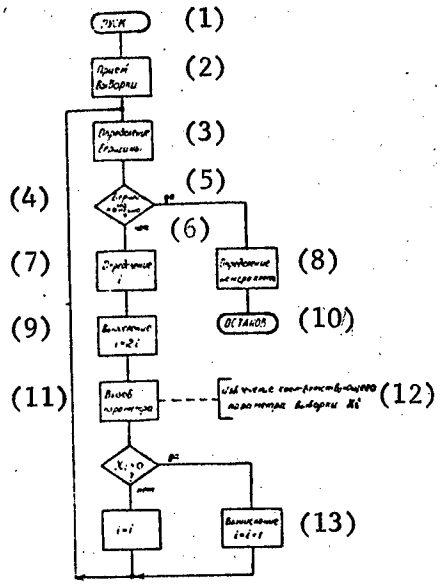


Figure 3. Recognition algorithm by means of passage along the RD

Key: 1. Start
2. Sampling reception
3. Node determination
4. Is the node finite?
5. Yes
6. No
7. Determination of i
8. Determination of the number of the class
9. Calculation
10. Stop
11. Parameter request
12. Extraction of the component parameter of sampling X_i
13. Calculation

Вершина Node	X2	F1	X3	X5	X1	F2	F3	X4	F6	F4	F5
i	1	—	2	3	4	—	—	5	—	—	—

The table gives a recording of the same structure (fragment of the "tree hierarchy" program) formed on the "top-down" and "left-to-right" principle. As can be seen only nonfinite nodes can have the serial number i. This number i is used in the recognition algorithm in searching for the next RD node. The recognition algorithm (Figure 3) is sufficiently simple and does not need any explanation. Let us only mention that it can be developed with a set of instructions of practically any MP. As a rule, it is developed with the use of shift, content-analysis and jump instructions. As an example, arrows in Figure 2 show the sequence of the passage of the tree when a sampling of the type of X1=1, X2=0, X3=0, Z4=1, Z5=1 arrives for recognition.

In conclusion, let us note that the recognition program was repeatedly tested on a minicomputer and proved to be workable and effective. We believe that the above approach to the realization of algorithmswith a dendritic structure can be used for solving a wide range of specific problems in the area of control, diagnostics, pattern recognition, etc.

Bibliography

1. Markovich, Z. P. "The Use of Graphic Models for Solving Problems of Technical Diagnostics", in the book "Kibernetika i diagnostika" [Cybernetics and Diagnostics], Riga, Zinatne, 1968, No 2, pp 49-62.

2. Vasenkov, A. A., editor. "Mikroprotsessornyye BIS i mikro-EVM. Postroyeniye i primeneniye" [Microprocessor Large-Scale Integrated Circuits and Microcomputers. Their Construction and Uses], Sovetskoye radio, 1980.

3. Academician Glushkov, V. M., editor. "Slovar' po kibernetike" [Cybernetics Dictionary], Main Editorial Office of USE [Ukrainian Soviet Encyclopedia], 1979.

10233
CSO: 1863/280

UDC 621.372.54

QUANTIZING AND ADDITIVE NOISE STABILITY OF ALGORITHM FOR REDUCING SIGNAL HISTOGRAM TO ASSIGNED VALUES

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript received 25 January 1983), pp 115-118

[Article by A.N. Ushakov and L.P. Yaroslavskiy, Moscow]

[Text] The transformation of signal distribution histograms to assigned values is used for correcting nonlinear distortions of imagery and holograms [1], standardization of imagery according to brightness distribution and for the preparation of imagery [2]. In [1] a simple algorithm is described which is based on a procedure for histogram equalization, i.e., on a transform of the signal with an observed frequency distribution into a signal with a uniform distribution in an assigned range. It consists of the following.

1. On the basis of the histogram $h_H(w)$ of the observed signal w to be transformed, a table of transforms $w \to u$ is established according to the formula

$$u(w) = \operatorname{int}\left[(M-1)\left(\sum_{k=0}^{w} h_H(k) - h_H(0)\right)\Big/\left(1-h_H(0)\right)\right],$$
$$w = 0, 1, \ldots, M-1,$$

where M is the number of quantization levels of signal w. This transform makes the distribution of signal values uniform.

2. For the required histogram $h_T(\hat{w})$ an analogous equivalence table is constructed:

$$\hat{u}(w) = \operatorname{int}\left[(M-1)\left(\sum_{k=0}^{w} h_T(k) - h_T(0)\right)\Big/\left(1-h_T(0)\right)\right].$$

3. By permutation of the inputs and outputs of the table $\hat{u}(w)$ a table $\tilde{w}(\hat{u})$ is defined which transforms the signal with uniform distribution into a signal with the assigned distribution $h_T(\hat{u})$.

4. From the two tables u(w) and $\tilde{w}(\hat{u})$ a single table is formed $\tilde{w}(\hat{u} = u(w))$ which gives the desired transform.

It is clear that for an infinitely large number of levels of quantization of the initial signal w that this algorithm precisely gives the required frequency distribution of the transformed signal. In the case of finite quantization levels this is

not the case and it is interesting to study the effect of the number of quantization levels on the precision of the tranform. In addition, the signal to be corrected is usually observed against a background of, in most cases, additive noise which can sharply increase as a result of the transform.

In the present paper, results are given for research on the influence of the effects on quantization of the signal and of additive fluctuation noise on the precision of the transform which were obtained by digital modeling for an example of the processing of a narrow-band simulated signal (interferograms) (Fig. 1, a).The histogram of signal values, its energetic spectrum and the signal section are shown in Fig. 1, b-d. The initial signal is subjected to nonlinear distortion The

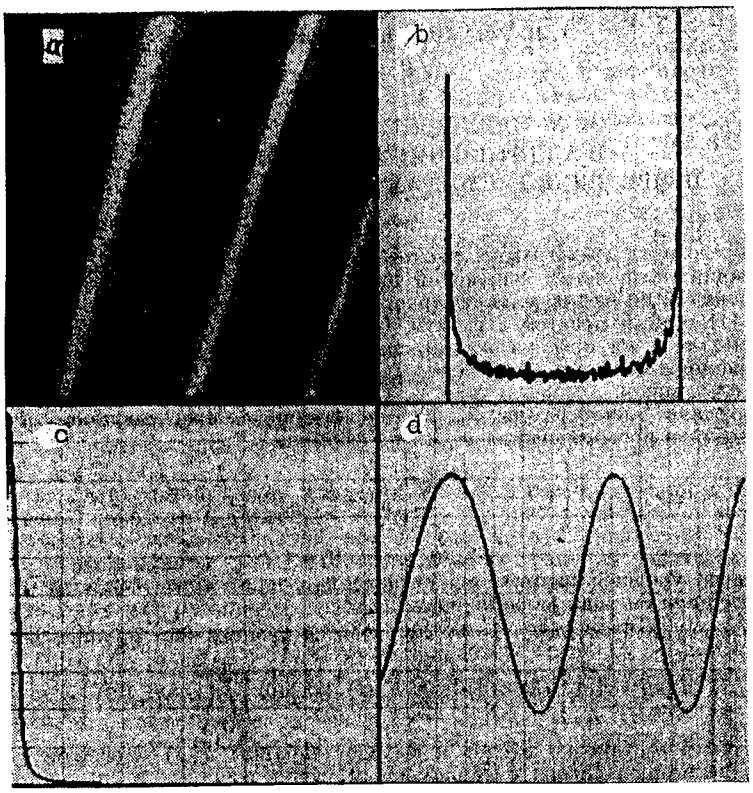


Figure 1

The relative difference of the energetic spectra of the initial and distorted signals (relative root-mean-square error) amounted to 20%. The distorted signal was quantized at 64, 30, 16, 8 and 4 levels. Fig. 2 shows sections of the distorted signal and its histograms for 30, 16, 8 and 4 quantization levels (two left columns) as well as sections of the signal and histograms after correction of distortions (two right columns). The figure shows that correction of the shape of the histograms shows itself in the bunching of the readings on the left end.

The relative root-mean-square values for nonlinear distortions at the input (NL) and output (KR) of the distortion corrector in relation to the number of signal quantization levels are shown in the graph (Fig. 3). From the graph it follows

that even for 4 levels of quantization the algorithm under study makes it possible to considerably decrease the level of nonlinear distortions.

For modeling additive noise a Gaussian white noise transducer was used with independent counts and it was supposed that the corrector does not use information on the presence of noise. Fig. 4 shows the results of the experiment with correction of distortion in the presence of additive noise in the distorted signal after quantization into 30 levels and application of noise consisting of 25, 50, 100 and 200% of signal power (S = 4N, S = 2N, S = N, S = 0.5N). In the two right columns are the results of correction which show that for a high noise level there would seem to be a significant "correction" of the signal histogram. However this correction amounts only to a change in the noise histogram.

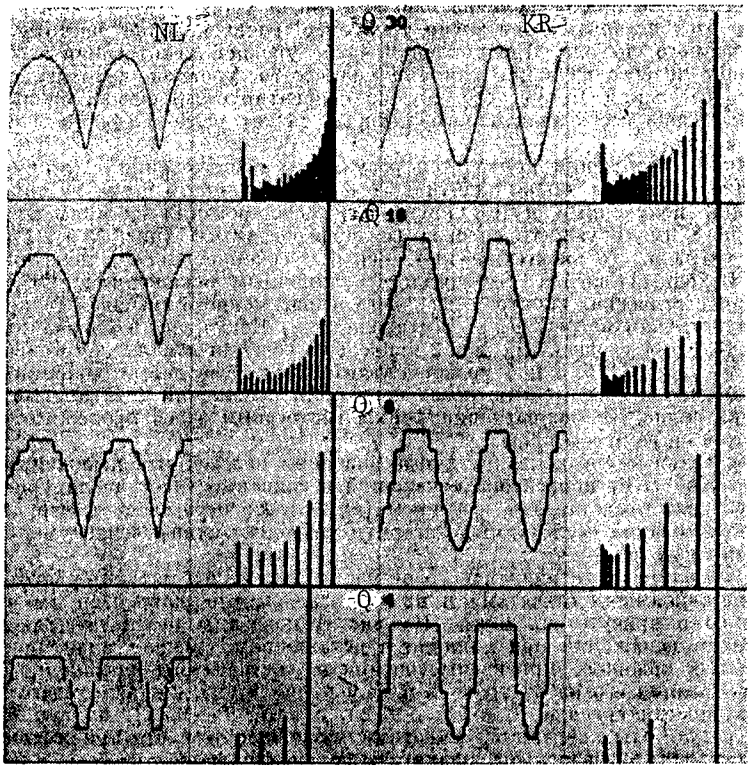


Figure 2

Fig. 5 shows the dependence of the relative level of nonlinear distrotion at the input and output of the corrector on the ratio of additive noise power to signal power for different quantization levels of the distorted signal. It is clear that the algorithm becomes ineffective only for noise amounting to 400% of the signal power

It should be noted that the "cost" for the reduction of nonlinear distortions is the increase in noise power at the corrector output. Experiments showed that for 8-64 levels of quantization and variation in the signal/noise ratio from 0 to 4 as a result of the correction the noise power increased by a factor of 2.

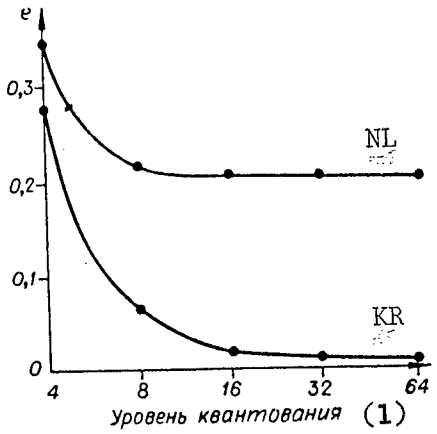


Figure 3

Key:

1. Quantization level

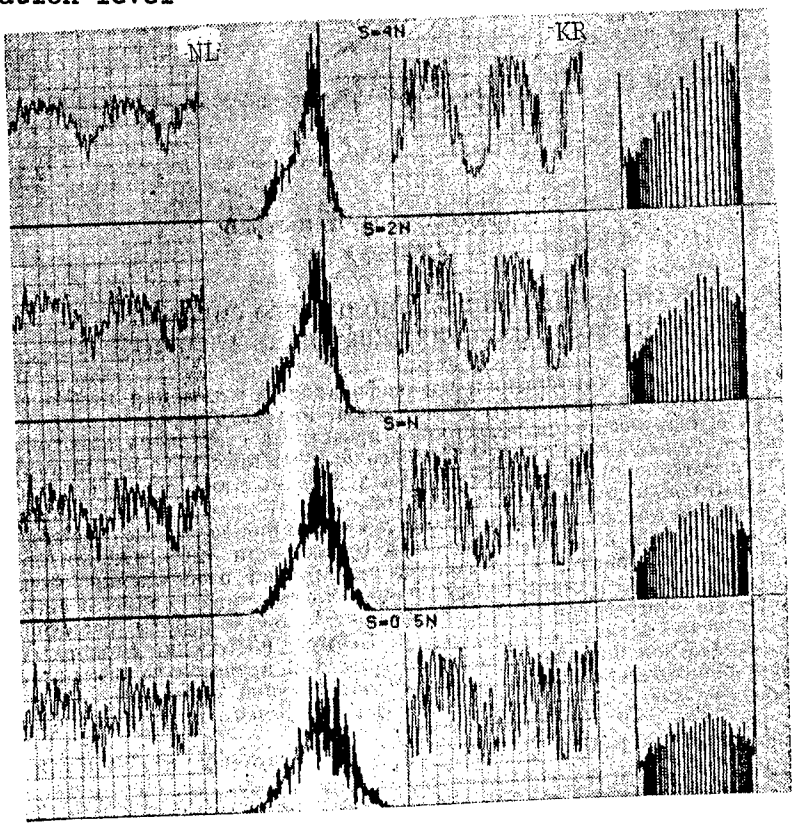


Figure 4

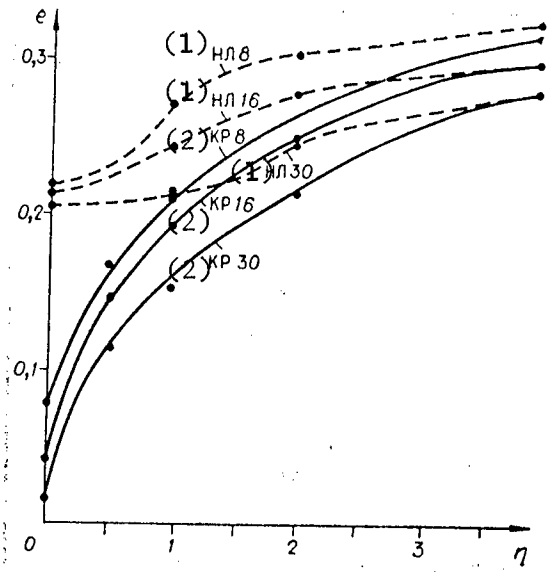


Figure 5

Key:

1. NL
2. KR

From the analysis of the experimental results the conclusion can be drawn that the nonlinear distortion correction algorithm is quite stable in relation to the number of quantization levels and the noise level.

Bibliography

1. Yaroslavskaya L.P. and N.S. Merzlakov. "Tsifrovaya golografiya" [Digital Holography], Moscow, Nauka, 1982.

2. Yaroslavskiy L.P. "Vvedeniye v tsifrovuyu obrabotku izobrazheniy" [Introduction to Digital Processing of Imagery], Moscow, Sovetskoye Radio, 1979.

12497
CSO: 1863/146

UDC 621.382

A PRIORI EVALUATIONS OF USEFUL SIGNAL FOR MORPHOLOGICAL SOLUTION ALGORITHMS

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984 (manuscript submitted 18 July 1983; received after final revision 25 November 1983), pp 118-120

[Article by A.G. Yermolayev and Yu.P. Pyt'yev, Moscow]

[Text] In imagery search, coincidence, identification and discrimination problems, methods of morphological analysis [1,2] make it possible to obtain good results for imagery with quantized brightness values at the cost of a very high noise-suppression requirement. The solution algorithms based on these methods minimize the probability of a posteriori error in the solution for equal a priori probabilities in problems with additive noise* and their optimization [3] makes it possible to plan the processing of imagery with maximum response speed for assigned limits on the signal/noise ratio and solution quality. However, if the noise intensity, as is usually the case, is known from the rating of the imagery recording unit (or can simply be evaluated for an equal field of view) then the value of the useful signal, without knowledge of which it is impossible to plan optimal operation of the solution facilities, depends essentially on the "context". Therefore the establishment of a priori evaluations of the useful signal for a broad class of imagery is an important stage in the design of specialized equipment for in-line processing.

Let us consider the problem of a search for a fragment whose shape is not more complex than the shape of the standard image $\varphi(x) = \sum_{i=1}^{k} \beta_i \chi_i(x)$ [1] in a certain image $f(x) = \sum_{i=1}^{m} \alpha_i \chi_i(x)$, which is set in the field of view X. It is initially supposed that such a fragment exists ($k < m$). Here and below, $\chi_i(x)$ are the indicator functions for the subsets of the same brightness A_i of the field of view X. Then the value of the useful signal [3] for the morphological solution rule for the minimum of the norm $\|(I-P_\varphi)(f(x+y)+v(x+y))\|$ is $\delta(y) = \|(P_\varphi - P_f)f(x+y)\|$ where P_φ is the operator for the shape of the standard fragment, I is the identity operator, v(x) is the noise, y is the parameter for the shift of the standard relative to the imagery during the search. By making use of the energetic norm it is possible to show that

$$\delta(y) = \sum_{i=1}^{k} \left(\sum_{j=1}^{m} \alpha_j^2 \mu_{ij}(y) - \left(\sum_{j=1}^{m} \alpha_j \mu_{ij}(y) \right)^2 \Big/ \mu_i \right) \qquad (1)$$

*For a sufficiently loose limitation, i.e., the noise density probability p(z) should be a monotonically decreasing function of $\|z\|$.

or, after transformation of the expression in the inner parenthesis

$$\delta(y) = \sum_{i=1}^{k} \sum_{j>n} (\alpha_{ij} - \alpha_{in})^2 \mu_{ij}(y)\, \mu_{in}(y)/\mu_i, \qquad (2)$$

where $\mu_{ij}(y) = (\chi_i(x), \chi_j(x+y))$, $\mu_i = \|\chi_i(x)\|$, and α_{ij} is the brightness of the imagery falling into the subset of the standard with the number i (j = 1, ..., m). Thus the useful signal depends on the squares of the differences of brightness in neighboring sets.

In problems of optimization of morphological algorithms an evaluation from below for the useful signal is of interest. It is not difficult to convince oneself that the evaluation (1) from below is zero. Therefore we will consider below the evaluations δ (y) for two different situations: 1) the value y is large (the standard is not very similar to the observed fragment), 2) y is quite small (the standard is close to the desired fragment). We will formulate the criterion for smallness below. In the first case, it is possible to assign statistical characteristics to the functions $\mu_{ij}(y)$ by supposing that y is a random value (i.e., the search for the fragment is carried out by the random step method). For the value δ(y) we obtain evaluations of the mathematical expectation and variance which are correct for the whole class of statistically similar imagery. In the second case, we construct an expansion of the signal δ (y) which is close to zero (since $\delta(0) = 0$).

For sufficiently large random shifts it is more natural to suppose that the combined distribution of values like μ_i and μ_{ij} (y) is uniform. This supposition covers the case in which large and small areas of the same brightness are in the field of view equally frequently (the histogram of the imagery is "sufficiently homogeneous"). We will describe a class of imagery with this characteristic:

1) we will use the Poisson process with the parameter λ and events occurring at moments $0, \tau_1, \ldots, \tau_m, \ldots$;
2) the number m is such that $\tau_m \geq$ mes X, while $\tau_{m-1} <$ mes X;
3) on X we discriminate the random areas $A_i \subset X$ so that $A_i \cap A_j \neq \emptyset$, $i \neq j$ and mes $A_i = \mu_i = \tau_i - \tau_{i-1}$, $i = 1, \ldots, m-1$ ($\tau_0 = 0$), while $\mu_m =$ mes $X - \tau_{m-1}$;
4) to the discriminated areas we assign the brightness values α_i from a certain set $\{\alpha_i\}$.

From the characteristics of the Poisson process it follows that the combined distribution of τ_i like that of μ_i is uniform [4] in any interval $(a,b) \subset (0,$ mes X). For the previously set image the value $\hat{\lambda} = m/\text{mes } X$ is an empirically valid evaluation of the parameter λ while the question as to whether this image belongs to the described class can be decided, for example, by the Vatson criterion [5].

On these assumptions it is easy to obtain the following evaluations:

$$E\delta(y) \geq \alpha_0^2 \left(\sum_{i=1}^{k} (1 - e^{-\lambda_i}) \frac{1}{\mu_i \lambda^2} - \frac{1}{2\lambda} \right) = \delta_0 \qquad (3)$$

and

$$D\delta(y) = \sum_{i=1}^{h} \mu_i^2 D_i,$$

where

$$D_i \leqslant (\Delta\alpha)^4 \left[1/4 - 1/\lambda_i + 2/\lambda_i^2 - 6/\lambda_i^4 + e^{-\lambda_i}\left(1/\lambda_i^2 + 6/\lambda_i^3 + 6/\lambda_i^4\right)\right] - \\ - \alpha_0^4 \left(1/2 - 1/\lambda_i + 1/\lambda_i^2 - e^{-\lambda_i}/\lambda_i^2\right), \quad (4)$$

$$\alpha_0^2 = \min_{i \neq j} (\alpha_i - \alpha_j)^2, \quad \Delta\alpha = \max (\alpha_i - \alpha_j), \quad \lambda_i = \mu_i \lambda.$$

The evaluations (3) and (4) become precise equalities for images with a constant brightness jump at the limit. These relations occur in both search problems and imagery identification problems.

In order to expand $\delta(y)$ in the neighborhood of zero we use formula (2). Here the values $\mu_{ij}(y)$ are measures of the intersection of the area A_i with neighboring areas for a shift for y. in this case,

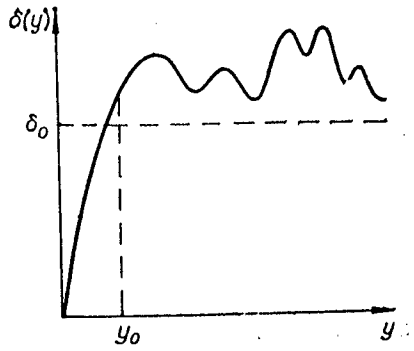
$$\mu_{ii}(y) = \mu_i - \sum_{i \neq j} \mu_{ij}(y),$$

while $\mu_{ij}(y) = Cy$ for sufficiently small y and therefore the expression for $\delta(y)$ has the form of a quadratic relation of the type $\delta(y) = ay - by^2$. It is possible to show that the monotonic area of $\delta(y)$ is determined by the condition

$$y \leqslant (1/2)\mu_{\min}/d_{\min} = y_0, \quad (5)$$

where d_{min} is the diameter of the area which is proportionately minimal in a direction perpendicular to the shift. Formula (5) is also a criterion for the smallness of the value y. Larger values of y are therefore set by the condition $y > y_{max}/d_{max}$.

Use of the evaluations (3) and (4) make it possible to increase the speed of response and simultaneously to reduce the memory of the solution equipment by 1-2 orders by selecting the optimal work mode [3]. The figure shows the relation $\delta(y)$ for the modeled imagery in the coincidence problem.



Bibliography

1. Pyt'yev Yu.P. "Morfologicheskiy analiz izobrazheniy" [Morphological Analysis of Imagery], DAN, 1975, Vol 224, No 6.

2. Pyt'yev Yu.P., V.V. Tekin and Ye.N. Terent'yev. "Sravnitelnyy analiz nekotorykh reshayushchykh algoritmov ha TsVM" [Comparative Analysis of Several Solution Algorithms for Digital Computer], Izv. AN SSSR, Ser. Tekhn. kibernetika, 1976, No 4.

3. Yermolayev A.G. "Optimizatsiya morfologicheskikh algoritmov identifikatsiya izobrazheniy" [Optimization of Morphological Algorithms for Imagery Identification], AVTOMETRIYA, 1982, No 4.

4. Kendall M. and P. Moran. "Geometricheskiye veroyatnosti" [Geometrical Probabilities], Moscow, Nauka, 1972.

5. Martynov G.V. "Kriterii omega-kvadrat" [Omega-Square Criteria], Moscow, Nauka, 1978.

12497
CSO: 1863/146

UDC 621.391:681.325.5.001

HIGH-SPEED RECEIVERS OF MICROPROCESSOR SIGNAL CONVERSION DEVICES

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (revised manuscript received 5 Apr 84) pp 55-57

[Article by candidates of technical sciences L. I. Sulin and V. V. Romanov]

[Test] The main effectiveness indicator of signal conversion devices intended for frequency-limited communication channels is the relative speed of data transmission, given certain limitations on the reliability of the received information.

Raising the relative speed of data transmission requires use of combined modulation methods in combination with effective processing at the receiving end. As a rule this means an increase in the algorithmic complexity of the data processing.

Growth in complexity of algorithms results in the need for developing complex specialized structures with rigid logical control, or synthesis of structures operating on the basis of microprogram control. The latter approach seems more preferable because use of BIS [large integrated circuit] microprocessor complexes that are series-produced by industry means that development time can be decreased, and the cost of the signal conversion devices is relatively low.

On the other hand when a signal conversion device is built on the basis of microprocessor systems, the unique features of the microprocessor systems must be accounted for (in the selection or development of the signal conversion algorithms).

First, it would be desirable for the algorithm to consist only (or predominantly) of simple operations such as addition and shift--ones in relation to which microprocessor systems exhibit the highest productivity.

Second, the algorithm must provide the possibility for pipeline data processing if owing to high labor-intensiveness it cannot be achieved by a single-processor system.

Third, the algorithm must be minimally sensitive to rounding-off and sampling errors. In this case minimum word length of buses and operating devices is achieved.

Analysis of the available algorithms showed that they do not fully satisfy the requirements.

A new algorithm for receiving discrete signals oriented on microprocessor-type signal conversion devices is proposed here.

This algorithm is the result of formalizing the principles of designing receivers equipped with feedback [1,2].

We will consider a channel with constant parameters, intersymbol interference and additive Gaussian noise, the properties of which are reflected in its low frequency model:

$$y_n = \sum_{m=0}^{M-1} h_{n-m} x_n + e_n, \; n = \overline{1, N},$$

where y_n--samples of the signal's complex envelope at the receiver input; h_{n-m}--samples of the complex envelope of the channel's pulse characteristic, x_n--samples of the complex envelope of the signal at the transmitter output; e_n--samples of the complex envelope of additive noise; M--number of samples of the communication channel's pulse characteristic accounted for. This system of equations can be written in the following matrix form:

$$y = Hx + e. \qquad (1)$$

Here the number of equations N is greater than the number of unknowns ($K=N-M+1$); therefore matrix equation (1) can be solved by the least square method. As a result we get [3]:

$$H^T H = H^T y. \qquad (2)$$

If this equation were to be solved without limitations imposed upon the set of permissible values of vector x, then the obtained solution would possess properties similar to those of solutions obtained by means of linear signal filtration--that is, the potential interference resistance of reception of discrete signals would not be insured. On the other hand there is no exact solution to equation (2) in relation to a limited set of permissible values of vector x_j, and therefore we need a criterion that would make it possible to select, from among all variants of messages that can be transmitted, the best one--one with the lowest probability of erroneous reception. The minimum root-mean-square error is such a criterion (in the case where the probabilities of appearance of individual messages are unknown), but its use requires application of algorithms characterized by high labor-intensiveness.

In order to reduce the labor-intensiveness of solving the equation in relation to the set of permissible values of vector x_j, we suggest using the minimax criterion, in accordance with which a solution x_m insuring minimization of the maximum absolute error of closure of system (1) is sought:

$$\max_l (F_l | y - Hx_i |) = \min_{x_i},$$

where F_j is a vector-line, all the elements of which are equal to zero except for $f_j=1$.

This criterion possesses an important merit: It is constructive, and it may be used to solve systems of linear equations involving a finite set of solution vectors. Matrix equation (1) is solved by a method similar to the relaxation method, differing in that a new approximation is selected at each iteration from the set of all signal values that can be transmitted.

The procedure for finding the solution can be written formally in the form of recurrent relationships.

The initial state: $x^0=x_1$; $\Delta_0(y)=y$.

The iterative process:

$$j:|F_j\Delta_i(y)|=\max_j;$$

$$x^{i+1}=\Xi(x^i+F_j\Delta_i(y));$$

$$\Delta_{i+1}(y)=\Delta_i(y)-H(x^{i+1}-x^i),$$

where x^i--permissible solution obtained in iteration i; Ξ--quantization operation defined as $\Xi(x^i+Q)=x^{i+1}$, if $||x^i+Q-x^{i+1}_k||=\min$ and $x^i\neq x^{i+1}_k$.

In order to prevent cycling of the process of finding new solutions, a rule is introduced according to which each component of the solution vector may vary by a limited number of times d, and this fact is fixed in vector B, the initial state of which is adopted as zero.

The algorithm described here effects reception of the message as a whole. Generalizing this algorithm, it would be suitable to form the solution vector in relation to C symbols ($C\geqslant L$), for which all elements of vector y containing a reaction to these symbols are received at the given moment in time. This makes it possible to find L initial estimates of the components of vector x and correct C-L other components at the time of reception of each block of symbols. Correction of previous solutions makes it possible to weaken the influence of subsequent symbols upon intersymbol interference. If information is transmitted in blocks of $L_{\text{ЛБ}}$ symbols, and $L=L_{\text{ЛБ}}$, then this algorithm effects transmission in general; but if $L=1$, reception proceeds on a symbol-by-symbol basis.

The flowchart of the described minimax search algorithm for reception of one block of symbols is shown in Figure 1. Solution vector x_m and a discrepancy vector, both of which are shifted by L positions prior to reception of the next block of samples, are the result of the algorithm's work.

The labor-intensiveness of this algorithm is the product of blocks 2-5. In this case the operations of determining the maximum absolute discrepancy and its number (blocks 5 and 2) are carried out on the basis of two-place comparison operations. In block 3 the quantization operation, which proceeds in the presence of quadrature amplitude modulation, reduces to rounding off

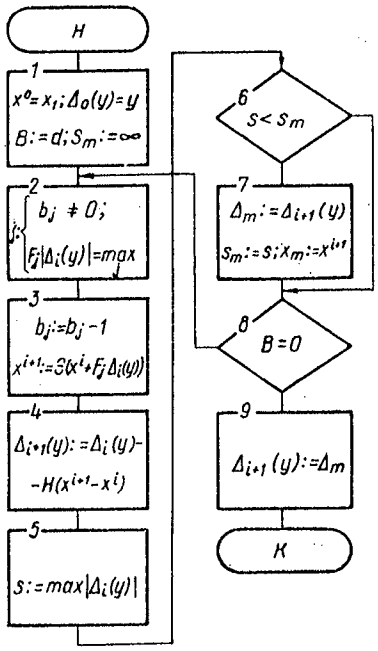


Figure 1. Flowchart of the Algorithm for Reception of Discrete Signals

the operand to a whole-number value belonging within the constellation of signals, while the operation of multiplying vector F_j by vector $\Delta_i(y)$ corresponds to determining element j of vector $\Delta_i(y)$. Products of the form $H(x^{i+1}-x^i)$ (block 4) are calculated beforehand and stored in the working memory for each permissible value of the difference $(x^{i+1}-x^i)$.

Thus the minimax search algorithm is effected completely through the operations of addition, subtraction, comparison, transfer and shift.

On the whole the labor-intensiveness of the minimax search algorithm (when L=10, C=32) is 250 operations per bit of received information, if it is assessed on the basis of addition operations. The labor-intensiveness of a harmonic corrector with 32 coefficients is 140 operations per bit under the same conditions.

Given a signal conversion device for standard tone frequency channels and data transmission speeds of 9,600-19,200 bits per second, the minimax search algorithm would require a microprocessor system with a productivity of 2-5 million operations per second; series 589, 1802 and 1804 higher-speed BIS microprocessor complexes would best be used to achieve this productivity. Considering that the cycle of a one-address addition instruction requires 1-2 μsec in such systems, we can assume that to run the proposed algorithm in modems for a tone frequency channel, three to five microprocessor systems and 2-4 kilobytes of a sectionalized common memory would be required. As a rule each of the microprocessor systems would include microprocessor sections, a microinstruction address control circuit, a programmable permanent memory and auxiliary connecting and switching circuits.

It would be suitable to organize parallel work of the processors when performing operators of each of blocks 1, 3, 4 and 7.

The word length of the microprocessor system is selected such that quantization and rounding off noise would reduce the interference resistance of the algorithm by not more than tenths of a decibel. Given a transmission speed of 9,600 bits per second, this can be achieved with an eight-bit operand representation. For a speed of 19,200 bits per second, the word length of the microprocessor systems should be increased to 12.

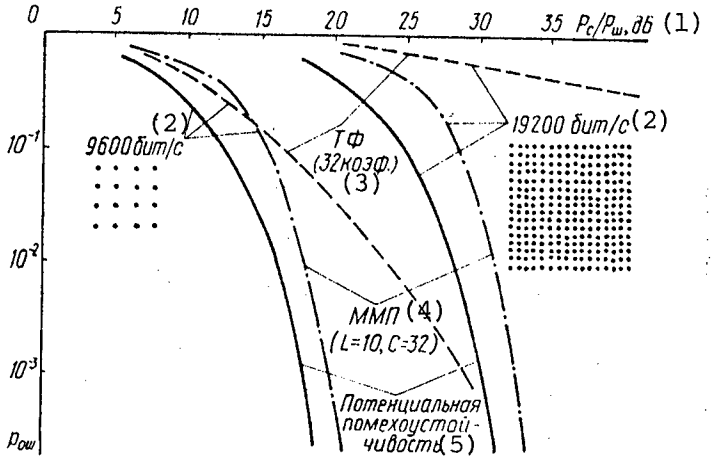


Figure 2. Dependence of the Probability of Wrong Reception of an Information Symbol on the Signal/Noise Ratio in the Communication Channel

Key:

1. db
2. bits per second
3. TF (32 coefficients)
4. MMP
5. Potential interference resistance

The interference resistance and labor-intensiveness of the algorithm were estimated using a computer and a statistical model of a discrete communication channel.

Figure 2 shows the results of modeling the work of a transversal filter with a resolver and the minimax search algorithm in a pulse tone communication channel with known pulse characteristic h_k [2]: -0.273, 0.059, 1, 0.059, -0.355, -0.077. KAM-16 and KAM-256 units characterized by a speed of 2,400 symbols per second were used for data transmission.

Analysis shows that even without a preliminary phase corrector, the minimax search algorithm insures high interference resistance of the signal conversion device. Use of a harmonic corrector under the same conditions does not permit attainment of a transmission speed of 19,200 bits per second in a tone frequency channel.

Thus use of microprocessors affords the possibility for increasing the algorithmic complexity of the discrete signal reception algorithm, and consequently for raising the effectiveness of a signal conversion device's receiver. The algorithm was modeled with a YeS-1033 computer. Next it is to be used to develop a high-speed signal conversion device for unswitchable tone frequency channels.

BIBLIOGRAPHY

1. Bel'fiore, K. A. and Park, Dzh. Kh. Jr., "Compensation by Means of Resolving Feedback," TIIER, translated from English, Vol 67, No 8, 1979, pp 67-83.

2. Klovskiy, D. D. and Nikolayev, B. I., "Inzhenernaya realizatsiya radiotekhnicheskikh skhem (v sistemakh peredachi diskretnykh soobshcheniy v usloviyakh mezhsimvol'noy interferentsii)" [Engineering of Radiotechnical Circuits (in Discrete Message Transmission Systems in the Presence of Intersymbol Interference)], Moscow, Svyaz', 1975, 200 pp.

3. Vasilenko, G. I., "Teoriya vosstanovleniya signalov (O reduktsii k ideal'nomu priboru v fizike i tekhnike)" [Theory of Signal Reconstruction (Reduction to an Ideal Instrument in Physics and Technology)], Moscow, Sov. radio, 1979, 271 pp.

11004
CSO: 1863/259

UDC 519.767

PROBLEMS OF DEVELOPING ALGORITHMIC DESIGN SYSTEM

Tashkent IZVESTIYA AKADEMII NAUK UZBEKSKOY SSR: SERIYA TEKHNICHESKIKH NAUK in Russian No 6, Nov-Dec 84 (manuscript received 20 Mar 84) pp 10-13

[Article by O. M. Nabiyev and K. K. Zagidullin, UzSSR Academy of Sciences, Uzbek Cybernetics Association of Science and Industry]

[Text] Algorithmization, understood as the "... problem of automating mental work, cannot be solved by using local algorithms; ... an entire algorithmic system ... is needed." [1]

An algorithmic design system (SAlP) is intended for building systems which control design and support design execution. Research on possible ways of building such systems has shown the need of refining the concept of a complex system. The concept of a complex system [2-4] is supplemented by the following principles:

1) A complex system (object in the real world) has properties of nonadditivity of characteristics. Nonadditivity of characteristics is understood in the sense of a system discontinuity of the second kind [5]. For example, an optimal sector program is not the sum of optimal enterprise programs;

2) A complex system (object in the real world) has the property of substantial limitedness of resources. Substantial limitedness of resources is the prerequisite of a system discontinuity of the first kind [5] consisting in a contradiction between the capabilities of various levels of control to store and process current information, on the one hand, and the requirements of optimality and efficiency, on the other.

It follows from these principles that trying to build an algorithmic design system from the "bottom up," i.e. first the individual banks and then the system as a whole, entails considerable difficulties in coordinating the interaction of the individual banks with each other.

It appears possible to build an algorithmic design system from the "top down," which assumes, in the first place, the existence of a general program to build such a system which prescribes in sufficient detail the following schemes:
--system functioning;

--system interaction with the user; and
--system development.

In connection with the high relevancy of the problem of building an algorithmic design system, research on similar directions is widespread. The large number of efforts prevents a complete listing of them; let us indicate only the basic directions:
--programming, modeling and data description languages;
--composition systems of support of programs;
--programming technology;
--instrumental design systems;
--simulation modeling;
--algorithmic design;
--theory of complex system;
--theory of algorithms; and
--pattern recognition.

In the program to build an algorithmic design system, undoubtedly, all achievements in this field are taken into account and used. In the ideal, efforts of researchers of various directions must be united, and failing this, coordinated.

Needed as the first step in building the program is a representation of an algorithmic design system, allowing presentation of the program without details, as a whole (zero level of consideration). Since such a system is an aggregate of processes on information, the interrelation of processes and information must be reflected at the zero level. Then this system at the zero level can be represented as a two-dimensional diagram (figure).

Key:
1. problem
2. formal
3. intra-system
4. model
5. processes
6. formalization
7. adaptation
8. modeling
9. computation
10. remodeling
11. readaptation
12. reformalization

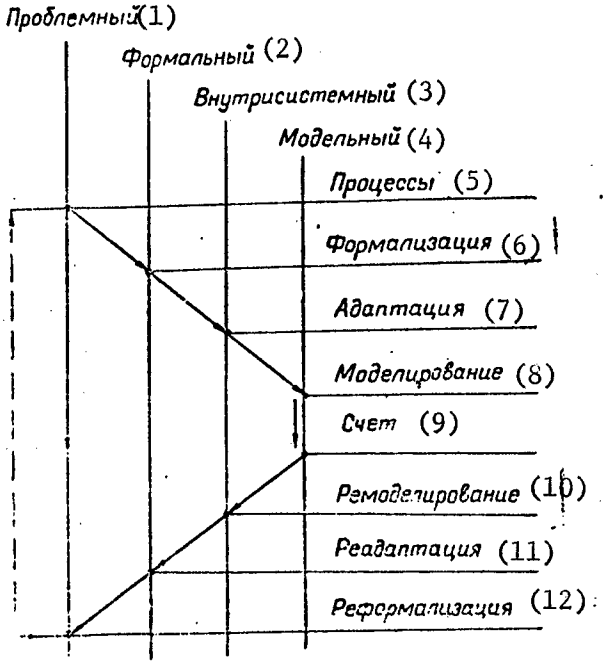


Fig. Algorithmic design system at the "zero" level

Prob'em Level of Information. In the general case, it defines the composition, type, interrelation and relation of information describing an object of design and the process of its functioning or design. For example, some

subsystem in an ASU [automated control system] is an aggregate of documents and normative, directive and current data of subject area.

Formal Level of Information. Defines the composition, type and interrelation of information on the problem level which has undergone special types of conversions aimed at achieving completeness of description, uniqueness, and consistency within the bounds of some specific goal. For example, for a local subsystem of a SAPR [computer-aided design system], output of design documentation (formal level) is represented by a description of an object being designed in some input language of description meeting the three requirements indicated above: completeness, uniqueness, and consistency.

Intrasystem Level. Defines the composition, type, interrelation and location of information on the formal level which has undergone special types of conversions aimed at supporting storage, access, change and mapping in specified spaces and time and functional bounds. This can be, for example, normative-reference data bases, current files of corrections, etc., on external storage for the algorithmic design system.

Model Level of Information. Defines the composition, type, interrelation and location of information on the intrasystem level which has undergone special types of conversions aimed at supporting the completeness, uniqueness, and consistency of a given model (informational, informational-logic, mathematical, etc.) of a specific copy of an object being designed in the algorithmic design system. This can be, for example, files of structures with interreferences at the level of a file element which comprise an informational model of a specific copy of the object being designed.

Having defined this way the levels of information and having accepted as the main assumption the principle that the basic content of work in an algorithmic design system consists in successive raising of the level of information, performing meaningful processing at the highest level, and then lowering the level of information to that specified, let us derive the natural division of the processes into groups:

1) processes raising the level of information;
2) processes performing meaningful processing on one level of information;
3) processes lowering the level of information (see figure).

Listed in the diagram are the conventional titles of processes which can be divided the following way:

a) formalization, adaptation, modeling;
b) computation; and
c) remodeling, readaptation, reformalization.

Let us examine the functioning of an algorithmic design system from another viewpoint. Let us represent an object of design in the following form:

1. Object of design:
1.1. Informational model;
1.2. Processes and their interrelation;
1.3. Components.

It is easy to see that in the formalization stage, a user of an algorithmic design system defines 1.1 and 1.2 in the most detailed way. Then the work of such a system can be represented as:

2. System functioning:
2.1. Supplementing definition of 1.1 and 1.2 with regard to their close connection;
2.2. Generation and selection of components from 1.3 based on results of 2.1.

There is now a large family of formal languages to describe information models (1.1) and processes of their interrelation (1.2). All the languages have a common property: orientation to a specific, rigidly specified, intellectually primitive processor (translator, generator, compiler) executing processing of the description in the language.

It is obvious that execution of 2.1 and 2.2 based on existing languages does not seem possible. The reason for this situation is doubtless: There are no facilities for describing semantics. Semantics cannot be formalized; this type of conviction is at the root of the negative attitude to the idea of algorithmic design in general.

In solving the problems in 2.1, naturally, questions arise on the existence of a solution and completeness in an algorithmic sense for a given algorithmic design system, which in turn have been inadequately researched.

In solving the problems in 2.2, questions of structural conversion of modules of programs are raised to a qualitatively new level.

Thus, one can identify the directions in research and development which are primary in the general program to build an algorithmic design system:

3. Primary directions of research:
3.1. Formalization of description of semantics;
3.2. Existence of solution and completeness of algorithmic basis;
3.3. Structural conversions of modules.

BIBLIOGRAPHY

1. Nabiyev, O. M.; Zagidullin, K. K.; and Kadyrov, A. V., "Structural Model of Algorithmic Design System," in the collection "Voprosy RASU" [Problems of Republic Automated Control Systems], Tashkent, No 17, 1979, pp 10-19.

2. Buslenko, N. P., "Modelirovaniye slozhnykh sistem" [Modeling of Complex Systems], Moscow, Nauka, 1968.

3. Buslenko, N. P., "Ob odnom klasse slozhnykh sistem. Problemy prikladnoy matematiki i mekhaniki" [On One Class of Complex Systems. Problems of Applied Mathematics and Mechanics], Moscow, Nauka, 1971.

4. Buslenko, N. P.; Kalashnikov, V. V.; and Kovalenko, I. A., "Lektsii po teorii slozhnykh sistem" [Lectures on Theory of Complex Systems], Moscow, Sov. radio, 1973.

5. Danil'chenko, I. A., "Two-Dimensional Integration of Automated Subsystem of Optimal Planning of Production of Machine Building Sector," KIBERNETIKA, Kiev, 1973.

6. Kabulov, V. K., "Current Problems in Developing Automated Control Systems," in the collection "Voprosy kibernetiki" [Problems of Cybernetics], Tashkent, IK s VTs AN UzSSR [Institute of Cybernetics and Computer Center, UzSSR Academy of Sciences], No 74, 1974, pp 5-17.

7. Kabulov, V. K., "Algoritmizatsiya v mekhanike sploshnykh sred" [Algorithmization in Mechanics of Continuous Media], Tashkent, Fan UzSSR, 1979.

8. Kabulov, V. K., "Algorithmic Direction in Cybernetics," in the collection "Budushcheye nauki" [Future Science], Moscow, No 10, 1977.

8545
CSO: 1863/261

COMPUTER SUBPROGRAM SPEEDS DEVELOPMENT OF CONTROL PROGRAMS

Moscow LENINSKOYE ZNAMAYA in Russian 13 Mar 85 p 3

[Text] A subprogram which has been developed in a department of the All-Union Electrical Engineering Institute will find employment in programming and computer technology for the purpose of accelerating the development and debugging of control programs in BASIC language. This subprogram is intended for multichannel parallel input-output of information and control of relays in systems for the automation of experiments and control of production processes, using M6000, M7000, SM1 and SM2 minicomputers. The subprogram takes up 500 bits of storage locations and is loaded by an absolute loader with the start address 132000 bits. The introduction of this subprogram in combination with a BASIC interpreting system shortens, by dozens of times, time for writing programs intended for input-output of analog information and the control of relays.

The use of a movable subprogram module makes it possible also to employ the subprogram in the development of control programs in FORTRAN language.

FTD/SNAP
CSO: 1863/218

UDC 621.391

CLASSIFICATION OF VITALITY SUPPORT SYSTEMS WITH STRUCTURAL REDUNDANCY

Baku DOKLADY AKADEMII NAUK AZERBAYDZHANSKOY SSR in Russian Vol 40, No 12, Dec 84 (manuscript received 20 Nov 83) pp 27-29

[Article by M.V. Kadzharov]

[Text] As controlled and monitored objects become more complex and their functional characteristics are expanded, the complexity of the controlling and monitoring complexes (UKK) also increases. Under these conditions the most important task when creating different UKK's is to support their vital activity, which is vitality. Therefore, research and development work on methods and facilities to support the vitality of complex systems is an extremely urgent scientific problem.

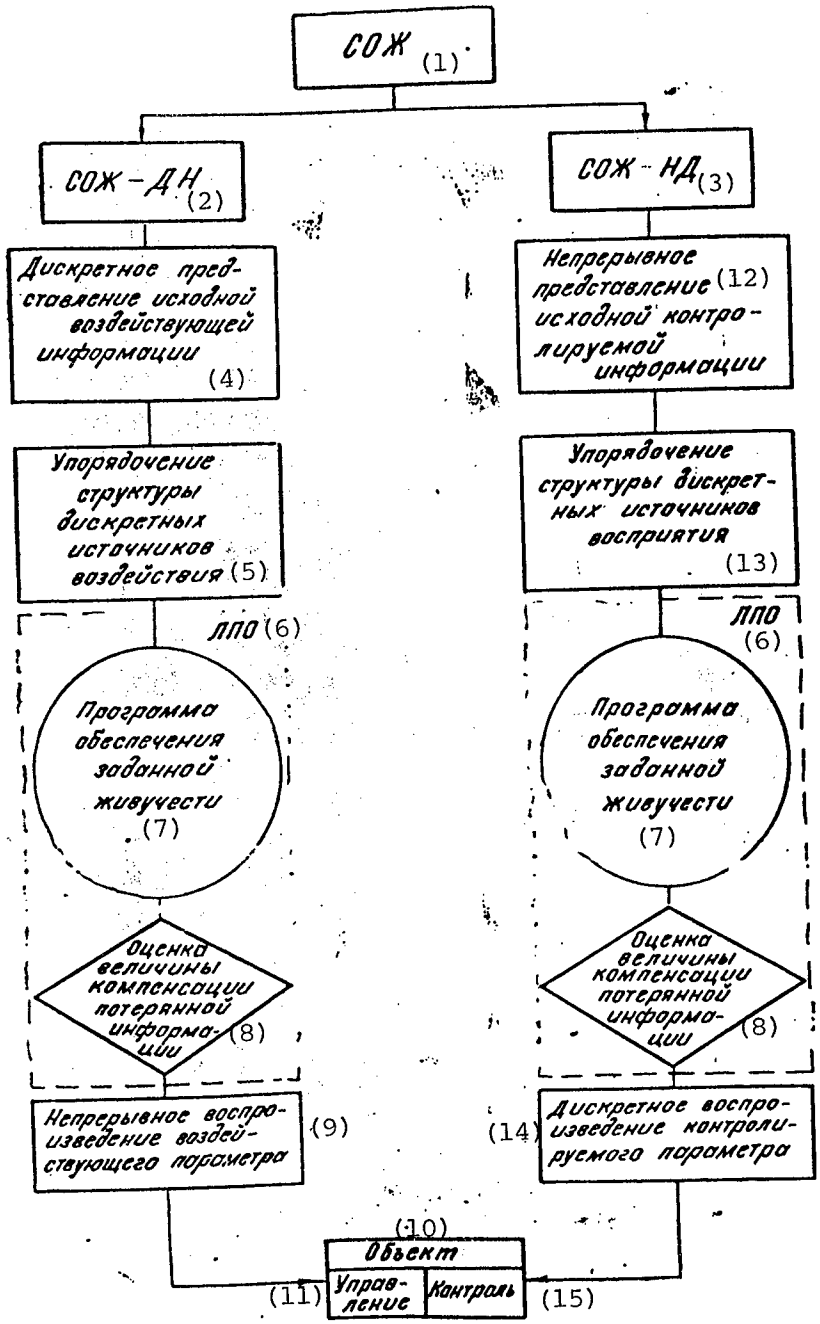
Ways have now been outlined for increasing vitality by means of redundancy, diagnosing, priority feature and so forth.

In this article we discuss the structure of the classification of vitality support systems (SOZh) with structural redundancy. Questions on the organization of SOZh's in UKK's are discussed in [1].

According to the thoughts developed in [1] on supporting the vitality of systems, and with due consideration for their main functional features, SOZh's are subdivided into subsystems of discrete-continuous SOZh's and continuous-discrete SOZh's. Each of the subsystems is characterized by a number of features with respect to representation and display of the initial information, evaluation of the amount of compensation for lost information and so on.

A block diagram of the functional algorithm of an SOZh, with subdivision of it into discrete-continuous and continuous-discrete SOZh's, is presented in the figure on the next page.

A discrete-continuous SOZh consists of acting and reproducing parts and a unit for logical transformations and evaluations (LPO). The action parameter is represented in discrete for, as a set of point information sources (TII) that are grouped (ordered) according to a certain feature. With possible limitations in such a system because of its proposed structural organization, the loss of the file of a parameter that is being reproduced is prevented. The total loss is distributed within the limits of the entire reproduction system.



Key:

1. SOZh
2. Discrete-continuous SOZh
3. Continuous-discrete SOZh
4. Discrete representation of initial acting information
5. Ordering of structure of discrete action sources
6. LPO
7. Program for support of given vitality
8. Evaluation of amount of compensation for lost information
9. Continuous reproduction of acting parameters
10. Object
11. Control
12. Continuous representation of initial monitored information
13. Ordering of structure of discrete sensing sources
14. Discrete reproduction of monitored parameters
15. Monitoring

In the acting system, the TII's that have lost servicing are in the vicinity of the elements that have acted, as a result of which the distribution of the parameter for action on the object is of an almost continuous nature and there is a certain amount of compensation for the action of the unserviced TII's.

Thus, as a result of the proposed SOZh organization method, there appears the possibility of supporting the given activity mode (given vitality) of a system as a whole without any additional expenditures.

A discrete-continuous SOZh has a structural organization similar to that of a continuous-discrete SOZh. And here, with possible limitations in the system, the output of the information file from under monitoring is prevented; the total loss of the information being reproduced is distributed within the limits of the system's entire space of states. TII's that have lost servicing are located in the vicinity of serviced elements. On the basis of an integral evaluation of the indicators of the acting TII's, it is possible--to some degree--to make a decision about the parameter in the unmonitored part.

Thus, thanks to the structural organization of a continuous-discrete SOZh there appears the possibility of fulfilling the conditions for maintaining the given viability of the entire system as a whole.

The logical transformation unit includes the operations for ordering the SOZh's internal structure for the purpose of supporting the given viability program, as well as the operation for evaluating the amount of compensation for the effect of the unserviced TII's.

The SOZh classification that has been made determines the category of the class of UKK problems, for the solution of which it is possible to use one of the subsystems that have been discussed.

BIBLIOGRAPHY

1. Kadzharov, M.V., DOKLADY AN AzSSR, Vol 38, No 3, 1982, pp 12-16.

11746
CSO: 1863/268

YeS-1045 COMPUTER USED IN GEOPHYSICS

Moscow PRAVDA 9 Apr 85 p 1

[Article by independent correspondent for PRAVDA, A. Kurkov, "The Keen Eye of the Geologists"]

[Text] Ukhta (Komi ASSR), 8 April. The computer center of the Pechorageofizika [Pechora Geophysics] Association is making final adjustments to a high-performance computer of a new generation YeS-1045. Four million operations per second: that today is the overall speed of the domestic computers installed here. Their application has allowed speeding up by two whole years the detailed seismic survey of the Khar'yaginskoye oil field, 100 kilometers from the Usa-Ukhta-Yaroslavl' pipeline and thus has permitted savings of tens of millions of rubles.

During the 20 years of existence of the computer center--the first in this branch of industry and the first in the autonomous republic--its possibilities have grown immeasurably. Figuratively speaking, if earlier geophysicists, trying to find out about the structure of a deposit, read with computer aid just the large newspaper headlines, today they can easily make out the very finest print.

9645
CSO: 1863/274

WORK ON NEW-GENERATION ROBOTS AT POLYTECHNICAL INSTITUTE

Leningrad LENINGRADSKAYA PRAVDA in Russian 5 Apr 85 p 2

[Text] More than 6,000 robots will be operating at Leningrad enterprises by the end of this year. The range of uses of these robots is constantly expanding, and robot-equipped sections have now been organized at the Leningrad Optical-Mechanical Association and the Leningrad Electromechanical Plant, for example. Manipulators have taken their place also in the first flexible automated production systems. The program "Intensifikatsiya-90" calls for further increasing the number of such mechanisms and improving their capabilities.

Personnel of the Central Scientific Research and Design-and-Experimental Institute of Robotics and Technical Cybernetics of the Polytechnical Institute imeni Kalinin are working on these problems. A number of robots of different types have been developed here in recent years. Some of them are in series production, and new models are now being developed which will be capable of moving parts of various weights and of replacing people where working conditions are difficult.

Work on development of robots of the next generation is simultaneously in progress at the institute. These robots will have artificial vision, be controlled remotely by voice, and be able to adapt to production conditions. Such devices are already being tested in laboratories.

(A photograph is given showing senior project engineer Ye. Ya. Belyayev and senior engineer S. V. Lobanov in the institute's laboratory of pneumatic robots.)

FTD/SNAP
CSO: 1863/257

UDC 658.52.011.56

FUNCTIONAL-LOGICAL SYSTEM FOR AUTOMATED PLANNING OF ROBOT PRODUCTION LINE CONTROL SYSTEMS

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (manuscript received 4 Apr 84) pp 28-30

[Article by engineer E. M. Pospelov and Candidate of Technical Sciences A. I. Savitskiy]

[Text] Combination of the merits of automated design with the advantages of the methods for preparing and operating a flexible automatic production operation [1,2] promotes a sharp rise in the effectiveness of design and production processes.

Let us examine, as the object of design, a control system for a robot production line (SU RTL) assembling printed boards. This system provides for:

assembly of printed boards with prescribed productivity and quality in accordance with the existing technology;

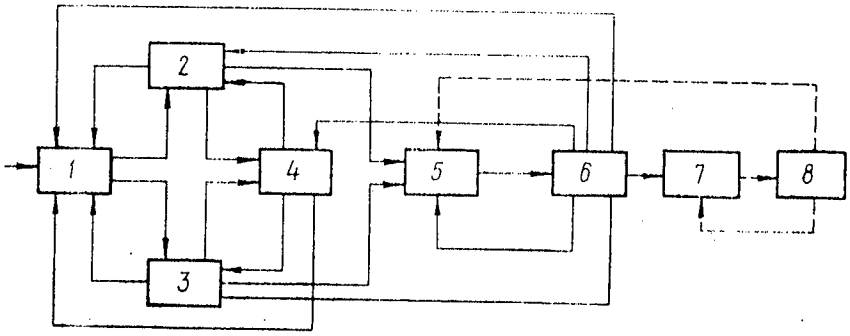
automatic adjustment of the robot production line to assemble one out of a prescribed set of types of printed boards while preserving the structure of the control system and the layout of the hardware complex of the robot production line.

Use of the modular-hierarchical approach makes it possible to distinguish design of an SU RTL, interpreted as a complex system, as a macrolevel consisting of a system of multilevel calculation macroprocedures. The flow chart for control system design is synthesized in the presence of uncertainties stemming from the modular-hierarchical approach itself [3].

When we design an SU RTL, we can distinguish design of control systems for a robot production unit, for the robot hardware complex and for the robot production line.

Design is carried out in both batch and multiprogram interactive mode. The engineer-user is viewed as a part of the software. Iteration methods are used in the design. The mutual relationships among the control system design stages are shown in the figure below. The design procedures are performed within the limits of each stage and in accordance with the flow chart using the support resources of the appropriate subsystem of the GAP [flexible automated

production] control system automated design system. The composition of the data files used and obtained in the course of design in each stage is reflected in the table.



Mutual relationships among stages of automated control system design: *1,2,3*--design of the structure, the control algorithms and the monitoring algorithms respectively; *4*--selection of the hardware complex; *5*--design the programs; *6*--modeling; *7*--preparation of the documents; *8*--manufacture and adjustment of the control system

In the first stage we design the structure of the control systems for the robot production units, the robot hardware complex and the robot production line. The main thing here is to determine the components of the system and the methods of their interaction. Moreover the following tasks are completed:

determination of the quantity of control modules in the control system, their mutual relationships and the requirements on them;

refinement of the structure of the control system and the information model on the basis of the results of stages 2, 3, 4, 6.

In the second stage we design the algorithms for controlling the instrument modules. The following are carried out in this case:

analysis of the cyclograms and algorithms of equipment operation with the purpose of determining the moments at which the states of the system and the performance of instrument modules participating in the work change;

determination of the composition of module control algorithms;

determination of the extent to which standard algorithms fit into the formed list of control algorithms;

development of special and general algorithms for module control;

parametrization of control algorithms and their change on the basis of the results of stages 4, 6.

Composition of Data Files of the Integrated Data Base

Stage	Content of Data Files Used in Design	Content of Output Data Files
1	Layouts of robot production unit, the robot hardware complex and the robot production line. Data exchange models	Composition and structure of control system. General requirements on control system modules
2	Cyclograms and algorithms of equipment operation. Standard control algorithms. Graphic models of elements in algorithm system	Instrument module control algorithms. General algorithms of three control levels. Technical requirements on the control system hardware complex. Technical assignment to develop special control algorithms
3	Equipment operating algorithms. Standard monitoring algorithms. Graphical models of elements of algorithm systems	Instrument module monitoring algorithms. Technical requirements on control system hardware complex. General algorithms for monitoring three-level control systems. Technical assignment to develop special monitoring algorithms
4	Structure of robot production cell, robot hardware complex, robot production line. General requirements on control system modules. Technical requirements on control system hardware complex. Characteristics of transducers. Characteristics of the computer, USO [entity connection device], interfaces, controllers	Composition and technical characteristics of devices in the hardware complex. Technical assignment to develop special monitoring and control devices
5	Control and monitoring algorithms. Standard control and monitoring program modules	Control and monitoring programs. Technical assignment to develop supplementary program modules
6	Structure of control system. Cyclograms and algorithms of equipment operation. Control and monitoring programs. Graphical models of the work zones of movable modules	Requirements to alter structure of control system, control and monitoring algorithms and program modules, and to replace hardware complex modules. Technical characteristics of control system corresponding to prescribed operating modes
7	Requirements of standards, GOSTs [all-union state standards]. Graphical models of system elements	Technical documents as required by GOSTs on technical and work planning

[Table continued on following page]

8	Technical documents to manufacture special devices. Technical documents as required by GOSTs on technical and work design	Amendments to technical documents

In the third stage we plan the algorithms for monitoring the function of instrument modules. The following is carried out with this purpose:

analysis of the cyclograms and algorithms of equipment operation with the purpose of determining the list of monitoring algorithms (tasks);

determination of the extent to which standard algorithms fit into the formed list of monitoring algorithms;

development of special monitoring algorithms;

development of general algorithms for monitoring the function of instrument modules;

parametrization of monitoring algorithms and their change on the basis of results in stages 4, 6.

The method of functional-cost analysis is used to select the hardware complex for the control system (the fourth stage). The control system is synthesized using maximally unified instrument- and program-compatible devices.

In the fifth stage the results of analyzing the control and monitoring algorithms are used to determine the possibility of employing standard program modules to act as such algorithms, technical assignments are drawn up for supplementary program modules, and the latter are developed. The control and monitoring program for all of the required types of printed boards is written up, and changes are made in it on the basis of the results of stage 6.

In the sixth stage, during simulation modeling of the control system, we refine the structure, work the control and monitoring algorithms into their final form, calculate the parameters of the processes in relation to particular operating conditions using known mathematical models, search for bottlenecks, determine the demands to be imposed on any changes to be made, distribute the program modules among different hierarchical levels and establish the frequency of their use.

In the seventh stage we write up the technical documents in accordance with GOST requirements. The computer graphics subsystem, which is used to carry out the tasks of this stage, is also employed in all stages to document the intermediate planning and modeling results. The program storage devices for the computers contained within the hardware complex of the control system are prepared in this stage as well.

The eighth stage involves manufacture of nonstandard instrument modules, creation of mock-ups of the control system, and its adjustment. In order for the design system to be complete, information must be made available on the object of the design throughout the entire time of its life cycle. This promotes improvement of the quality of not only the object being designed but also the design process itself. The appropriate algorithms are foreseen for processing statistical data in the robot production line control system.

The functional-logical design system examined here is being applied at a certain enterprise as part of an automated system for designing flexible automated production of printed boards, and it may be recommended for broad use.

BIBLIOGRAPHY

1. Zhuk, K. D., "Avtomatizirovannoye proyektirovaniye logiko-dinamicheskikh sistem" [Automated Design of Logical-Dynamic Systems], Kiev, Naukova dumka, 1981, 205 pp.

2. Mayorov, S. A. and Orlovskiy, G. V., (editors), "Gibkoye avtomaticheskoye proizvodstvo" [Flexible Automatic Production], Leningrad, Mashinostroyeniye, 1983, 376 pp.

3. Norenkov, I. P., "Vvedeniye v avtomatizirovannoye proyektirovaniye tekhnicheskikh ustroystv i sistem" [Introduction to Automated Design of Technical Devices and Systems], Moscow, Vysshaya shkola, 1980, 368 pp.

11004
CSO: 1863/259

UDC 620.179

AUTOMATED FLAW DETECTION INFORMATION-MEASURING SYSTEM

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (revised manuscript received 18 Jan 84) pp 48-50

[Article by engineer B. M. Berezyuk and Candidate of Technical Sciences Yu. M. Shumkov]

[Text] Now that the metrological characteristics of nondestructive control have improved, the following problems have become important: insuring high efficiency and reliability of such control, and providing for the processing of multiparametric flaw detection information in such a way as to permit separation of the parameters being checked; building a mathematical model of flaws in order to associate any flaw detection information received with the parameters of different types of flaws; providing for current display of flaw detection information in a form convenient to the operator; forming and storing a flaw detection log on checked articles and so on.

One of the most promising ways of solving these problems integrally is to develop and create general-purpose flaw detection information-measuring systems based on highly productive computers making it possible to completely automate the acquisition, processing, display and storage of flaw detection information [1-3]. Moreover flaw detection information-measuring systems should be designed on the basis of the modular principle, making sure that the hardware and software of individual modules are compatible, which will make it possible to efficiently change the system's structure depending on the particular requirements. Development of applied programs for a flaw detection information-measuring system (development of the algorithms for flaw inspection in a real measuring process) requires a large volume of statistical flaw detection information, obtained using samples with standard flaws. Self-contained measuring devices permitting the recording of flaw-detection information on storage devices (punched tape, punchcards and so on) can be used for this purpose. This information can then be processed by general-purpose computers. But when series-produced multichannel K-489 digital voltmeters are used as the output blocks of such self-contained measuring devices, the flaw detection information is recorded on punched tape using codes designed for the Mir computer. It is impossible to process flaw detection information obtained in this fashion and to display it in a necessary and convenient form with a Mir computer. To solve this problem we developed a complex of applied programs, the structure of which is shown in Figure 1. The programs were run experimentally with a YeS-1022 computer. The complex of programs is being used in

experimental research being carried out in order to accumulate flaw detection information for testing purposes; algorithms for inspecting flaws in the course of the work of a flaw detection information-measuring system are being developed on the basis of such information.

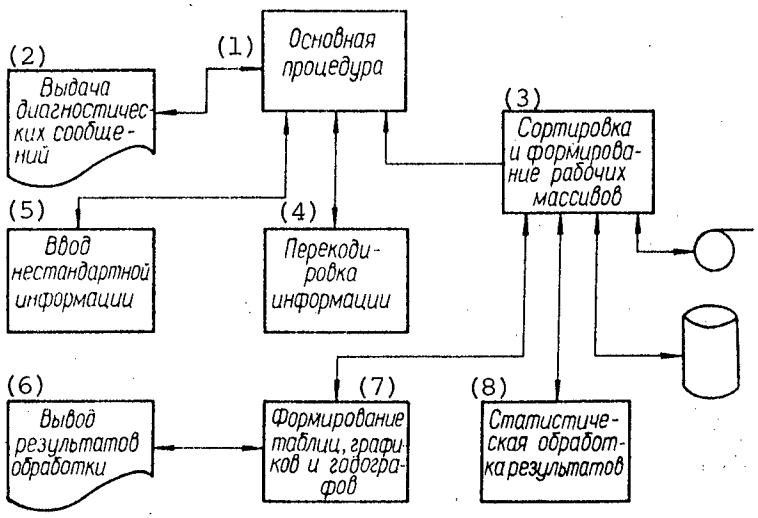


Figure 1. Structure of a Complex of Programs for Processing Flaw Detection Information with a YeS Computer

Key:

1. Basic procedure
2. Output of diagnostic messages
3. Sorting and formation of working files
4. Information recoding
5. Input of nonstandard information
6. Output of processing results
7. Formation of tables, graphs and hodographs
8. Statistical processing of results

An experimental model of a flaw detection information-measuring system intended for article quality control in production was created. The SM-4 process control computer complex, which is characterized by high speed, a wide assortment of peripheral devices, a "Common Bus" interface making it possible to compatibly connect nonstandard external devices, and well-developed systemic and applied programs, was selected as the basic computer for the flaw detection information-measuring system. The structural diagram of a flaw detection information-measuring system used for automatic control of long welded joints by the eddy current method is shown in Figure 2.

Besides the standard equipment contained in an SM-4 process control computer complex (DP--memory dispatcher, RA--arithmetic expander, BPZ--floating decimal block, OZU--main memory, UVPMD--magnetic disk external memory, KUVPMD--UVPMD controller, UVPML--magnetic tape external memory, KUVPML--UVPML controller,

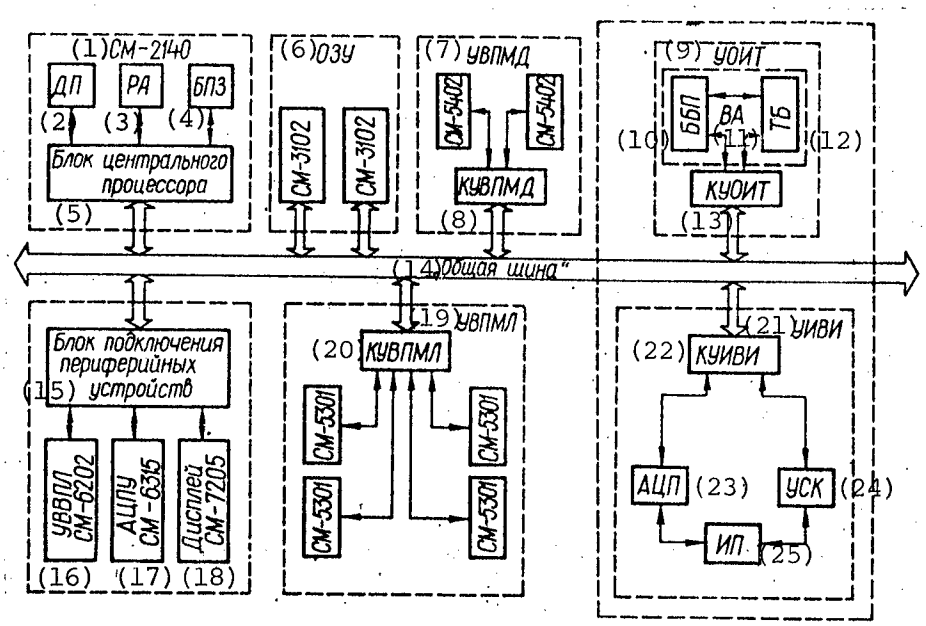


Figure 2. Structure of a Flaw Detection Information-Measuring System Based on an SM-4 Process Control Computer Complex

Key:

1. SM-2140
2. DP
3. RA
4. BPZ
5. Central processor block
6. OZU
7. UVPMD
8. KUVPMD
9. UOIT
10. BBP
11. VA
12. TB
13. KUOIT
14. "Common Bus"
15. Peripheral device connecting block
16. UVVPL
17. ATsPU
18. Display
19. UVPML
20. KUVPML
21. UIVI
22. KUIVI
23. ATsP
24. USK
25. IP

UVVPL--punched tape input-output device, ATsPU--alphanumeric printer), the flaw detection information-measuring system now contains nonstandard devices intended for information measurement and input--the UIVI, and for information display on the screen of a television receiver--the UOIT.

The UIVI consists of:

a scanner (USK) including a scanning unit based on an RDD-1 discrete plotter, an eddy current converter, the flaw detection measuring instrument (IP) and an analogue-to-digital converter (ATsP) (the hardware was developed in the Physical Mechanical Institute of the Ukrainian SSR Academy of Sciences);

a KUIVI--the UIVI controller, by means of which the UIVI communicates by way of the "Common Bus" interface with the SM-4 process control computer complex.

The KUIVI includes a USK control device (UUSK) (the hardware was developed by the Lvov Polytechnical Institute).

The UOIT is composed of:

a visual analyzer (VA), the principal units of which are a memory buffer block (BBP), a television block (TB) and a block providing communication with the controller (VA) (the hardware was developed by the Physical Mechanical Institute of the Ukrainian SSR Academy of Sciences);

the KUOIT--the VA controller intended for providing communication to the UOIT with the SM-4 process control computer complex by way of the "Common Bus" interface (the hardware was developed in the Lvov Polytechnical Institute).

Specific physical addresses are assigned to the registers of the UIVI and UOIT controllers, and the procedures of working with them are the same as with the registers of standard devices of the SM-4 process control computer complex [3]. Owing to presence of the KUIVI and its UUSK in the UIVI, the processor is free to process the data files in intervals between collection of information on the state of the article being tested--that is, data processing can be carried out in real time or close to it.

In order to permit the flaw detection information-measuring system to operate in real time, a real time operating system designed specially for a flaw detection information-measuring system with regard for connection of non-standard UIVI and UOIT external devices was made the basis of the programs developed by the Lvov Polytechnical Institute. The programs consist of:

a UIVI driver (systemic) program; a UOIT driver (systemic) program; a TEST (applied) program.

The driver programs can be loaded into the real time operating system, and they permit access to the UIVI and UOIT controllers through standard resources of the real time operating system. The TEST program is intended for organization of the work of the hardware and software on the basis of the adopted algorithm for scanning a long welded joint.

During scanning, a long welded seam is conditionally divided into L frames consisting of M lines with N points in each line. The eddy current converter, which is program-controlled by the UUSK, makes measurements at each point on its trajectory as it moves successively from one line to the next in the frame. After the measurement result is converted into binary code and a complete word of the established format is formed, this result is transferred by control bits to the memory of the SM-4 process control computer complex in response to an interrupt request. The process measurement results are transferred by frame by frame to the TB screen for display, and they may be printed by the ATsPU or recorded in the magnetic disk file.

The system for checking flaws in long welded joints described here was designed for use with the electromagnetic (eddy current) nondestructive control method, but it can be used as the basis for automating flaw detection in articles

employing other nondestructive control methods (for example ultrasonic). The anticipated annual economic impact from introducing the flaw detection information-measuring system into production is about 450,000 rubles.

BIBLIOGRAPHY

1. Klyuyev, V. V., Kurozayev, V. P. and Khvatov, L. A., "A Complex of Resources for Automatic Nondestructive Quality Control of Tubing in a Working 130 Rolling Mill," DEFEKTOSKOPIYA, No 2, 1976, pp 77-84.

2. Rodin, A. A. and Sukhorukov, V. V., "Automation of Eddy Current Apparatus for Nondestructive Control Based on Microprocessors," in "Metody i pribory avtomaticheskogo i nerazrushayushchego kontrolya. Elektromagnitnyye metody" [Methods and Instruments for Automatic and Nondestructive Control. Electromagnetic Methods], Riga, 1981, pp 16-22.

3. Berezyuk, B. M., Kudryavtsev, A. T., Tsukornik, G. V. and Shumkov, Yu. M., "An Automated Measuring System for Checking Welded Joints by the Electromagnetic Method," in "Tezisy dokladov vsesoyuznoy konferentsii IIS-81. (L'vov, 27-29 sent. 1981 g.)" [Abstracts of Reports of the All-Union Conference IIS-81. (Lvov, 27-29 September 1981)], Lvov, 1981, pp 135-137.

11004
CSO: 1863/259

UDC 681.3.06

RED-GRAPHIC EDITOR IN PRINTED CIRCUIT BOARD DESIGN SYSTEM

Novosibirsk AVTOMETRIYA, in Russian No 5, Sept-Oct 1984 (manuscript received 8 May 1984) pp 42-46

[Article by E. A. Talnykin, Novosibirsk]

[Text] Introduction

[1] gives the main principles for a program complex for the design of printed circuit boards based on the "Elektronika" computer family or the SM-4 which was developed at the Academy of Sciences USSR Automation and Electrometry Institute. The present paper considers the main component of the complex which is an interactive system for graphic editing of printed circuit board topology (RED). The system works either on a computer with graphic peripherals connected by a link line or on a microcomputer in an autonomous configuration at the work site [2]. Terms and concepts introduced in [1,2] will be used below. (Since the paper does not contain complete instructions for work with the system, certain elements related to technical details are not described.)

Principles of Editor Operation

Printed board designs are stored in archives in file form. In order to work with printed boards there is a virtual field in the editor corresponding to dimensions of 1600 X 1600 mm on the circuit boards with resolution of 0.025 mm. Information can be stored on 15 layers of the virtual field which makes it possible to simultaneously edit up to 15 layers of the printed board. Immediately after the system has been turned on the virtual field is empty and can begin to create a new design. If it is necessary to scan and check a printed board design existing in the archives it can be extracted for editing ("reading" in the virtual field). The initial copy of the printed board does not take part in the editing process. At the end of the editing process a design is obtained which can be written into the file (usually in a new version of the initial file).

Since the printed circuit board design consists of printed elements and connections, editing amounts essentially to the removal of old and the formation of new elements by means of the automated features of the graphic editor.

During the editing process, part of the virtual field of the printed board (rectangular window)constitutes the representation field. A fragment of the represent-

ation field can be shown on the display screen. The representation field is discrete. It separates into two or three maps each with its own color [2]. The dimensions of the representation field are determined by map capacity. Each of the layers of the printed board may not always be represented or may be represented only on one of the maps. Several layers may be represented on one map if this does not lead to ambiguity for the user. The screen shows the layer design of the printed board which is geometrically similar to its real image (on inspection) to the extent that this is allowed by the selected resolution which is determined by the assigned mapping scale. The unit scale was arbitrarily fixed at 2.5 mm for 6 representation elements.

During the work of the editor, a certain grid is always established to whose nodes all newly introduced elements of the printed board are reduced (by means of rounded coordinates). The minimum possible step of the grid is 0.025 mm.

Representations of masks of the design slide are not incorporated into the system but are stored in the form of individual text files. The description language for the slides is presented in [1]. Sometimes, in order to obtain faster sketching during the work of the editor,a stylized variant of the design slide is used while the real variant is used for preparing the printed board.

During the design of the printed circuit board repeated fragments are encountered and the RED editor incorporates machrotechnology facilities in order to work with them. A macro is a fragment of the printed circuit board design which can be shifted several times in different positions and with various orientations.

An editing session usually consists of reading the printed board design; selection of the design slide file; assignment of the correspondance of printed board layers to the memory display maps; selection of the fragment for representation; setting of the scale, configuration of the representation field and other parameters. If the editing has not been completed and the session must be interrupted then it is possible to store the entire content of the system in the file and, if necessary, to continue the work from the previous position.

There are two modes in editing operations, i.e., command and functional (graphic) modes which will be decribed below in detail. If, in order to complete certain operations a transition to the functional mode is not necessary, it is possible to use the editor from a terminal which does not have graphic equipment.

Command Mode

In this mode, interaction of the operator with the system is carried on at the level of text commands. An indicator that the system is ready to rceive the next command is the apperance on the screen of the prompt "RED". By the input of command string it is possible to correct errors after it has been received by the operating system.

The command string consists of the name of the command and possible parameters separated by points or spaces. The name of the command may be abbreviated. Most of the most used commands can be abbreviated to a single letter.

We will describe the main directives used in this mode: [in English text - ed.]

READ file name
Read in the virtual field the design of the printed circuit board written in the indicated file. The information read from the file is mixed with previous data and cannot be separated.

WRITE file name
Write in the indicated file the current state of the printed board in the edited design.

SAVE file name
Save in the indicated file the current state of the work session. Not only is information on the printed board stored but also all additional parameters necessary for continuing the interrupted session.

OLD file name
Resume the previously interrupted work session.

RESET
Put system in the initial state and clear virtual field of printed board. The command is equivalent to restarting the system.

SUSPEND
Effect a stop in the system freeing the terminal for work with other programs. Start up of work of editor is carried out by the operating system command (RESUME).

SCALE m:n
Fix scale for representation of printed board on display (setting is relative to the arbitrary unit scale).

GRID d[xy]
Fix grid with step equal to d. The additional parameters (x,y) determine the coordinates of one of the nodes of the grid. Thus the grid can be shifted for a value which is not a multiple of the step.

FIELD n:m
Fix parameters. Here n is the number of the elements of the representation in the line, m is the number of the map on which the representation should be laid out.

MAP l/m
Fix representation of layer l of the printed board on the map with number m of the display. The layers which are not indicated in the command are not represented.

SLIDE file name
Read from the indicated file the description of the design slide making it possible to create the representation of the printed board which is similar to the real one.

WMACRO file name
Write the macros which are determined at a certain moment in the file with the

indicated name.

RMACRO file name
Read from the file the macros which were previously written by the command WMACRO.

DISPLAY [xy]
Transition to functional mode. The lower left corner of the representation is fixed at the point with the coordinates (x,y) of the field of the printed board. The command makes it possible to arbitrarily shift the representation field of the board. If the parameters are left out of the command, there is a transition to the functional mode and the representation is stored.

Functional Mode

In the functional mode, each function is operated by pressing the corresponding button on the keyboard. The implicit argument for almost all functional operations is the current position of the locator on the display screen. Shifting of the locator on the screen as well as screen scaling and screen shift with respect to the imagery are carried out by the program software base of the graphic display [2] and are not described here. Independently of whether graphic dialogue equipment is present in the work site configuration (functional keboard or means for positioning locator) the software base always makes it possible to handle the situation by means of the alphanumeric keyboard and additional equipment only makes operations more convenient. We will present our descriptions in terms of the conventional names of the functions (function keys) and will write them in quotations.

The system has a set of fixed parameters, for example, current layer and current instrument for connections, printed elements and plate holes. The values of these parameters can be changed in command as well as function mode.

Path Construction

Paths can be in the form of a broken line with possible transitions from layer to layer. In order to begin construction of paths, the locator must be brought to the starting point of the path and the "Start" button must be pressed. The locator then moves to the next point of the broken line and by pressing the key "Layer N" the number of the layer is indicated from which the next segment of the path is to be drawn. If at a certain node of the path a transition is necessary to another layer, this is carried out by pressing the button "Transition" after the path has approached the indicated point. The path is executed by the current instrument for connections while the transition from layer to layer is carried out by the instrument for printed circuit holes.

Printing of Elements

There are several methods for inserting printed elements into the designed board. One of these has already been described and consists of fixing circuit holes during the process of path design. In order to introduce the printed element at the position of the locator on the current layer it is necessary to press the key "Print" and then to introduce the number of the mask. The general element is printed by the current instrument for printed elements by pressing the key "General".

Pointer Operation

In order to realize certain functions, a mechanism is necessary for distinguishing the elements of the printed board. Pressing the button "Search" leads to illumination of the element which is in the neighborhood of the locator. The last illuminated element is considered to be indicated as long as the locator has not shifted. The following operations can be carried out on the indicated element by pressing the corresponding buttons: "Removal"- removal of indicated element;"?"- find information on the element (number of mask, layer, coordinates); "Cut": cut the indicated connection into two parts at the point where the locator is positioned; "Replace": replace the mask for the indicated element (the current mask is fixed for the connection or printed element depending upon the object discriminated); "Link": distinguish as a whole the entire link formed by the indicated connection in its layer. The result is that the link is illuminated and is then repeatedly illuminated by pressing the button "Link" even if the position of the locator is changed or the screen shifts in relation to the representation. This makes it possible to remove the link while "Replace" leads to the replacement of the masks of the masks of all connections by the current instrument for connections.

Definition of Macros

In order to determine a macro, the rectangle containing the required fragment of the board is assigned whose sides are parallel to the coordinate axes. This is realized by bringing the locator to one of the vertices of the rectangle (base point of the macro) and pressing the button "Start Fragment". The locator is then shifted to the opposite vertex of the rectange and the button "End Fragment" is pushed. Any button linked to the macro is then pushed.

Macro Activation

In order to activate macros, it is necessary to bring the locator to the assumed location of the macro base point and to press the button "Activate" and then to press the button with the name of the macro. The corresponding macro will then appear on the screen. After verification by the operator, the fragment is entered into the circuit board and if it is rejected it is deleted from the screen.

In addition to those enumerated, there is a series of functions for setting various parameters, switching on and off the representation of the storage map, determining locator position, fixing priorities for layers on the representation, drawing grids, etc. and there is also a button for returning to command mode.

Conclusion

The equipment presented above has been used over a two year period in several work units and organizations and has been shown to be a convenient and effective support system for the design of printed circuit boards. This equipment, which is not serially produced, is necessary for the organization of work sites and is small in size and easily duplicated so that it is possible to reproduce the system without special efforts.

We would like to note the contribution of P.M. Peslyak who adapted the system for work as a component of an autonomous work station as well as S.L. Ivashin who studied

the various protocols for computer links. I.A. Nikolayeva and V.V. Krepets were the first and most active users of the system. They supplied essential support in the testing and also made many constructive suggestions as to the functional composition and organization of the user interface.

Bibliography

1. Talnykin E.A. "Sistema proyektirovaniya pechatnykh plat na EVM semeystva "Elektronika"" [System for Printed Circuit Board Design on "Elektronika" Family of Computers], AVTOMETRIYA 1984, No 4.

2. Kovalev A.M. and E.A. Talnykin. "Graficheskiy displey rastrovo tipa dlya sistem dvukhkoordinatogo proyektirovaniya" [Raster-Type Graphic Display for Two-Coordinate Design Systems], AVTOMETRIYA 1984, No 4.

12497
CSO: 1863/146

COMPUTER-AIDED DESIGNING OF MICROWIRE ALLOYS, MICROCIRCUITS

Kishinev SOVETSKAYA MOLDAVIYA in Russian 15 Feb 85 p 4

[Excerpt] An SM-4 computer which is in operation at the Electrical Instrument Building Scientific Research Institute of the "Mikroprovod" (microwire) Research and Production Association cannot be classed among the computers with the largest capacities. It is not distinguished by high speed and does not possess a huge storage capacity. However, it is bright and knows how to think logically, analyze and carry on a dialog with a partner. All of this is necessary, since the computer performs the role of designer in such cases.

"What is your computer capable of?," I asked Candidate of Technical Sciences S. Zotov, head of the institute's department of materials, technology and automation of production processes.

"First of all, a few things must be explained precisely. Our computer does not work in isolation. What I mean is that it plays the chief role in a technical complex. As for your question: the computer is now programmed to design alloys with the necessary properties for microwire and microelectronics, and to issue microcircuit diagrams. Next will come the designing of circuit boards for electrical measuring instruments and household electronic devices and, in the more distant future, the solution of all design problems that are characteristic of our institute."

Sh. Landa, head of the institute's laboratory of automation and automated systems, showed me a printout from the computer. Figures were arranged in even columns on this sheet of paper. They were variants proposed by the computer.

"Imagine that you are a developer," said Sh. Landa. "You examine variants proposed by the computer and then feed more rigid parameters into it. The computer now proposes not hundreds of variants but only five, for example. You make the requirements for the alloy still more rigid. You then see only one optimal variant proposed for you."

We were then familiarized with the use of the computer for designing microcircuit diagrams. Points of light first flashed onto the dark blue screen of a

video terminal, reminding us of a starry sky. They were terminal areas--locations of components of a microcircuit. Straight lines running from point to point then appeared; routes of conductors were being marked.

"In the end, we obtain a perforated tape from which a mask is made," said Sh. Landa.

The process of introducing the computer to the designing of circuit boards has not yet been completed. The programmers say jokingly that the computer is still learning. The benefits which designers obtain from this are already obvious, however,

(A photograph is given showing S. Zotov, Sh. Landa and senior engineer T. Zuyeva analyzing computer output data for the designing of alloys.)

FTD/SNAP
CSO: 1863/218

UDC 681.324

"ELEKTRONIKA-60"-YES MULTI-COMPUTER SYSTEM

Novosibirsk AVTOMETRIYA in Russian No 5, Sept-Oct 1984
(manuscript received 12 January 1984) pp 102-104

[Article by F. A. Zhuravel', E. B. Kruglyak, V. V. Savel'yev and
A. V. Shafarenko, Novosibirsk]

[Text] For many experiments it is necessary to have a real time data collection and control system and large archives of programs for processing the aggregated data. In addition, there must also be convenient means for representing processing results, producing hard copies, etc. However in many cases it is not economically expedient or suitable to have a powerful computer system at the work site. The indicated requirements can be satisfied by microcomputer complexes located directly at th work sites and by collective use computers. One of the variants of the solution to this problem is presented in the present paper.

In the method realized by the authors, multi-computer functioning is distributed as follows:

The microcomputer carries out data collection and control of the experiment in real time, initial data processing, equipment testing and preparation of the unit for the experiment as well as in-line representation of computation results;

The multi-computer system carries out creation, storage and management of archives of data and programs, extended data processing, transmission of results to work site and generation of hard copy.

This type of distribution supposes that the microcomputer operates under the control of a real time operating system.

At the work site, the experimenter uses the "Elektronika-60" microcomputer for which a large amount of equipment has been developed which is necessary for the automatic system: controller crate, equipment for data collection and control, display equipment, communication links, etc.. The microcomputer has the RT-11 (version 4) real time operating system in which the exchange of data with the units is carried out by drivers [1]. The link driver carries out access to the multi-computer disc memory. On the YeS-5052/5061 disc a special disc structure is organized: for every work site a file is created which is divided into several non-intersecting fragments with the same dimensions which are called below, random access virtual units. There are usually 2 or 3 with dimensions of 2000 blocks

with 512 bytes. One of the random access virtual units can serve as a resident unit of the RT-11 operating system. The operating system's file structure is created on the random access virtual unit and access and protection, etc., are implemented.

Exchange with the random access virtual unit is initiated by the microcomputer and occurs by batches which can be of two types: header batch and data batch.

The header batch contains the number of the random access virtual unit, number of block in random access virtual unit with which operation begins, number of transmitted words and control word. This type of protocol makes possible the required data transmission and transmission errors are easily detected.

The figure shows the diagram for this communication protocol. It shows that the protocol is asynchronous, i.e., it is realized according to the "interrogation-response" principle. There is an internal timer in order to eliminate possible "sticking" in the multi-computer system communication interface.

In addition to the main protocol, the system uses an additional one which makes it possible to organize the work of the microcomputer and of the multi-computer system terminal. In this case, the batch consists of one byte and the receiver and transmitter operate independently. The assigned program serves for transmission of directives to the multi-computer system from the terminal and supports the microcomputer as a loader.

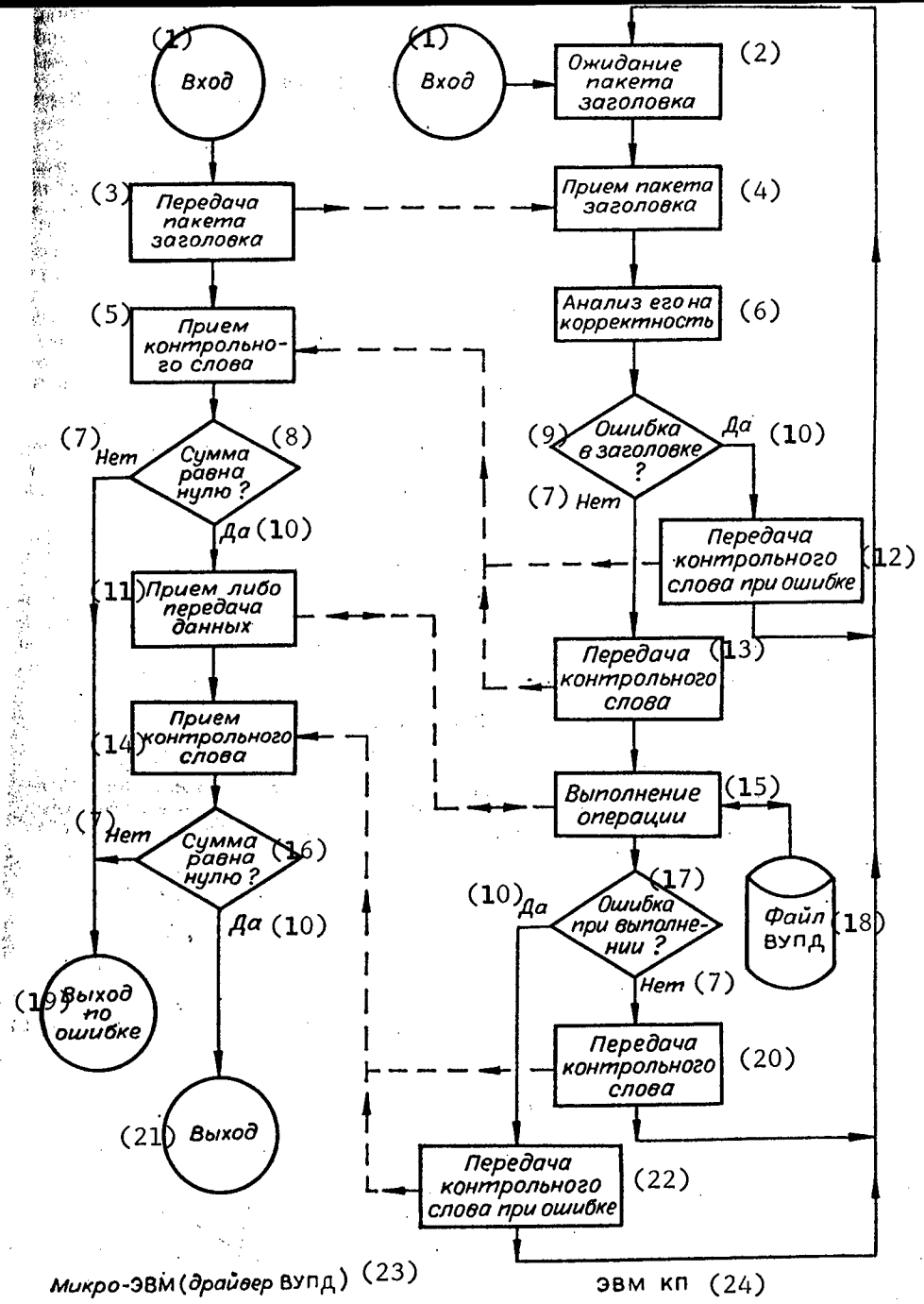
For the YeS computer, the authors developed a high-speed sequential interface and multiplex channel interface block. The exchange between the multiplex channel interface block and the interface is based on the bus principle which allows up to 16 interfaces with one multiplex channel interface block. A compatible sequential interface is used at the work site [2]. The exchange by means of one- and two-byte words occurs with a speed of up to 20 K words/s.

The interface has control and status byte registers and carries out the commands "Write", "Read", "Control" and "Read Status". The status register holds information on the functional mode of the interface and on situations which have arisen in the exchange process.

The following exchange modes are envisaged: byte or two-byte, time-out activation, generation of attention signal.

The main components of the complex software are: resident equipment driver; loading equipment and dialogue program microcomputer; program for access to peripherals from the communication link in the multicomputer system.

In order to work with different types of peripherals of the multi-computer system several drivers are necessary. However the RT-11 operating system does not allow them to be used simultaneously for one physical unit (on the communication link). Because of this function of the driver there was a constraint on the main task which is to exchange with the random access virtual unit. The other peripherals of the multi-computer system are considered supplementary and the possibilities for exchange with them reduce to directives for the writing of files. Such directives are given directly from the terminal by a supplementary protocol.



Key:

1. Input
2. Waiting for header batch
3. Header batch transmission
4. Header batch reception
5. Control word reception
6. Analysis for correctness
7. No
8. Sum = 0?
9. Error in header
10. Yes
11. Data reception or transmission
12. Transmission of control word for error
13. Transmission of control word
14. Control word reception
15. Execution of operation
16. Sum = 0?
17. Error in execution?
18. Random access virtual unit file
19. Error output
20. Control word transmission
21. Output
22. Transmission of control word for error
23. Microcomputer (random access virtual unit driver)
24. Multi-access computer

The program for access to the peripherals of the multi-computer system carries out the following functions:

1) Reception and analysis of requests for exchange with random access virtual unit, input/output of data from magnetic disc storage, reception/transmission of data on communication link (main protocol);

2) Reception, syntactic analysis and execution of directives:
PRINT- output on alphanumeric printer of RT-11 files located in random access virtual unit;
PUNCH-output on perforated tape of RT-11 files located in random access virtual unit;
TAPE- exchange between magnetic tape storage and random access virtual unit;
BOOT- loading of RT-11 operating system or file in binary tape format. This directive is used, in particular, for loading texts stored in the random access virtual unit.

After loading the system, the exchange between the "Elektronika-60" computer and the multi-computer system is carried on by means of a resident driver. In this case, the user works within the RT-11 operating system and can use all its possibilities: editors, translators, auxiliary routines, drivers, etc.. In particular, the user can set the driver for any unit (including resident random access virtual unit) and make other loadings.

Attention should be given to the fact that the "Elektronika-60" computer can operate in the system under discussion under RT-11 control for a minimum configuration (central processor, primary storage with 16 K words, terminal interface and loader).

The described complex consisting of five experimental work sites has already been in use for more than a year and was shown to be highly effective and convenient in operation.

In conclusion, the authors would like to thank their associates A.B. Bolychevskiy and K.A. Gilev at the terminal systems laboratory of Novosibirsk State University for very fruitful discussions and for making available some RT-11 programs and loading programs without which it would not have been possible to carry out the work

Bibliography

1. "RT-11. Software Support Manual", Waynard [sic], Massachusets, DEC, 1980.

2. Bazhan A.I. et al. "Mnogomashinnyy kompleks avtomatizatsii fizicheskogo eksperimenta (tekhnicheskiye i sistemnyye sredstva)" [Multi-computer Complex for Automated Physical Experiment (Hardware and System Equipment], In collection: "Tr. Vtorogo vsesoyuz. seminara po avtomatizatsii nauchnykh issledovaniy v yadernoy fizike i smezhnykh oblastyakh" [Proceeding of the Second All-Union Seminar on Automation of Scientific Research in Nuclear Physics and Related Areas], Novosibirsk, IYaF SO AN SSSR, 1982.

12497
CSO: 1863/146

UDC 681.3:656.2

RECOMMENDED HARDWARE AND FUNCTIONAL TECHNOLOGY OF DATA CONCENTRATION POINTS

Moscow AVTOMATIKA, TELEMEKHANIKA I SVYAZ' in Russian No 3, Mar 85 pp 12-13

[Article by G. P. Zheleznyakov, chief, PKTB (not further identified) Department, Rail Transport Automated Control Systems]

Data concentration points (PKI) and data points (IP) are created on the railroads as data subdivisions of the automated system for operational traffic control (ASOUP).

PKI are organized at sorting, section and large freight stations, and in the railroad's divisions and administrations. They are intended for information services and for utilization of the calculation results of points of data origin not equipped with hardware for direct work with a DVTs [railroad computer center] or SVTs [computer center network] via telegraph or telephone communication channels.

PKI workers are responsible for reception of data to be transmitted to a computer from operators of technical offices, store cashiers, station attendant operators, depot attendants and other railroad subdivision workers attached to the given concentration point.

Data are received by a computer storage device in the form of documents or by telephone. Data obtained in this fashion are checked out and prepared on the storage device for transmission to the DVTs. Prepared and received in this way, data are then transmitted from the storage devices to the DVTs. After computer check-out of these data, workers of the PKI prepare and transmit correction messages or corrected initial texts to the DVTs. If there are errors in the data that cannot be corrected by a PKI worker on his own, the content of the data is verified at its point of origin.

PKI workers prepare and transmit inquiries to the DVTs computer on the basis of a prearranged schedule or as requested by users assigned to them. The obtained results are transmitted to users via communication channels, courier or telephone. All received and transmitted messages and calculation results are recorded in special logs.

When data concentrators based on SM-1800 microcomputers are used as the technical base of the PKI, the functions of PKI workers listed above are

performed automatically, without the participation of personnel. When terminals are installed at work stations, the workers themselves prepare and transmit data to the DVTs computer. In addition they may be assigned the responsibilities of collecting data by telephone from points of data origin within the zone serviced by the PKI, and of transmitting these data to the computer.

To support the principal functions, PKI are outfitted with apparatus for data retrieval, preparation, reception, transmission and output (print-out). This includes T-63 and F1100 telegraph apparatus, DZM-180 alphanumeric printers, TAP-3A, TAP-3V, TAP-3S and TAP-34 data transmission devices (APD), YeS9002 and YeS9004 devices for data preparation on magnetic tape (UPDNML), SM-1800 microcomputers and other apparatus. The particular composition of the apparatus is determined by the volume-time characteristics of data flows, the place at which the apparatus is installed, the quantity of users, the existing communication system and its planned development, data delivery time and so on.

If the time of data delivery to the user is 15-30 minutes, PKI are equipped with the following devices: for a data transmission volume of up to 50,000 symbols/day--T-63 and F1100 telegraph apparatus; when the volume of transmitted data is up to 500,000 symbols/day--TAP-3 and TAP-34 data transmission devices.

To reduce data delivery time (to 1 minute) and to increase the volume of data transmitted (0.5-1.5 million symbols/day), PKI are outfitted with SM-1800 microcomputers, while railroad department and administration PKI are outfitted with AP-64 user stations, and YeS7920 or SM-1800.

The manning of a PKI and its terminals is determined on the basis of the hourly output norm, which includes: data preparation (on punched tape or magnetic tape)--2,500-3,000 symbols; data transmission via switched telegraph lines--3,000 symbols; data transmission via switched telephone channels--40,000 symbols; data transmission via dedicated telegraph channels--5,500-10,000 symbols; reception and transmission of messages via telephone, with verification--up to 1,000 symbols; message correction--500 symbols; visual message control--12,000 symbols; installation of punched tape for transmission and recording--15,000 symbols. Here is a sample list of positions occupied by PKI workers: senior process engineer, process engineer, process technician, senior telegraphist (operator), telegraphist (operator), electrician.

PKI service brigades are created in different sections for planned preventive maintenance and preventive repairs in terminal equipment departments of the DVTs.

Let us examine an actual PKI in relation to the devices it uses. A PKI based on a TAP-3A(B) data preparation device (Figure 1) is outfitted with T63 and F1100 telegraph apparatus and a DZM-180 alphanumeric printer. In addition MTK-2 and KOI-7 coders and decoders must be set up in order to organize direct computer data input-output and DZM-180 print-out. Data are prepared by attached line subdivisions and in the PKI itself using telegraph apparatus, and then they are transmitted via communication channels to the PKI, where they are outputted on punched tape and printed out on paper.

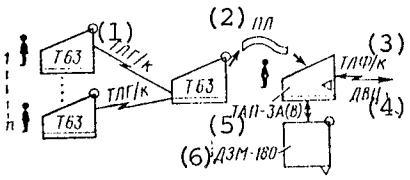


Figure 1

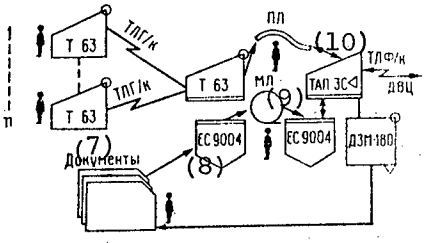


Figure 2

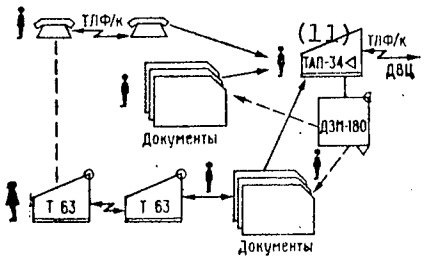


Figure 3

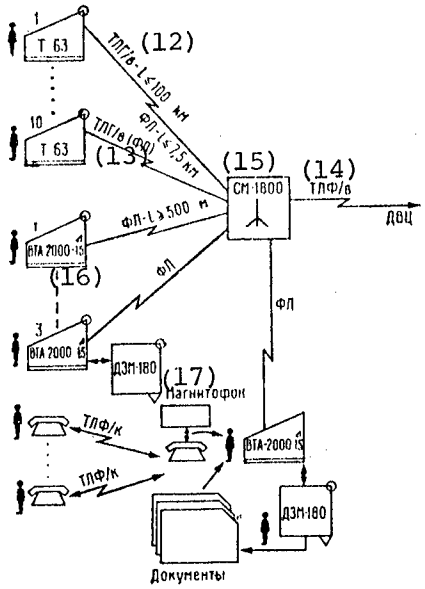


Figure 4

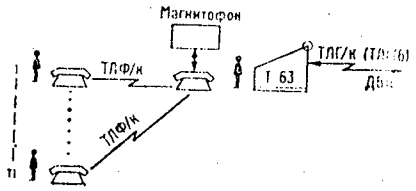


Figure 5

Key:

1. Keyed telegraph
2. Punched tape
3. Keyed telephone
4. DVTs
5. TAP-3A(V)
6. DZM-180
7. Documents
8. YeS9004
9. Magnetic tape
10. TAP-3S
11. TAP-34
12. Reserved telegraph
13. FL [not further identified]
14. Reserved telephone
15. SM-1800
16. VTA-2000-15
17. Tape recorder

Data are transmitted to and received by the DVTs by keyed telephone communication channels as follows: using an intermediate device (punched tape using the MTK-2 code) at the DVTs in point-to-point communication; directly to the computer when a coder and a decoder are connected to the TAP-3A(B).

A PKI based on a TAP-3S data preparation device (Figure 2) is set up similarly, but it is additionally supplied with YeS9002 or YeS9004 devices for data preparation on magnetic tape.

Data are collected from line subdivisions assigned to the PKI via telegraph communication channels. Data are prepared by users in the PKI itself with a YeS9002 or YeS9004 device for data preparation on magnetic tape. Data are transmitted to the DVTs via keyed telephone communication channels, directly to the computer. Data from the DVTs are received by a PKI by way of a TAP-3S using a DZM-180 alphanumeric printer and magnetic tape.

Data are transmitted from the PKI to users in attached line subdivisions via telephone, or they are prepared for telegraph apparatus and subsequently transmitted by telegraph channels.

A PKI based on a TAP-34 intelligent terminal (Figure 3) is supplied with one or several TAP-34 depending on the selected procedures for working with the telegraph apparatus, a telephone equipped with headphones and a microphone, and a tape recorder with which to record data transmitted via telephone.

Such a PKI basically services users of a line subdivisions in which a TAP-34 is installed, and users of low-activity stations having a data transmission volume within up to 2,000-3,000 symbol/day. Data are fed directly into the TAP-34 from the primary document or a printed-out teletype message using a video terminal. When data are received via telephone, the operator feeds the data into the TAP-34 as he listens.

Data are transmitted to the DVTs by way of a TAP-34 using keyed telephone communication channels, a floppy disk storage device and a video terminal or photoelectric reader. Data are transmitted from the PKI to users, or they are prepared for telegraph apparatus and then transmitted by the latter.

A PKI based on a data concentrator (Figure 4) is equipped with an SM-1800 microcomputer and terminal equipment, while PKI users are furnished with VTA-2000-15 video terminals, DZM-180 alphanumeric printers (up to 4) and T-63 and F1100 telegraph apparatus (up to 10).

In contrast to the preceding variants of PKI organization, when an SM-1800 microcomputer is employed the user feeds data directly into the microcomputer from his terminal. Transmission of data to the DVTs, receipt of replies to inquiries or acknowledgments for transmitted messages, and delivery of messages to the user are functions of the concentrator. This significantly reduces the data delivery time. Only dedicated channels are required to connect terminals to users.

When users at low-activity stations must be connected via keyed telegraph communication channels or via telephone to a PKI, a separate terminal--a VTA-2000-15--must be provided (for work with users transmitting data via telephone); in all other cases telegraph apparatus must be furnished to the operator communicating with users and the microcomputer.

A data point (IP) based on T63 or F1100 telegraph apparatus (Figure 5) is furnished with a tape recorder and a telephone equipped with headphones and a microphone. The IP is organized to collect and transmit data from remote low-activity line subdivisions. The IP operator receives data via telephone, prepares them for telegraph transmission and transmits them via a keyed telegraph communication channel directly to the DVTs computer.

11004
CSO: 1863/221

SHIPPING LINE'S MANAGEMENT SYSTEM PLAGUED BY COMPUTER PROBLEMS

Moscow VODNYY TRANSPORT 28 Mar 85 p 3

[Abstract] The author discusses tasks for the further advancement of computer-aided management methods in the Baltic Shipping Line (BMP).

The author relates that large computerized systems which the shipping line has put into service make possible new approaches to managing the operations of the line and its divisions. One of these systems, which is intended for continuous scheduling of operations of the shipping line's fleet, operates on the principle of continuous planning using YeS computers and video-terminal systems. This system enables dispatchers and other personnel to obtain, on a daily basis, prospective plans for the operation of individual transport vessels and the line's fleet as a whole for the next 90 days or more.

The author goes on to discuss organizational and technical problems which he says are holding up the further development of the management information system (ASU) "Parokhodstvo" [shipping line] and of the ASU "Morflot" [merchant fleet] as a whole. In particular, problems with the technical maintenance of computers and peripheral equipment are limiting the efficiency of state-of-the-art technology with which the Ministry of the Merchant Fleet has equipped shipping lines' computer centers. The author complains that delays in the repairing of assemblies and units of computers are causing too much downtime and impeding the solution of routine problems.

Attention is called to shortcomings in centralized technical maintenance of computer technology by specialized organizations of the All-Union Computer Systems Association ("Soyuz EVMkompleks"). Limitations of its services are said to create handicaps for computer centers which are in continuous operation, such as the BMP's computer center. The author explains that the association's organizations provide services to users only on weekdays and Saturdays from 7:30 a.m. to 8:00 p.m. If problems arise at other times, the computer center's own personnel are supposed to handle the trouble-shooting. But spare parts are not available for this, since they are made available only from centralized locations of the service organizations. Moreover, the computer center has only 18 employees for round-the-clock servicing of the center's four computers, whereas computer operation requirements call for seven persons per computer.

FTD/SNAP
CSO: 1863/246

GETTING SHIP COMMANDERS TO RELY ON COMPUTERS

Moscow KRASNAYA ZVEZDA in Russian 11 Apr 85 p 2

LUSHIN, V., Captain 1st Rank, deputy commander of a naval force, Hero of the Soviet Union

[Abstract] The author discusses problems of getting commanders of naval vessels to rely on ship automated control systems (ASU) and to develop skills in their everyday use. After citing some good and bad examples of this in exercises, he addresses one of the causes of failure to rely on automated control systems. He observes: "...In the course of combat training, sometimes it seems to us that one task or another is more easily solved by the old methods. And in practice sometimes this attitude seems to be justified. [This is] because combat exercises do not always pose a genuinely tense tactical situation, because there are simplifications and a formal approach to the solution of the problems posed..."

The author goes on to make the following remarks: "The use of the ASU in navigation deserves special discussion here. The ASU cannot be viewed as something by itself, isolated from combat training work in which weapons are used. It is true that in ordinary conditions, the determining of the ship's location, the calculation of safe passing with respect to targets, keeping a set course while sailing in formations, and a number of similar tasks are rather easily solved by traditional methods. Is it necessary to assign such tasks to a machine?

"This question is not as simple as it may seem. Certainly it is not good to complicate the solution of problems in actual combat. This also is not always wise in certain situations of navigation. On the other hand, the commander must develop for himself solid skills of working in a situation that is complex to the maximum extent, where he will be totally occupied with the main problem--using weapons, and all other problems must be assigned to subordinates and to technology. It is no good waiting for special situations in order to acquire such skills; it is necessary here intentionally to complicate methods of solving simple problems.

"This is important so that from duty watch to duty watch, the commander can form and consolidate a stereotype of the obligatory use of the computer, so

that he feels uncomfortable without its data. This is also so that the need for constant dialog with electronic aids is felt as well by all duty officers, navigators and officers of the combat information post."

FTD/SNAP
CSO: 1863/289

WORK ON REPUBLIC COLLECTIVE-USE COMPUTER SYSTEM IN ARMENIA

Moscow SOTSIALISTICHESKAYA INDUSTRIYA in Russian 7 May 85 p 2

KHODZHAMIRYAN, Yu., deputy chairman of the Armenian SSR Council of Ministers

[Abstract] The author reports on work in Armenia on a republic automated management information system based on a collective-use computer system. Its purpose is to improve coordination of the work of existing computer centers and data-transmission networks, and to amass experience which can be used in the development of regional and nationwide systems.

The author notes that about 100 automated management information systems are now in operation in the republic. These systems are based on more than 270 computers, the majority of which are state-of-the-art technology. The author notes, however, that little has been done to achieve language, methodological and information compatibility among the data bases of computer centers belonging to different agencies. This is an obstacle to more efficient large-scale economic management. Moreover, ministries and agencies continue to open new computer centers, although many computers are underloaded. At the Armenian Academy of Sciences' computer center, for example, the average daily operating time of a large-capacity "YES-1045" computer is 2.4 hours, while a model "1030" computer has been completely idle lately. A similar situation is said to exist at other computer centers. A number of collective-use computer centers have been created; however, they are essentially local systems operating in isolation from one another.

The author relates that the republic collective-use computer system is being organized on the basis of the computer network of statistical reporting agencies and the computer centers of a number of republic ministries and agencies. The system of the republic Ministry of Communications and of local data-teleprocessing and transmitting systems will be used for exchanging information. These systems will be linked by means of a republic computerized message-switching system, which is now under development. The Message Switching Center in Yerevan, which has computers of the "YeS" series, will become a kind of central exchange for this system, which will be compatible with individual users' communications equipment and terminals. The creation of territorial message-switching centers is envisaged at a later date, for

the purpose of providing all of the republic's computer centers and information users with access to the system.

The first phase of the collective-use system is to be introduced this year. This phase will ensure information exchange between computer centers of the Armenian Central Statistical Administration and a number of republic ministries and agencies.

CSO: 1863/329

EXPERTS VIEW PROSPECTS, PROBLEMS OF FLEXIBLE AUTOMATED PRODUCTION

Moscow NTR: PROBLEMY I RESHENIYA in Russian 22 Jan 85 No 2 p 3

[Abstract] The lengthy article records a discussion by six experts of prospects and problems of the development of flexible automated production facilities (GAP) and their introduction in industry. The six are: corresponding member of the USSR Academy of Sciences (AN SSSR) I. M. Makarov, USSR deputy minister of higher and specialized secondary education, chairman of AN SSSR's council on the problem "Robotics and Automated Production"; academician K. V. Frolov, director of AN SSSR's Institute of Machine Science, chairman of AN SSSR's council on theory of machines and machine systems; Doctor of Technical Sciences L. I. Volchkevich, head of a chair of instruction of the Moscow Higher Technical School imeni Bauman; Doctor of Technical Sciences V. A. Kudinov, first deputy general director of the research and production association "Experimental Scientific Research Institute of Metal-Cutting Machine Tools"; and Candidate of mechnical Sciences V. N. Vasil'yev, member of the State Committee for Science and Technology, head of an administration.

The experts identify the technologies involved in the creation of GAP, and discuss the types of production for which GAP systems are ideally suited and the optimum make-up of the systems. Emphasis is placed on flexible modules of equipment from which various combinations of production lines can be assembled for specific types of products. It is said that the modular-unit design cuts by 9-10 months the time required for the creation of systems that make up GAP.

Frolov addresses the need for keeping in mind scientific problems of the formulation of a general theory of comprehensively automated production. In support of this, he says AN SSSR's Institute of Machine Science is doing research on the theory of machines and mechanisms, numerical programmed and adaptive control, technical diagnostics, robots and robotic systems, coordinate measuring machines, laser technology, and systems of control of measurement and production processes using laser beams.

Kudinov points out that one problem of existing technology is the reliability of machine systems, which he says is still not high. The cost of repairs to unreliable machines is running two to five times as much as their initial cost. He says the need for diagnostic devices has become especially acute

in this connection. Vasil'yev replies that the Institute of Machine Science has developed diagnostic methods which reveal problems in the design and technology of equipment during the stage of prototype development, but admits there is a need for a technical diagnostic system that would reveal all defects and potential problems of equipment that has already been installed.

Raising the question of having plants properly prepared for the introduction of GAP, Frolov says that manufacturers of GAP equipment specify conditions of the workplace before it is delivered, buy many users consider the preparations to be too costly and try to cut corners. As a consequence, sensitive control systems begin to fail, and plant managers complain that GAP systems are not as good as their old, multipurpose equipment. Volchkevich says that in a considerable number of production-automation projects designers have attempted to make mechanisms and devices that merely imitate the actions of human operators, and have neglected the main considerations of comprehensive automation, flexibility for adjusting to new product types, and reliability and high quality of output. He refers to 'half-baked' systems which have required 20-30 times as much capital expenditure, but have raised efficiency by only about 2.5 times.

FTD/SNAP
CSO: 1863/289

UDC 681.324

SELECTION OF THE CONFIGURATION OF A DISTRIBUTED COMPUTER NETWORK

Kiev MEKHANIZATSIYA I AVTOMATIZATSIYA UPRAVLENIYA in Russian No 1, Jan-Mar 85 (revised manuscript received 24 Jul 84) pp 33-37

[Article by Candidate of Economic Sciences A. M. Gostrik and Candidate of Economic Sciences Ye. Ya. Karpovskiy]

[Text] Improvement of the national economy's control presupposes creating distributed computer networks on the basis of the wide use of microprocessor technology and microcomputers. This brings on the problem of selecting the configuration of a network consisting of n microcomputers with links $\gamma_{ij} \in \Gamma$ among them.

The problem of selecting the configuration of a distributed computer network may be stated as

$$H \longrightarrow \min_{(n,\ \Gamma)} \tag{1}$$

$$T_{\text{p}} \longrightarrow \min_{(n,\ \Gamma)} \tag{2}$$

$$Q_{\text{a}} = 1 - P_a \longrightarrow \min_{(n,\ \Gamma)} \tag{3}$$

$$C \longrightarrow \min_{(n,\ \Gamma)} \tag{4}$$

given the constraint

$$T_{\text{p}} \leqslant T_{\text{p}}^{D}, \tag{5}$$

where H--labor-intensiveness of the control tasks (outlays of manual labor), man-hours; T_{p}--mean data processing time required for a control decision, hr; P_a--probability of correct one-time completion of a complex of algorithms for solving control problems in the distributed computer network; C--corrected outlays on creating and operating the distributed computer network, thousands of rubles; T_{p}^{D}--permissible maximum mean time to generate a control decision, hr.

Thus selection of the configuration of a distributed computer network is a four-dimensional variable-scale vector problem of optimizing numerous functions in the presence of severe contradiction of criteria [1]. The following formulas are used to calculate the particular indicators Q_a and T_p [2]:

$$Q_a = 1 - P_a = 1 - \exp\left[-P_0^t(I-P)^{-1}R\right];$$

$$T_p = P_0^t(I-P)^{-1}M,$$

where P_0--vector reflecting the initial state of the control algorithm; $P=\|p(i, j)\|$ --matrix of probabilities $p(i, j)$ of transition from node i to node j of the graph of the control algorithm; $M=\{m_i\}$--vector of the average times for performance of the functions of the operators of the algorithm associated with node i; $m_i=\sum_{k\in I}\tau_k$; R --vector of the probabilities of error-free fulfillment of the operators of the algorithm associated with graph node i; τ_k--mean time to complete k operators of the algorithm with a computer; t--vector transposition operator.

The numerical value of mean outlays of manual labor can be determined from the expression

$$H = \frac{1}{D}\sum H_i,$$

where H_i--average outlays of manual labor to carry out control task i; D--quantity of tasks carried out by the distributed computer system.

The problem of selecting the configuration of the distributed computer system is solved using factor analysis and the main components method. Let N variants for which constraint (5) is satisfied be selected from among N_B possible variants of the configuration of the distributed computer network as a result of preliminary analysis. The estimates of particular effectiveness criteria x_{ij} ($i=\overline{1,N}$ $j=\overline{1.4}$) of each of the N variants forming a $4\times N$ matrix are known. The following formula is used to normalize and center variables x_{ij} ($i=\overline{1,N}$, $j=\overline{1.4}$):

$$y_{ij} = \frac{x_{ij}-M(x_i)}{\sigma(x_i)}, \quad i=\overline{1,N}, \; j=\overline{1,4},$$

where $M(x_i)$ and $\sigma(x_i)$ are respectively the mathematical expectation and standard deviation of x_{ij}.

Let us construct the matrix $Y=\|y_{ij}\|$ that will serve as the starting point for calculating matrix $R=\frac{1}{N}YY^t$. In this case $R\neq I$, where I is a unit matrix. This inequality allows us to postulate presence of an averaged quality criterion for the distributed computer network; this quality criterion can

be gaged by latent factors (the main components) reflecting the presently evolved scientific level of development of distributed computer networks, the sophistication of engineering concepts and the accomplishments of the theory and practice of creating hardware for distributed computer networks.

In order to determine latent factors using the main components method, we calculate: diagonal matrix Λ of eigenvalues $\lambda_r \in \Lambda$ (the characteristic equation $|R - \lambda_r I| = 0$ is solved) and the r-th eigenvectors $U_r \in V$ of matrix R. The sought matrix of main component weights equals $A = U\Lambda^{1/2}$, and the values of the main components are calculated using the formula

$$F_r(i) = \frac{1}{\lambda_r} \sum_{j=1}^{4} a_{rj}\, x_{ij}. \tag{6}$$

Using the expression $v_r = \sum_{j=1}^{4} a_{jr}^2$, we determine the contribution of the main components to the overall dispersion of the effectiveness indicators of the different configuration variants of the distributed computer network under analysis. For main component g, for which $v_g = \max_{\{r\}} v_r$, we seek a certain variant of distributed computer network configuration (k) in relation to which

$$F_g(k) = \min_{\{i\}} F_g(i). \tag{7}$$

This variant k is adopted as the solution to the problem of multicriterion selection (1) - (5).

The justification for the proposed method can be found primarily in the fact that inequality $R \neq I$ is fulfilled. The accuracy of the problem's solution is determined by the weight v_g of main component g. In most practical calculations, $v_g > 0{,}65$ (accuracy is not less than 80 percent--that is, it is adequate in comparison with informal (objective) selection methods based on expert assessments). The range of applicability of the proposed method is limited by Thurstone's formula, which in this case takes the form $N \geqslant 16$.

When $N < 16$, or when $N \geqslant 16$ but $v_g < 0{,}65$, problem (1) - (5) should be solved by using the method of minimizing the sum of the deviations of the particular effectiveness indicators of the distributed computer network from ideal values $E_{id} = \{1, 1, 1, 1\}$,--that is, by seeking a variant k such that

$$E_k = \min_i \sum_{j=1}^{4} (1 - E_{ij}), \tag{8}$$

where

$$E_{i1} = \{\min_{\{i\}} H_i\}/H_i; \tag{9}$$

$$E_{i2} = \{\min_{\{i\}} T_{pi}\}/T_{pi}; \tag{10}$$

$$E_{l3} = \{\min_{\{l\}} Q_{ai}\}/Q_{ai}; \quad (11)$$

$$E_{l4} = \{\min_{\{l\}} C_l\}/C_l. \quad (12)$$

Use of this method foresees presence of a certain number of variants N of distributed computer network configurations, in the formation of which it would be suitable to utilize the structural properties of the morphological graph G_M, the nodes of which are elements $z_{ls} \in Z$ of a morphological block, where z_{ls} is the s-th hardware type intended for the l-the production operation (function) involving tasks of the distributed computer network $(l = \overline{1, M}, s = \overline{1, m_l})$. The node corresponding to element z_{ls} is associated by its edges to all nodes of graph G_M that are associated with elements $z_{l-1,s}$ and $z_{l+1,s}$ of lines $(l-1)$ and $(l+1)$ of the matrix $Z = \|z_{ls}\|$. Any path $i_{s_1} \ldots s_l \ldots s_M$ passing from any node z_{1s} of the graph through node $z_{ls}\,(1 < l < M)$ to some node z_{MS} gives us an arbitrary (i-th) variant of the configuration of the distributed computer network. The following algorithm is proposed for formal generation of different variants using the morphological graph G_M:

1. For every line l of matrix Z, write the disjunction

$$Z_l = \bigcup_s z_{ls}; \; l = \overline{1, M}, \; s = \overline{1, m_l}.$$

2. In accordance with the rules of logical multiplication and addition, form the set of configuration variants of the distributed computer network:

$$\Omega = \& z_l = \&(\bigcup_s z_{ls}), \; |\Omega| = N.$$

Table 1. Raw Data for Selecting a Variant of the Configuration of a Distributed Computer Network

Raw Data in Relation to Different Groups of Tasks	Groups of Tasks		
	Automated Design System	Accounting-Analytical	Summary-Analytical
D_i, each	83	8	3
H_i, man-hours	5.0	3.7	2.1
$\tau_{\rho i}$, min	21.6	9.2	5.3
τ_i, mo	12	1	3
τ_k, μsec	2.2	2.2	2.2
ρ_k	$1 - 10^8$	$1 - 10^8$	$1 - 10^8$

Note: $\tau_{\rho i}$--average time to carry out task i; ρ_k--probability of correct computer fulfillment of operator k of the task algorithm.

Table 2. Specific Variants of the Configuration of a Distributed Computer Network, Values of Particular Effectiveness Criteria and General Optimization Indicators

(1) № конкурентоспособного варианта	(2) Значения частных показателей эффективности				Значения скалярных критериев оптимизации (6)	
	H, чел.-ч (3)	$T_р$, ч (4)	$Q_а \cdot 10^4$	C, тыс. руб. (5)	F_i	E_i
1	12,08	1,22	3,8	21	6,579	1,484
2	9,2	0,8	5,1	42	2,402	2,0
3	7,2	0,74	7,4	42	0,777	2,065
4	9,84	1,12	4,9	42	3,981	2,097
5	6,00	0,63	9,8	49	—0,982	2,075
6	7,2	0,72	5,6	49	0,726	1,97
7	6,92	0,92	4,6	49	1,549	1,888
8	9,2	0,82	4,2	49	2,338	1,928
9	7,6	0,76	6,4	49	0,924	2,101
10	6,96	0,87	4,1	49	1,469	1,78
11	4,8	0,43	11,2	63	—3,211	1,842
12	6,28	0,64	9,1	63	—1,296	2,18
13	7,6	0,73	4,6	63	0,526	1,962
14	7,12	0,93	3,7	63	1,234	1,813
15	5,4	0,47	10,3	63	—2,6	1,965
16	4,4	0,35	6,2	84	—3,707	1,443
17	4,2	0,27	4,2	91	—4,048	0,912
18	4,68	0,41	5,3	84	—3,173	1,515
19	7,2	0,72	3,6	91	—0,755	1,811
20	5,2	0,44	4,9	84	—2,732	1,594

Key:
1. Number of competitive variant
2. Values of particular effectiveness indicators
3. Man-hours
4. hr
5. Thousands of rubles
6. Values of scalar optimization criteria

The methods for generating the different configuration variants of a distributed computer network and carrying out multicriterial selection on the basis of the main components method were used in the practical development of a distributed computer network for a regional agroindustrial association. The Elektronika-100/25 microcomputer was chosen as the basic computer module. The distributed computer network is intended to carry out 94 tasks. Table 1 shows the quantitative characteristics of these classes of tasks; these characteristics were used as the raw data for calculating the particular effectiveness criteria for the different configuration variants of the distributed computer network.

Twenty competitive variants of distributed computer networks, the values of the particular effectiveness criteria of which are shown in Table 2, were arrived at by using different methods of backing up the basic computer module (for which $C = 21{,}000$ rubles) and for building hierarchical computer structures. The values of H_i and C_i were arrived at by direct calculation. The values of T_{pi} and Q_{ai} were determined by simulation modeling using the procedures

described in [2]. An algorithm making it possible to obtain the components of the vector of factorial loads $\{a_{1g}/\lambda_g,\ a_{2g}/\lambda_g,\ a_{3g}/\lambda_g,\ a_{4g}/\lambda_g\}$, among which the vector {0.968, 0.969, -0.439, -0.842} corresponds to the maximum eigenvalue $\lambda_1 = 0.695$, was developed in order to calculate the values of the scalar optimization criterion given in Table 2. Taking (7) into account, the expression used to determine the general optimization criterion $Fg(k)$ assumes the form

$$F_g(k) = 0{,}968\,H_i + 0{,}969\,T_{pi} - 0{,}439\,Q_{ai} - 0{,}842\,C_i.$$

For comparison Table 2 shows the results of using expressions (9) - (12) to calculate the sums E_i of the deviations of the particular effectiveness indicators of variant i of the distributed computer network configuration from ideal values $E_{id} = \{1, 1, 1, 1\}$. The solution was the same for both methods--variant 17, Table 2. In comparison with (8) - (12), the proposed method for selecting the configuration of a distributed computer network makes it possible to reveal correlations between particular effectiveness indicators and to use them to derive the general optimization criterion.

BIBLIOGRAPHY

1. Rua, B., "Problems and Methods of Decision Making in Tasks Involving Multipurpose Functions," in "Voprosy analiza i protsedury prinyatiya resheniy" [Problems of Analysis and Decision Making Procedures], Moscow, Mir, 1976, 312 pp.

2. Karpovskiy, Ye. Ya., "Nadezhnost' spetsial'nogo matematicheskogo obespecheniya upravleniya" [Reliability of Special Control Software], Kiev, Vishcha Shkola, 1982, 152 pp.

11004
CSO: 1863/259

UDC 519.856

NECESSARY AND SUFFICIENT CONDITIONS FOR CONVERGENCE OF ITERATIVE PROCEDURES

Kiev DOKLADY AKADEMII NAUK UKRAINSKOY SSR SERIYA A FIZIKO-MATEMATICHESKIYE I TEKHNICHESKIYE NAUKI in Russian No 2, Feb 85 (manuscript received 23 Jan 84) pp 64-65

[Article by L. G. Bazhenov, UkSSR Academy of Sciences Institute of Cybernetics, Kiev, presented by Academician V. S. Mikhalevich]

[Text] Of great theoretical interest is studying conditions for convergence of iterative procedures in mathematical programming. It should be noted that the evolution of nonlinear programming methods has been governed to a considerable extent by the capabilities of studying the convergence of them. The necessary and sufficient conditions for convergence of algorithms are derived in [1] for relaxation methods. In connection with the spread of nondifferentiable optimization problems, nonrelaxation methods, not guaranteeing reduction of the efficiency function in each step, are now attracting attention. At the same time, studying the convergence of nonrelaxation procedures is considerably difficult. Sufficient conditions for convergence of nonrelaxation methods which allowed obtaining a number of new results for nondifferentiable optimization problems were derived in [2]. These conditions were modified in [3].

In this article, necessary and sufficient conditions are presented for convergence of nonrelaxation procedures in mathematical programming. Let us consider the nonlinear programming problem

$$\min f(x) \text{ when } x \in X \subset R^N \qquad (1)$$

Let us define, following [1], the concept of an algorithm designed to solve problem (1).

Definition. For all $s \geq 1$, let us define the set $X_s \subset X$ and the algorithmic mapping $A_s: X_s \to X_{s+1}$. Let us assume that with a given $x^1 \in X_1$, x^2, ..., x^s are generated. If $A_s(x^s)=\emptyset$, the procedure is halted. Otherwise, the next point x^{s+1} is selected from the relation $x^{s+1} \in A_s(x^s)$.

Let there be given some set $X^* \subset X$. Let us call any point in the given set X^* a convergent point. Let us consider an algorithm for finding the convergent points. For the given problem and set of points $X^* \subset X$, let us call an algorithm convergent if it meets these conditions:

1. If an algorithm stops the search at point x, this means that either there is no convergent point or point x is convergent.

2. Let us assume that an algorithm generates an infinite sequence of points, none of which is convergent. Then if all the points are not part of a compact set, set X^* is empty; but if all the points are part of a compact set, the limit of any converging subsequence belongs to X^*.

Let us define in set X the function $d:X \longrightarrow R^1$ the following way:

$$d(x)=\inf \left\{ ||x-y|| : y \in X^* \right\}$$

Theorem. Let there be for the given problem (1) an algorithm, which generates points x^s, and a set of convergent points $X^* \subset X$.

Let us consider the following conditions 1 and 2.

Condition 1. If for some x and s, set $A_s(x)=\emptyset$, then either the algorithm indicates that x is a convergent point or that $X^*=\emptyset$.

Let us assume that an algorithm generates an infinite sequence of points, none of which is convergent. Then if $X^* \neq \emptyset$, there must be a compact set $X_0 \subset X$ such that $x^s \in X_0$ for all s.

Condition 2. Let an algorithm generate an infinite sequence of points, none of which is convergent, and let all points x^s be part of compact set X_0. Then there is a continuous function $W:X_0 \longrightarrow R^1$ which meets the requirements:

Z1. For any $\delta > 0$, there is a number N such that for any s>N there is a number $L_s>0$, such that for all $1 > \text{or} = L_s+s$

$$W(x^1) < \text{or} = W(x^s)+\delta .$$

Z2. For any $\alpha > 0$, there is a number N such that for any s>N and such that $d(x^s) > \text{or} = \alpha$, there are $\gamma > 0$ and a number $m_s > s$ such that

$$W(x^{m_s}) < \text{or} = W(x^s)-\gamma .$$

If set X^* is closed, then any algorithm converges then and only then when it meets conditions 1 and 2.

Proof of the theorem is based on the following lemma.

Lemma. Let $W(x)$ be a continuous function of $W:X \longrightarrow R^1$. Let us assume that for any $\delta > 0$ there is a number N such that for any $s > N$ there is a number $L_s > 0$, such that for all $l >$ or $= L_s + s$

$$W(x^l) < \text{or} = W(x^s) + \delta.$$

In addition, let us also assume that for some sub-sequence $\left\{ x^{n_k} \right\}$, $x^{n_k} \longrightarrow x'$. Under these assumptions

$$\lim_{s \to \infty} W(x^s) = W(x').$$

SUMMARY. The necessary and sufficient conditions for convergence of iterative procedures in mathematical programming are proved.

BIBLIOGRAPHY

1. (Zangvill, U. I.), "Nonlinear Programming," Moscow, Sov. radio, 1973, 312 pages.

2. Nurminskiy, Ye. A., "Chislennyye metody resheniya determinirovannykh i stokhasticheskikh minimaksnykh zadach" [Numerical Methods of Solving Determinant and Stochastic Minimax Problems], Kiev, Nauk. dumka, 1979, 159 pages.

3. Norkin, V. I., "On Conditions for Convergence of Combined Algorithms in Nonlinear Programming," KIBERNETIKA, No 6, 1979, pp 73-77.

8545
CSO: 1863/247

UDC 681.3.06

SIMPLE SOLUTION TO PROBLEM OF STACK AUTOMATA SYNTHESIS

Kiev DOKLADY AKADEMII NAUK UKRAINSKOY SSR SERIYA A FIZIKO-MATEMATICHESKIYE I TEKHNICHESKIYE NAUKI in Russian No 2, Feb 85 (manuscript received 17 Jan 84) pp 68-69

[Article by I. V. Red'ko, Kiev State University, presented by I. I. Lyashko, Academician, UkSSR Academy of Sciences]

[Text] There are quite a few different solutions to the problem of stack automata synthesis. But they are all poorly structured and very unwieldy.

Presented in this work is a simple and natural solution to this problem. It is a composition of three procedures performed successively. The first builds from any context-free grammar a special recursion specifying the same language as the source grammar; the second builds from any such recursion a quasi-regular expression, defined below, specifying a language defined by this recursion. Finally, the third procedure by induction from the quantity of operations of the quasi-regular expression synthesizes an automaton with stack memory representing the language specified by this expression.

1. Construction of Recursion from Context-Free Grammar. Let $G=(N, T, P, S)$ be any context-free grammar (N and T are nonterminal and terminal alphabets, $N \cap T=\emptyset$), and $L(G)$ is the language produced by grammar G [1]. Let us build a recursion [2] which specifies language $L(G)$. For this purpose, let us recall its definition.

Let $L_1, \ldots, L_n$ be any languages in alphabet $V=N \cup T$, where $N=\{A_1, \ldots, A_n\}$. By recursion is meant that n-ary algebraic operation which assigns to a sequence of languages $L_1, \ldots, L_n$ a language in alphabet T designated $A_i ::= L_i (i=\overline{1,n})$ and constituting $\bigcup_{k=0} L^{(1)}_k$, where $L^{(i)}_0=\emptyset$, $(i=\overline{1,n})$, and $L^{(i)}_{k+1} = L_i(L^{(1)}_k, \ldots, L^{(n)}_k)$. Here $L_i(L^{(1)}_k, \ldots, L^{(n)}_k)$ is an ordinary superposition of languages $L^{(1)}_k, \ldots, L^{(n)}_k$ on language L_i ($A_1, \ldots, A_n$ in nonterminal symbols.)

Let us now compare each nonterminal symbol A_i of grammar G to language $L_i=\{\alpha \mid A_i \longrightarrow \alpha$ and $A_i \longrightarrow \alpha \in P\}$ and let us consider the recursion $A_i::=L_i(i=\overline{1,n})$. It is easy to see that a recursion so constructed specifies language L(G).

2. Construction of Quasi-Regular Expression from Recursion. By a quasi-regular expression is meant a correctly constructed expression of a sub-algebra of a quasi-regular algebra produced by base languages, i.e. by the aggregate of languages consisting of an empty language or languages consisting of an empty word, or single-letter words. In the process, by a quasi-regular algebra [2] is meant an algebra, the medium of which is a set of languages in a specified alphabet, and the operations: theoretical-set union, multiplication (in the sense of multiplication of subsets of a free semi-group) and quasi-iteration (unary recursions).

Construction of a quasi-regular expression from a given recursion is based on the following formula [2]:

$$\begin{cases} A_1 ::= L_1 \\ \cdots\cdots \\ A_n ::= L_n \end{cases} = \begin{cases} A_1 ::= L_1 \\ \cdots\cdots \\ A_{n-2} ::= L_{n-2} \\ A_{n-1} ::= L_{n-1}(A_1, \ldots, A_{n-1}, A_n ::= L_n) \end{cases}$$

In fact, this formula allows reducing the n-arity of recursion by one. Therefore, no matter what n-ary the recursion, by applying this formula to it sequentially n-1 times, we will come to unary recursion (quasi-iteration), i.e. to a quasi-regular expression.

3. Synthesis of Stack Automaton from Quasi-Regular Expression. Let there be given any, but fixed, quasi-regular expression. Let us construct an automaton with stack memory which represents the language defined by this expression. For this purpose, from the set of different definitions of the concepts of a stack automaton and representation, it is convenient to select the following.

By a stack automaton or automaton with stack memory (MP-automaton), let us understand the ordered seven

$$M = \langle K, \Sigma, \Gamma, \delta, z_0, q_0, F \rangle,$$

where K is a non-empty finite set of states, Σ is a non-empty finite set of input symbols, Γ is a non-empty finite set of symbols of stack memory, δ is the mapping of the set $K\times(\Sigma\cup\{\varepsilon\})\times\Gamma$ to the set of all finite subsets of the set $K\times\Gamma^*$, $q_0\in K$ — is the initial state, $Z_0\in\Gamma$ is the marker of stack memory, and $F\subseteq K$ is the set of final states.

Functioning of the MP-automaton is specified by transitions from one configuration (p, ω, α) to another configuration (p', ω', α'), where a transition relation $\models$ is defined as a reflexive-transitive closing of a direct transition relation $\vdash$.

Two configurations $(p, x\omega, \alpha z)$ and $(q, \omega, \alpha, \gamma)$ are in the relation $\vdash$, then and only then, when $(q, \gamma) \in \delta(p, x, z)$.

It is said that a language L(M) is represented in an MP-automaton M, if $L(M) = \{\omega \mid (q_0, \omega, z_0) \models (q, \varepsilon, d)$ for certain $q \in F$ and $\alpha \in \Gamma^*\}$.

Let us implement inductively the construction of an MP-automaton from a given quasi-regular expression.

For base languages representing them, MP-automata are built trivially.

Let there be now the already constructed MP-automata M_1 and M_2, representing languages L_1 and L_2, respectively. Then MP-automata representing languages $L_1 \cup L_2$ and $L_1 \cdot L_2$ are very simply constructed, where $\cup$ is a theoretical-set union, and $\cdot$ is multiplication. In fact, while not limiting generality, one can assume that the initial states of MP-automata M_1 and M_2 are not final. But then to construct the MP-automata mentioned it is sufficient to "bond together" the initial states of MP-automata M_1 and M_2, and in the second, to "connect" the final states of M_1 to the initial state of M_2.

Finally, let us consider the last case. For this purpose, let us assume that an MP-automaton M, representing any, but a fixed, language L is given. Let us show that constructing a new MP-automaton representing quasi-iteration of language L from M presents no labor, as in the preceding cases. Actually, for this it is sufficient to organize a "reset" to the initial state in each state of MP-automaton M, from which there is a transition by the action of a nonterminal symbol and specified quasi-iteration.

Let G be any context-free grammar, R be the recursion constructed in section 1 from G, Q be the quasi-regular expression constructed in section 2 from R, and finally, M be the MP-automaton constructed from Q by the method presented in section 3. Then it follows directly from the construction given above that there is a

Theorem. L(G)=L(M).

SUMMARY. A simple procedure is given for push-down automatic machine synthesis. It is a composition of three procedures performed successively.

BIBLIOGRAPHY

1. (Ginsburg, S.), "Mathematical Theory of Context-Free Languages," Moscow, Mir, 1970, 250 pages.

2. Red'ko, V. N., "Some Problems of Theory of Languages," KIBERNETIKA, No 4, 1965, pp 53-65.

8545
CSO: 1863/247

UDC 62-501.5:621.391.272

GENERALIZATION OF THEORY OF CONDITIONALLY OPTIMUM FILTERING FOR CASE OF NON-WHITE NOISE IN OBSERVATIONS

Moscow AVTOMATIKA I TELEMEKHANIKA in Russian No 4, Apr 85
(manuscript received 2 Feb 84) pp 69-76

SILUYANOVA, I. D.

[Abstract] Theoretical analysis and practical computations have shown that in problems with non-white noise the quality of filtering in some cases is not sufficiently good, thereby necessitating the development of more efficient filters. In an earlier article (AiT, No 10, pp 61-69, 1980) the author extended the conditionally optimal evaluation method developed by V. S. Pugachev ("Determination of State and Parameters of Continuous Nonlinear Systems," AiT, No 6, pp 63-79, 1979) to a case when the noise in observations was not white noise, but was determined by a stochastic differential equation. In this article the author proposes an approach for solving problems of the presence of non-white noise in observations which involves differentiation of the observed signal prior to the appearance of white noise and the intooduction of all these derivatives into the equations for the admissible filters. The validated method is applicable for a great number of practical problems in which observation noise with an adequate accuracy can be approximated by a random function. References 12: 10 Russian, 2 Western.
[287-5303]

CALL FOR IMPROVING AUTOMATION TRAINING AT HIGHER SCHOOLS (LETTER)

Moscow PRAVDA in Russian 31 Mar 85 p 2

[Article by G. Khutskiy, Doctor of Technical Sciences, Professor (Minsk)]

[Text] When they were students, engineers who are now employed at design bureaus, industry institutes and enterprises did not study robotic systems and flexible automated production facilities. Consequently, they know nothing about them. As is known, the economy does not derive full benefit from even the most sophisticated machine if it is in incompetent hands.

The USSR Ministry of Higher and Specialized Secondary Education (Minvuz SSSR) has prepared a list of professional areas and fields of specialization for higher schools, for the purpose of training engineers in the fields of robotics, microprocessor technology, and systems for automated designing of flexible production units. Curricula and syllabi have been drafted. A number of higher schools have begun actively to train students in line with them. The Moscow Machine Tool Building and Tool Institute and the Moscow Higher Technical School imeni Bauman became the first to do this.

A school, "Robots and Robotic Systems", was created also at our own Belorussian Polytechnical Institute about two years ago. Instruction was provided in three specialized fields at first, and another three were subsequently added. The first group of robotic engineers will graduate from the institute in 1986. The training of engineers for flexible automated production facilities is in progress in two other schools. As a result, the institute will be able to present diplomas to engineers in eight of the 12 specialized fields defined by Minvuz SSSR's list.

A far larger number of such specialists is required. However, dozens of higher schools continue to train engineering specialists in machine-building technology and the production of metal-cutting machine tools and tools using old curricula. The fact that not all enterprises are transferring to automation at once is cited in defense of this. Those who are not familiar with robots will also find work, it is said.

As for the true causes of this situation, it should be acknowledged that officials of higher schools are uneasy about the prospect of having to retrain their instructors, upgrade their qualifications and replace laboratory equipment with new and sometimes costly equipment. This is a troublesome business, to be sure. But one cannot go on doing things the old way.

Inertia is making itself felt in republic ministries of higher and specialized secondary education. They are not requiring higher schools under their jurisdiction to revise their practices as the times require. As a result, young engineers now have to be retrained on the job immediately after graduating from higher schools. And this certainly is not the best variant for training specialists.

FTD/SNAP
CSO: 1863/246

TASKS OF NEW DRIVE TO PUT COMPUTERS IN CLASSROOMS

Moscow IZVESTIYA in Russian 3 Apr 85 p 2

[Excerpt] At a [recent] meeting of the Politburo of the Central Committee of the Communist Party of the Soviet Union, measures were outlined for ensuring computer literacy of students of secondary educational institutions, and for the broad introduction of computer technology in the educational process. A resolution of the Party Central Committee and of the USSR Council of Ministers was adopted on this question. It calls for the introduction, beginning with the new school year, of a course "Fundamentals of Information Science and Computer Technology" at all of the country's secondary educational institutions.

Prof. V. Monakhov, director of the USSR Academy of Pedagogical Sciences' Scientific Research Institute of Content and Methods of Instruction, related the following about how the introduction of the new discipline is being organized:

"This subject will appear in the schedules of students of the higher grades beginning September 1 of this year. But I should mention right away that the course is not being introduced where a void exists. Many educational-industrial complexes have been training school students in operation and programming of computers.

"The main requirement of computer literacy is the training of students to use computer technology in their subsequent practical activity.

"Our institute, together with the Siberian Branch of the USSR Academy of Sciences, has developed a program of instruction and has prepared a textbook. The main goal of the new course is to help students form concepts of the rules and methods of solving problems on computers, and to develop elementary skills of using microcomputers for solving problems. In the very near future, information science and computers must become the basis for studying a whole list of school subjects on a qualitatively different level.

"Now about the textbook, 'Fundamentals of Information Science and Computer Technology'. Its content is based on three fundamental elements: information, algorithm, computer.

"All physics and mathematics instructors will undergo appropriate training before the new school year in seminars and courses, which will be offered in oblast centers. A new specialty--information science instructor--is to be introduced at teachers' colleges.

"There are especially numerous problems connected with the creation of suitable material and technical resources. The introduction of the new course will require no less than 50,000 computer rooms at schools, and hundreds of thousands of computers must be made available to education.

"Right now we are busy selecting computers that are the best suited for our schools. We are being aided in this by the experience of foreign colleagues. A conference was just held in which specialists of member countries of the Council for Mutual Economic Aid took part. Our scientists have visited Great Britain, France and Japan, where the teaching of computer fundamentals is on a high level."

FTD/SNAP
CSO: 1863/257

COMPUTER TRAINING LAGS IN SOME VUZ'S

Moscow PRAVDA 21 March 85 p 2

[Interview with Professor P. Polukhin, doctor of technical sciences and head of the Moscow Institute of Steel and Alloys, by. I. Saltykov: "Acceleration of Thought; Higher Schools--Order for Specialists."]

[Text] Many of today's freshmen, after finishing VUZ's, will have to work at enterprises and shops filled with computerized and automatic devices. Will they be prepared to deal skillfully with new technology? How can computer education be obtained for future specialists? This is the subject of conversation between our correspondent and Professor P. Polukhin, doctor of technical sciences and Hero of Socialist Labor. The Moscow Institute of Steel and Alloys, which he heads, was the first among engineering VUZ's of the country to introduce early mastery of computers by students.

[Question] Petr Ivanovich, first of all, please explain what is meant by "early mastery."

[Answer] The fact is, school graduates (excluding those from specialized schools) have not up to now received knowledge about computer technology. Therefore, freshmen master the course, "Computer Programming and Numerical Methods." A large part of these lessons is allotted to laboratory work. The students consolidate the knowledge and skills they receive by solving problems suggested in the study of fundamental and general engineering disciplines. Dialog with a computer soon becomes habitual. Freedom from routine calculation work allows young people to concentrate on what is important: to master more deeply the natural laws and facts accumulated by science, to study them, to analyze, and to compare them with one another. Thereby, a strong basis is built for forming specialists of a broad profile.

Second year students become acquainted with methods for controlling modern equipment using microprocessors. The next important discipline for the engineers, "The Organization of an Experiment," is taken a year later.

Metal technology has always flourished at the juncture of theoretical forecasting and practical experience and not, I am not afraid to say, art. Now, the process of creative research has sharply accelerated and, in place

of the former trial and error that often went on for years, there is experimentation that is compressed to the maximum in time through the use of computers, relying on mathematical statistics, mathematical planning, and automation. The course, "The Organization of an Experiment" was also called upon to teach the future engineer the strategy and tactics of modern research.

The "fine-tuning" of computer education takes place in the specialized departments, taking into account the concrete problems that graduates will have to solve after finishing the VUZ. These years are filled with intensive and varied us of computers in fulfilling academic tasks and course projects. But all this has become possible only on the basis of early mastery of computers by students.

I would like to call your attention to one circumstance that, in my opinion, is very important: in the transition of VUZ's to providing computer literacy to their charges, the USSR Ministry of Higher and Secondary Specialized Education has permitted our institute to grant a certificate of computer mastery along with the engineering diploma. The decision on its issuance is made by the state examination commission during the defense of the thesis project. To earn the "second diploma" is not easy, even for those who should successfully pass examinations in all disciplines that we have now introduced for "computer education." Many graduates still do not possess such certificates. The exacting requirements that appear in this regard are necessary: our personnel requestors must be sure that the specialists coming to them know how to apply the knowledge they have received creatively to the advantage of the production plant.

[Question] Petr Ivanovich, it is known that a teacher can impart to the student only that knowledge that he himself has perfectly mastered. However, a majority of teachers in higher schools studied in VUz's when cybernetics and its applications remained the lot of a relatively small circle of specialists. Now they have had to re-learn how to associate with computers "hands on." What difficulties have been encountered in eliminating the original gap?

[Answer] Really, this gap has not yet been overcome completely. Although a majority of our teachers are specialists of very high qualifications and many have made serious contributions to the development of metallurgy, I confess, others have been confused by the necessity to search for a common language with machines. They had to introduce, through their own efforts, computer teacher "illiteracy liquidation."

Teachers have begun to be taught in a 300-hour program in our institute in the faculty for raising qualifications. In addition, specialized courses have been introduced on programming and methods for developing dialog software for minicomputers. Already, 180 persons have undergone such special training, approximately a third of our teaching collective.

But every teacher in general engineering and special departments must master computer technology somehow. And yes, for representatives of the physical sciences it is a sin to lag behind their wards in ability to

establish contact with computers. Therefore, we are now introducing a certificate for teachers for mastering computer technology.

It is evident that under present conditions a number of VUZ collectives appear not to be able to solve such a problem alone. Moreover, it cannot be left to the discretion of rectorates or to the initiative of individual VUZ's. The system for raising the qualifications of teachers in the area of computer use should becoome universal. Apparently there will have to be reorientation or new openings of faculties for raising qualifications in VUZ's which are well equipped with computers.

[Question] The system of applying computers in the academic process as proposed by your institute has received broad recognition in domestic higher schools. However, from one of the rectors acquainted in detail with your experience, the following happened to be overheard: "Give them honor, of course, and glory...But what can be done for VUZ's that do not have such a powerful stock of computer equipment?" Other specialists are expressing a similar point of view. What do you say about this?

[Answer Apparently, there is a basis for such apprehensions. Acquainting students with modern computers when limited to lectures and seminars means seriously complicating the solution to computer education problems.

Therefore, the question of supplying VUZ's with modern computer technology has become more acute than ever. It is not possible to procrastinate any more.

True, USSR and republic ministries of higher and secondary specialized education are trying as much as they can to help, but their possibilities are limited.

Sometimes the question arises as to the effectiveness of expenses for computers in the educational process. Here, you see, there is no direct rationalization of production and there is no immediate return. The machine is "built into" the educational process, and therefore, the profit from it is something different: it must be calculated not in rubles but in the growth of specialists' knowledge and in the new quality of thinking which turns out in the final analysis to be super-profitable for the state. After all, no one would think of demanding compensation for reducing the publication price of multiplication tables...

Often, VUZ's acquire computers through income from contract work or with the help of personnel recruiters of enterprises and branches of industry when they, in hopes of hiring specialists of a higher class, allot the necessary hardware. However, even if the money is available, there usually is nothing to buy with it: computer technology is distributed in advance, almost piecemeal. And that which is actually received frequently does not correspond to our needs.

The fact is that computers are supplied both to VUZ's and to industrial enterprises with the same configurations. Meanwhile, the specific character of educational institutions is such that studies are conducted at the same time for many students. Therefore, VUZ computers should have large memory and the maximum possible number of displays and graphic devices.

In addition, computer technology designated for use in VUZ's requires designs that permit its inclusion in computer networks. A universal system, possessing "collective" memory, is able to constantly add the information that it accumulates, drawing it out of the general mass of works. It is understandable that a student in this case passes more quickly from passive mastery of skills in using computer equipment to creative activity based on the stored knowledge of the whole computer network. Uninspired study is replaced by the rush of accelerated though, giving birth to a new type of specialist, who is accustomed to possessing and making the best use of the latest information from related fields of science and technology.

Finally, for VUZ's that still do not have their own experience in computer training, it is useful to develop several standard variants of academic classes for computers of different computer power.

Computer software plays a large role in the effective use of computers. The demand is growing in the country not only for literate users but also for highly qualified programmer-specialists capable of creating quality software. There should be a concentration of their training in a few VUZ's that are more prepared for this and have been supplied with the necessary technology.

Thus, the circle of problems that must be solved is rather broad, beginning with allotment of funds for computer technology for higher schools and ending with specialized programmer training. It is evident, especially now at the threshold of the new five-year plan, that USSR Gosplan together with USSR Ministry of Higher and Secondary Specialized Education, must take cardinal measures for the solution of these problems.

9645
CSO: 1863/274

NEW ACADEMY ORGANIZATIONS PROMOTE COMPUTER TRAINING, R&D

Moscow MOSKOVSKAYA PRAVDA in Russian 21 Apr 85 p 3

MULADZHANOV, Sh. (interviewer)

[Abstract] The article is an interview given on the occasion of Soviet Science Day by academician K. A. Valiyev, deputy director of the USSR Academy of Sciences' Institute of General Physics. Valiyev also holds the positions of director-organizer of the academy's Institute of Microelectronics, which was created recently, and deputy academician-secretary of the academy's new Department of Information Science, Computer Technology and Automation.

Valiyev hails the country's rapid progress in microelectronics and information science, and he mentions important directions of work at organizations with which he is affiliated. Personnel of the Institute of General Physics are giving support to the automotive industry and medicine in the introduction of laser and optical-fiber technology, for example. Valiyev also responds to questions regarding cooperation among scientists, educators and industrial specialists in expanding training programs and promoting technological progress in computer technology and microelectronics. He reports that the academy's information-science department and institutes and laboratories are taking part directly in preparing curricula for higher schools and improving their technical resources.

FTD/SNAP
CSO: 1863/289

PROJECTS OF THE COMPUTER CENTER OF THE ACADEMY'S SIBERIAN BRANCH

Moscow PRAVDA in Russian 8 April 85 p 7

SOLOMENKO, Ye., correspondent (Novosibirsk)

[Abstract] The lengthy article reports on activities of the Computer Center of the Siberian Branch of the USSR Academy of Sciences. The center's director is academician A. Alekseyev. The center is located in a five-story building in Novosibirsk.

Examples of the center's work in support of various industries and services are reported. Alekseyev directed a cycle of work on physical-technical principles of vibration sounding of the Earth's interior, for gathering geological information. The laboratory of automated systems for gathering information, which is headed by B. Glinskiy, built field computer complexes for geological parties to use in this method of geological surveying. S. Vas'kov, head of the department of applied geophysics, related that his group is now working on a satellite hook-up for transmitting the geological information from the field to the computer center. It is noted that the center is the base of a recently opened center for processing of geo-information, which analyzes aerospace photographs not only for geological purposes, but also in support of agriculture and forestry.

In addition to applications of computers, the center reportedly is also doing research on computer technology and computer systems of the future. It has designed a number of computer networks, one of which provides collective use of all the computers of facilities of the Siberian Branch located in Novosibirsk. This network is to be extended to all institutes of the branch throughout Siberia. One of the center's most important long-term projects is the "MARS", which is an acronym for 'modular' asynchronous expandable system.' It aims at flexible computer complexes which will be adaptable to specific conditions of a given research institute or design bureau. V. Kotov, the center's deputy director, explained that such complexes require the development of super-computers that will mark the advent of the fifth computer generation. Collaborating with the center on this project are the Computer Center of the USSR Academy of Sciences in Moscow, the

Estonian Academy of Sciences' Cybernetics Institute, and industry design bureaus and enterprises. It is said that "MARS" machines will be capable of recognizing visual images and accumulating information independently, and of working with imprecise data. They will be capable of working out their own programs for the solution of problems posed for them.

FTD/SNAP
CSO: 1863/329

- END -