Non-profit joint-stock company

ALMATY UNIVERSITY OF POWER ENGINEERING AND TELECOMMUNICATION

IT-engineering Department

DIGITAL CIRCUIT DESIGN
Part 1

Instructional Guidelines for implementing of laboratory works
for students majoring in
5B070400 – Computer science and software

Almaty 2017

COMPILERS: Shanayev O.T., Musapirova G.D., Utegulov A.N. Digital circuits. Part 1. Instructional guidelines for the implementing of laboratory works for students majoring in 5B070400 – Computer science and software. – Almaty: 2017. – 37 p.

This Manual presents a series of laboratory assignments focused on studying functional characteristics of basic components, of digital systems, namely specifically, logical elements, combinational and sequential devices, and mastering the tools required for performing the research. laboratory assignments are organized in the form of virtual experiments carried out by simulating the devices under studing these the Electronics Workbench integrated environment.

Two final assignments in the manual are devoted to the study of the internal structure and principles of operation, of the individual building blocks, and a microprocessor system. Figures 39, tables 6, references – 6 titles.

Reviewer: Assistant of the Department "Safety Labour and Environmental Engineering" Duisenbek J. S.

Published according to the supplementary plan of non-profit joint-stock company “Almaty University of Power Engineering and Telecommunication” for the year 2017

Master plan 2017, pos. 17

Oryngali Tulegenovich Shanaev,
Gulzada Dauletbekovna Musapirova,
Aidos Nurzhanovich Utegulov

DIGITAL CIRCUIT DESIGN
Part 1

Instructional Guidelines for implementing of laboratory works
for students majoring in
5B070400 – Computer science and software

Editor U.B. Serikbaeva
Standardization specialist N.K. Moldabekova

Signed in print
Circulation 30 exem.
Volume 2,25. ed.edition.

Format 60x84/16
Typographical paper № 1
Order Price 1125 tg.

Copy and duplicating bureau of the
noncommercial joint-stock company
"Almaty University of Power Engineering and Telecommunication "

050013, Almaty, st. Baytursynova, 126

Introduction

Our course, "Digital circuit design" is devoted to the study the functional foundations digital system equipment and circuit design principles of their development. The course is required in order to ensure fascilitation of students' theoretical knowledge of the subject and their acquisition of practical skills in the development of digital systems of various modes of operation.

There are two factors to be considered during the development of microprocessor systems: design of the system hardware that provides physical implementation of the solution of the stated problem, and definition of the instruction set (software), that allows drawing up a program of actions and interactions of system’s functional components to solve the problem. Consequently, in order to obtain complete knowledge of the microprocessor system and mastering of both its hardware and software parts is required. Final assignments in the lab manual are devoted to the study of the internal structure and principles of operation of the microprocessor system and its individual building blocks (input / output devices, memory and microprocessor devices). Implementing of laboratory assignments is organized in the form of virtual experiments carried out by modeling the structure of the devices under investigation and their operation in the Electronics Workbench integrated environment.

1 Laboratory work №1. Logical elements

Learning Objectives:

– study functional characteristics of the logical elements;
– observing signal delay in logical elements and measuring its duration;
– mastering the application of logic analyzer for investigating digital device;
– mastering the principles of separate operational unit creation.

1.1 Equipment

Computer, Electronics Workbench development environment.

1.2 Practical information and methodical guidelines

The simplest method of investigating the nature of a logical element (or a more complicated digital device) is to determine the output level for different combinations of logic levels at its inputs. Hereafter, this method of investigating the device operation will be called as the key of operation (or switching mode).

More visual and faster results can be obtained by applying a sequence of rectangular pulses of different frequencies to the inputs of the element (or device) and comparing the observed time-base diagrams of its input and output signals using a logic analyzer. Hereafter, this method of investigating the device operation will be called to as the pulse mode of operation.

It is necessary to create a single-pulse generation subcircuit (block) operated by keystrokes, as an introduction to creating individual functional units is this assignment. The main advantage of these subcircuits is the possibility of representing volume circuits in the form of several functional blocks of a sturctured circuit, which undoubtedly contributes their clarity. Another advantage of using sub-circuits is that they can be reused multiple times in other experiments by transferring them from one file to another.

One of the widely used of elements for constructing digital circuits is the tri-state element, that the third state is the high-impedance (ie disabled) state. This assignment demonstrates one of the simplest options of using such kind of devices.

1.3 Activities

1.3.1 Study the operation of logical elements:

– assemble the circuit for investigating the behavior of NOT gate in different investigation modes (figure 1.1);

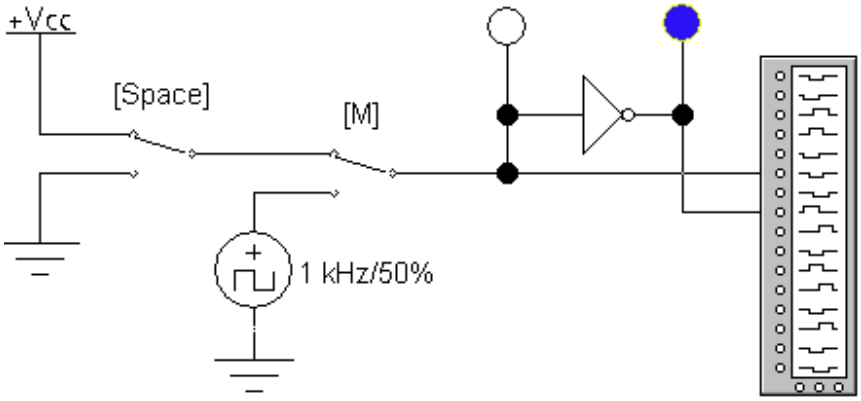

Figure 1.1

– set the M switch to upward position (key mode) and verify the operation of the element under investigation with the help of indicators. Write down the results into a table;

– set the M switch to downward position (pulse mode) and verify the operation of the element under investigation with the help of the logic analyzer. Copy the results into a Bitmap (Copy as Bitmap) and include them in the report.

1.3.2 study the work of two-input logical elements:

– Assemble the circuit for investigating the behavior of AND and NAND gates (figure 1.2) and study their operation. Copy the results into a Bitmap and include them in the report.

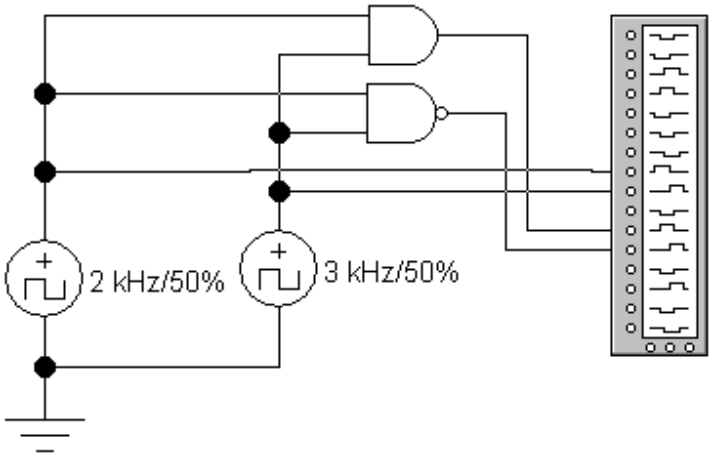

Figure 1.2

– transform the circuit for investigating the behavior of OR and NOR gates and verify their operation;

– transform the circuit for investigating the behavior of XOR and XNOR gates and verify their operation.

1.3.3 Observing the signal to a logical element with an oscilloscope (figure 1.3), measure the length of that delay.

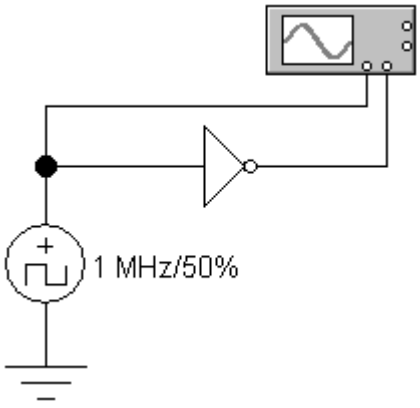

Figure 1.3

1.3.4 Assemble the single-pulse generator circuits based on XOR и XNOR elements, observe their operation and form them as separate blocks under the names ‘j’ and ‘t’, relatively (figure 1.4).

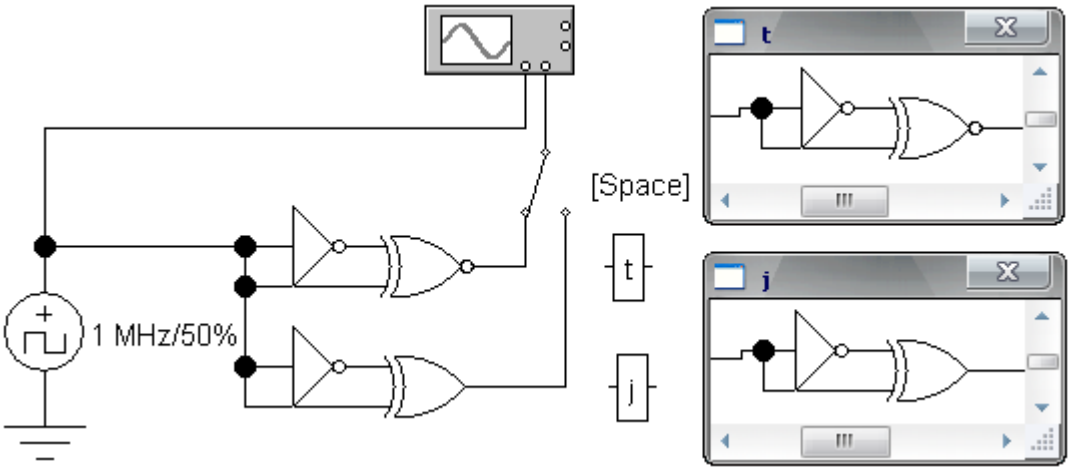

Figure 1.4

1.3.5 Assemble the single-pulse generator circuits based on other two-input elements (AND, NAND, OR, NOR) , and observe their operation. Pay attention to the nature and time interval generation of a single pulse on respective circuits.

1.3.6 Assemble the circuit for investigating tri-state buffers and their implementation (figure 1.5). By changing the state of the switch, and to be sure that one of the generated pulses corresponding to the identity of the control input of one of the elements is translated to the output of the circuit.

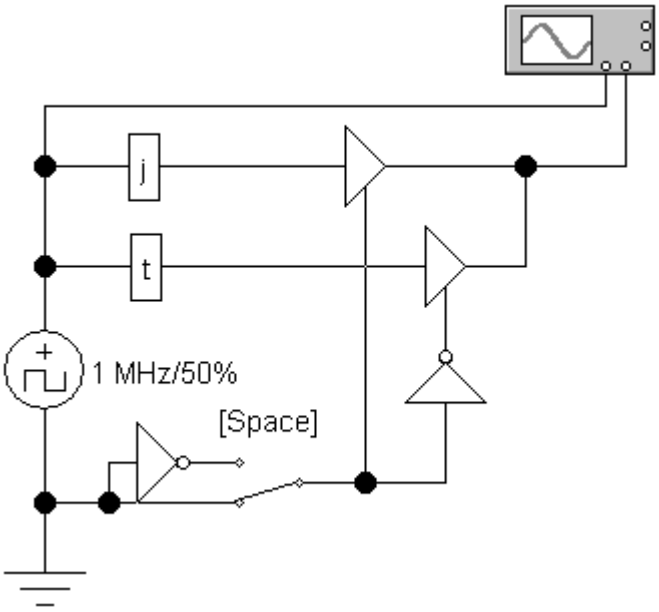

Figure 1.5

1.4 Review questions

1. Describe the logic behind operation of the studied elements.
2. How can you increase the number of inputs for AND and OR elements in practice?
3. How can you increase the number of inputs for NAND and NOR elements in practice?
4. Explain the possibility of implementing the functions NAND and NOR elements with the help of basic elements.
5. Explain the difference between single pulses generated on the basis of XOR and XNOR.
6. Explain the nature of single pulses generated on the basis of AND and NAND elements.
7. Explain the nature of single pulses generated on the basis of OR and NOR elements.
8. Explain the process of applying the logic analyzer for studying digital device operation.

2 Laboratory work №2. Combinational devices

Learning Objectives:

– mastering the techniques for assembling combinational circuits based on logical elements;

– mastering the application of logic analyzer for creating a combinational device circuit.

2.1 Equipment

Computer, Electronics Workbench environment.

2.2 Practical information and methodical recommendations

Creation of a combination device circuit based on logical elements, that implements a function defined by a truth table, includes following stages

– obtain a logical expression corresponding to the tabular representation of a given function;

– minimize the obtained logical expression, or convert it to a form intended for the implementation of the circuit on the basis of certain base elements (for example: NAND gates);

– assemble the device circuit according to the obtained expression

Having connected additional elements (titled keys and indicators respectively), required for conducting the investigation, to the resulting circuit it is possible to verify the conformity of its operation with the tabular representation of a given function.

The Electronics Workbench software includes a device called Logic Converter, with which you can implement all stages of synthesis of a combinational device circuit. Application of this specific device for creating the circuit of a combinational device is suggested in the Activities section of the laboratory assignment.

As an illustrative example, figure 2.1 shows the implementation of the circuit according to option F0 (table 2.1), carried out with the help of a logic converter.

Table 2.1

A	B	C	F0	F1	F2	F3	F4
0	0	0	1	1	0	1	0
0	0	1	0	1	1	0	1
0	1	0	0	0	1	1	0
0	1	1	1	1	0	1	1
1	0	0	1	0	1	0	1
1	0	1	0	1	1	1	0
1	1	0	1	1	0	0	1
1	1	1	1	0	1	1	1

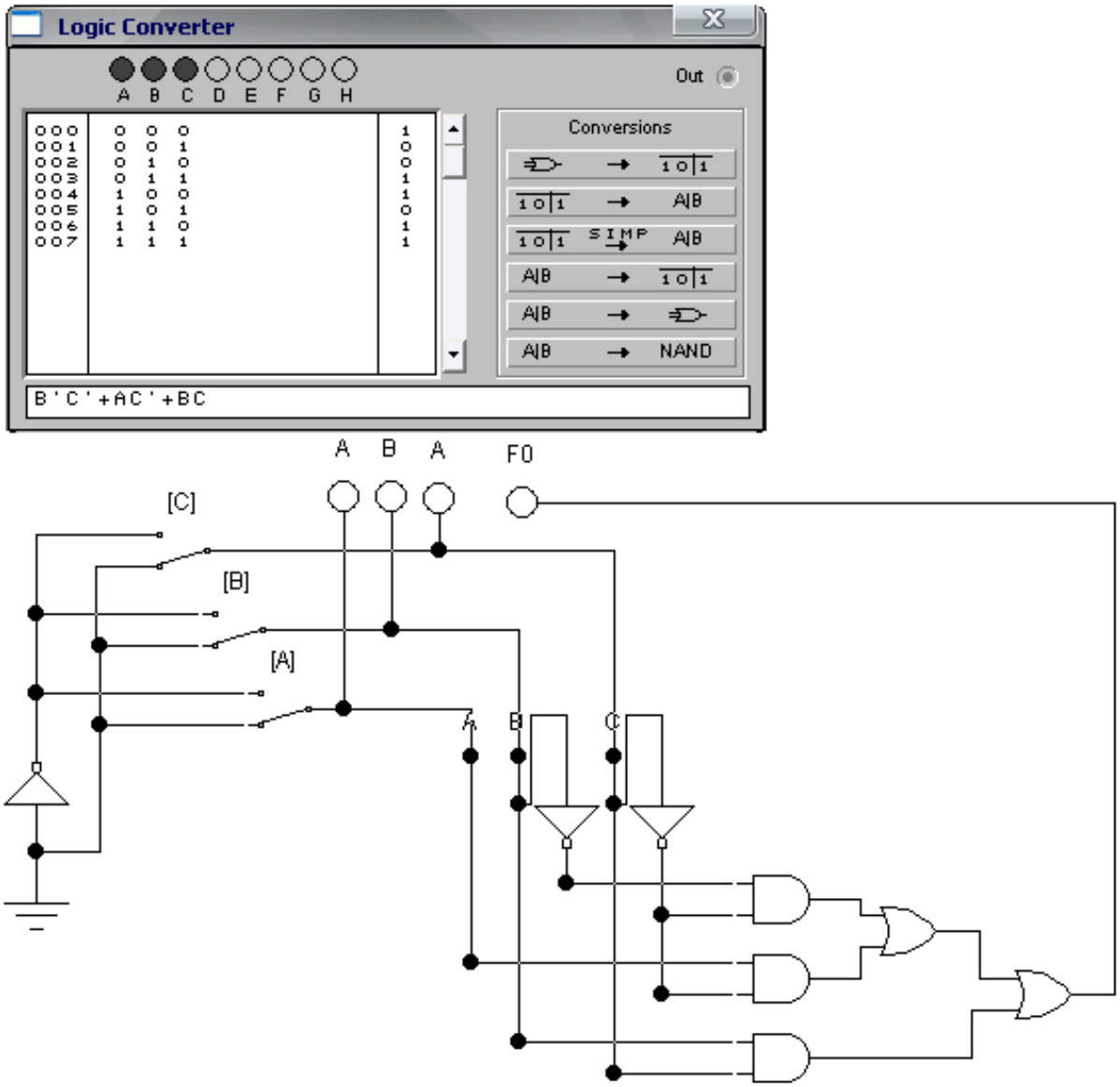

Figure 2.1

2.3 Activities

2.3.1 Assemble the circuit of a combinational device, performing the function defined in the table according to options assigned by the instructor applying the method described in section 2.2 (table 2.1):

- on the basis of AND, OR, NOT elements;
- on the basis of NAND elements;
- on the basis of NOR elements.

2.3.2 Synthesize the circuits of suggested combinational devices with help of Logic Converter:

- on the basis of AND, OR, NOT elements;
- on the basis of NAND elements.

2.3.3 Connect the elements required for conducting an investigation (figure 2.1) to the resulting circuits. Verify their correct correct operation by applying different combinations of logic levels on the inputs of the circuits, and by observing the levels of output signals.

2.4 Review questions

1. Explain the feasibility of building NAND and NOR element functions with the help of basic elements (AND, OR, NOT).
2. Explain the feasibility of building basic element functions (AND, OR, NOT) with the help of NAND or NOR elements.
3. Explain the procedure for the synthesis of circuits of combinational devices.
4. Explain how to obtain a minimized logical expression by direct transformation.
5. Explain the procedure for obtaining the minimized logical expressions using Karnaugh maps.
5. In what way do we need to transform a disjunctive logical expression for constructing a circuit on the basis of NAND elements?
6. How can we obtain an expression for implementing a circuit of a combinational device on the basis of NOR elements?
8. Explain application of Logic converter for creating circuits of combinational devices.

3 Laboratory work №3. Basic combinational devices

Learning objectives: learning the behaviors of encoder, decoder, multiplexer, demultiplexer and adders.

3.1 Equipment

Computer, Electronics Workbench environment.

3.2 Activities

3.2.1 Investigate the operation of 74147 encoder shown on the circuit of figure 3.1.
3.2. Investigate the operation of 74138 decoder shown on the circuit of figure 3.2. Herein, the conditional counter (count) is used to supply the address code of the decoder. A single pulse generator in the circuit can be used by pressing the (j), which was set up in the assignment no 1.

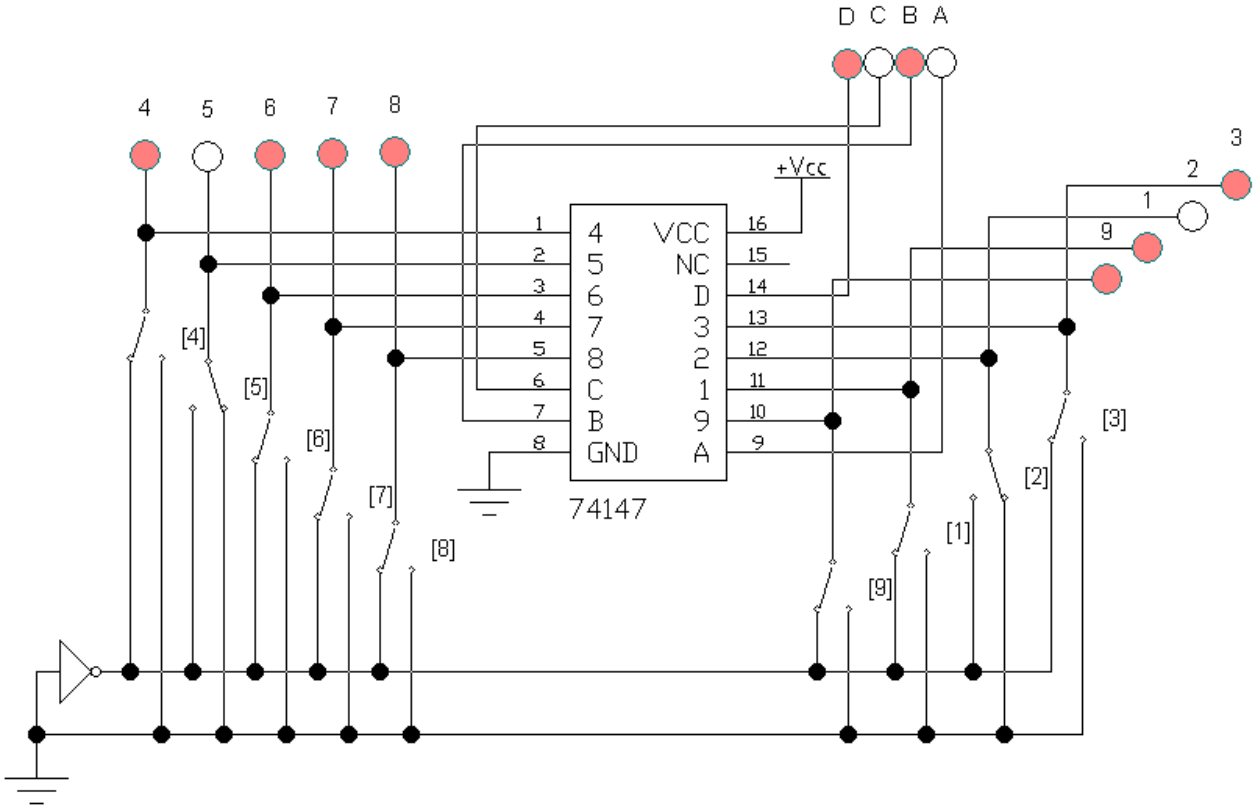

Figure 3.1

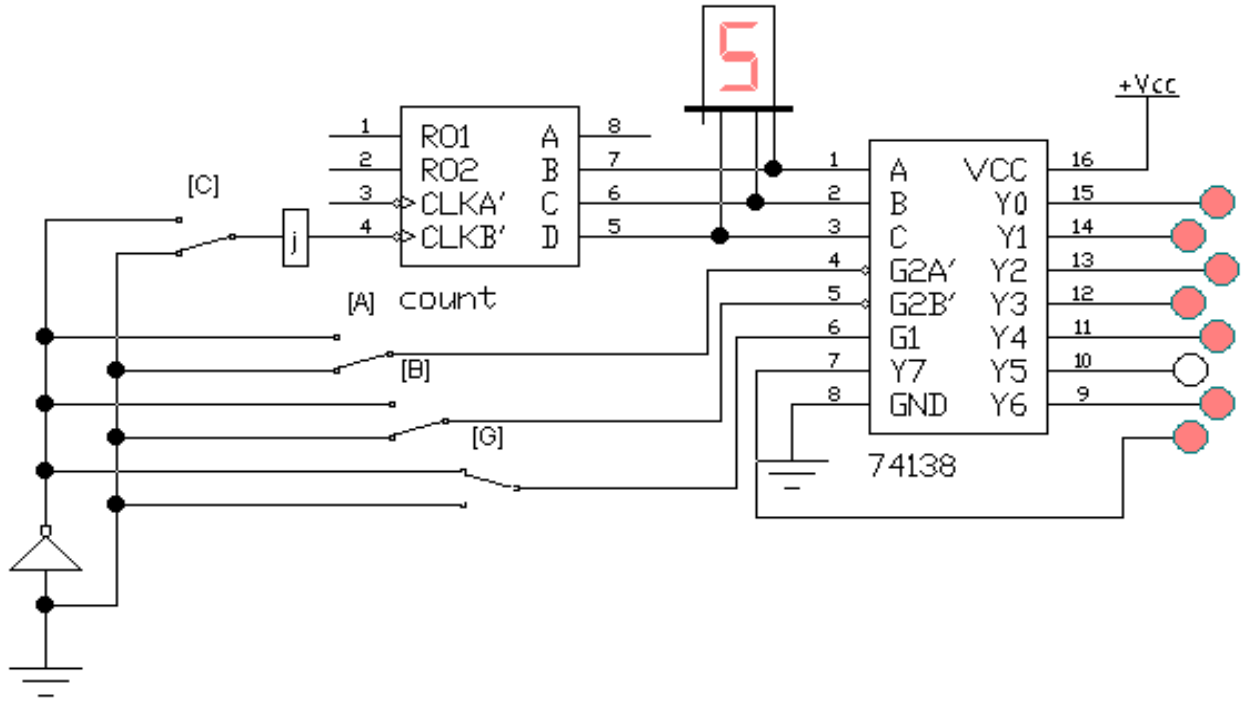

Figure 3.2

3.2.3 . Investigate the operation of 74153multiplexer with the help of logic analyzer (figure 3.3). Here, counter 1 is used to set the information signals, and counter 2 – for address codes and the enable signal

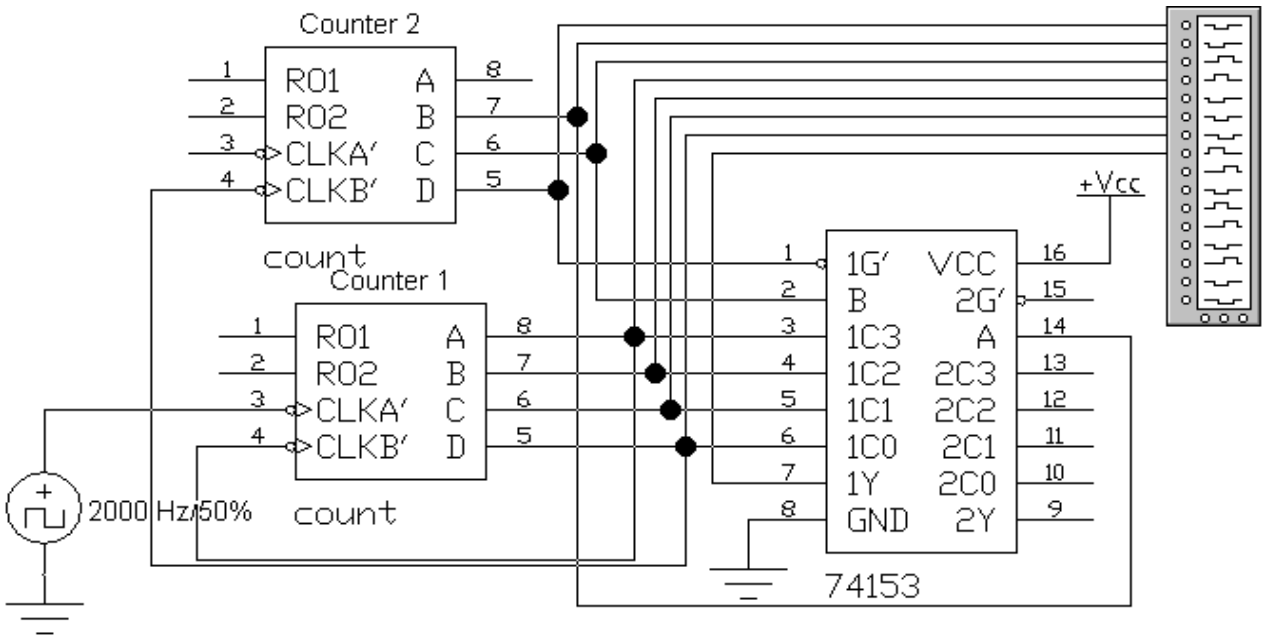

Figure 3.3

3.2.4 Investigate the operation of demultiplexer with the help of logic analyzer by building a an ivestigation circuit, similar to the one used to investigate the operation of multiplexer.

3.2.5 Build the half-adder and single-bit full adder circuits with the help of logical elements and verify their operation.

3.2.6 Register the single-bit full adder circuit as a sub-circuit then create a circuit for a 4-bit adder and verify its operation.

3.3 Review questions

1. Explain encoder's principles of operation
2. Explain decoder's principles of operation
3. Create a decoder circuit on the base of logical elements with characteristics provided by the instructor.
4. Explain multiplexer's principles of operation
5. Explain demultiplexer's principles of operation
6. Create a decoder circuit on the base of logical elements and a decoder.
7. Explain the half-adder and single-bit full-adder circuits.

4 Laboratory work №4. Universal logical modules

Learning Objectives**:**

– mastering the ways of implementing a decoder for performing logical functions;

– mastering the methods of tuning a universal logical module on the basis of a multiplexer.

4.1 Equipment

Computer, Electronics Workbench environment.

4.2 Practical information and methodical recommendations

A logical function can be implemented using a decoder. Let us demonstrate this possibility using an example from laboratory assignment no. 2. Analyzing the logical expression $F = \overline{A}\overline{B}\overline{C}V\overline{A}BCV\overline{A}B\overline{C}VAB\overline{C}VABC$, that describes the behavior of the device, we can see that conjunctions in the expression, match the activated outputs of the decoder. In order to implement this function on the basis of a decoder with inverted outputs, it is necessary to transform the logical expression using DeMorgan's Laws. The expression obtained - $F = \overline{\overline{A}\overline{B}\overline{C}V}\,\overline{\overline{A}BCV}\,\overline{\overline{A}B\overline{C}V}\,\overline{\overline{A}BCV}\,\overline{\overline{ABC}}$ shows that in this case we need to use a NAND element (figure 4.1).

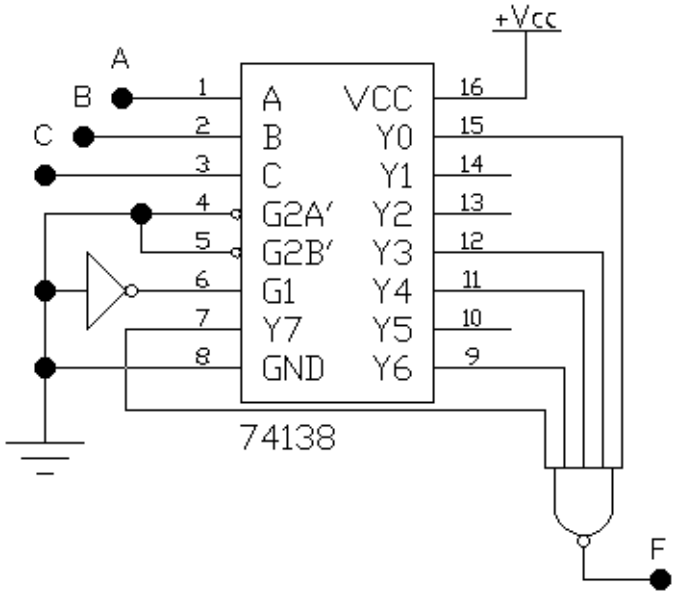

Figure 4.1

On the basis of a multiplexer we can create a universal logical module (ULM) that can be used for implementing significantly diverse functions. We can illustrate various methods of creating the ULM for implementating examples of functions of two, three and four variables on the basis of a multiplexer 4-1 (tables 4.1-4.3 and figures 4.2-4.4).

Table 4.1

A	B	F1
0	0	1
0	1	1
1	0	0
1	1	1

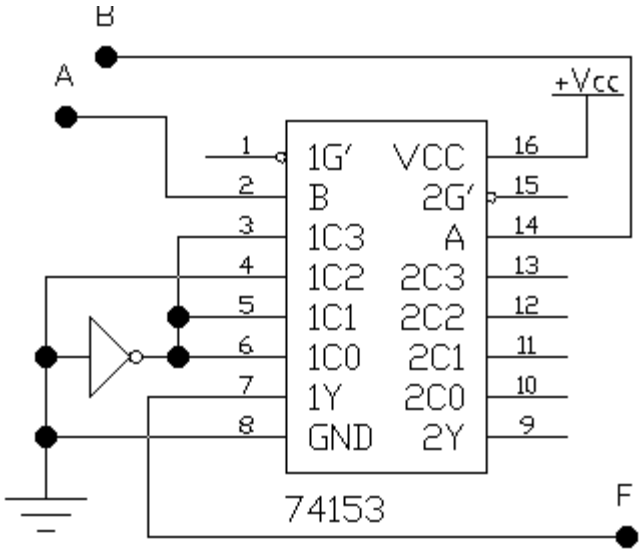

Figure 4.2

Table 4.2

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1		1
1	1	1	0

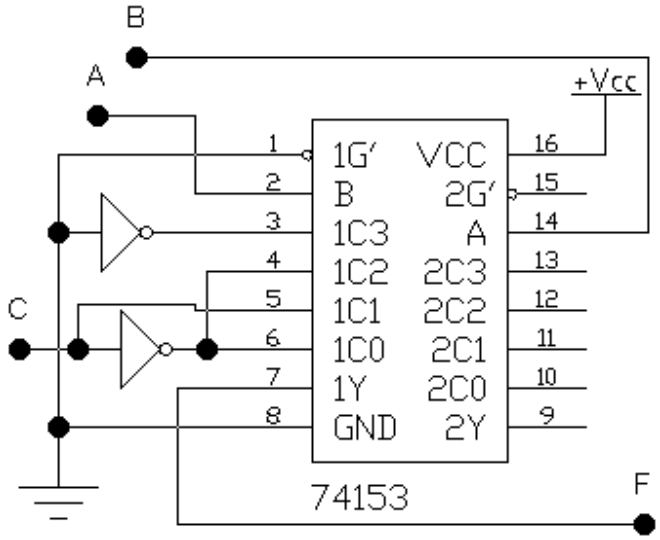

Figure 4.3

Table 4.3

A	B	C	D	F	A	B	C	D	F
0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	0	0	1	1
0	0	1	0	0	1	0	1	0	0
0	0	1	1	1	1	0	1	1	0
0	1	0	0	1	1	1	0	0	0
0	1	0	1	1	1	1	0	1	0
0	1	1	0	1	1	1	1	0	0
0	1	1	1	1	1	1	1	1	0

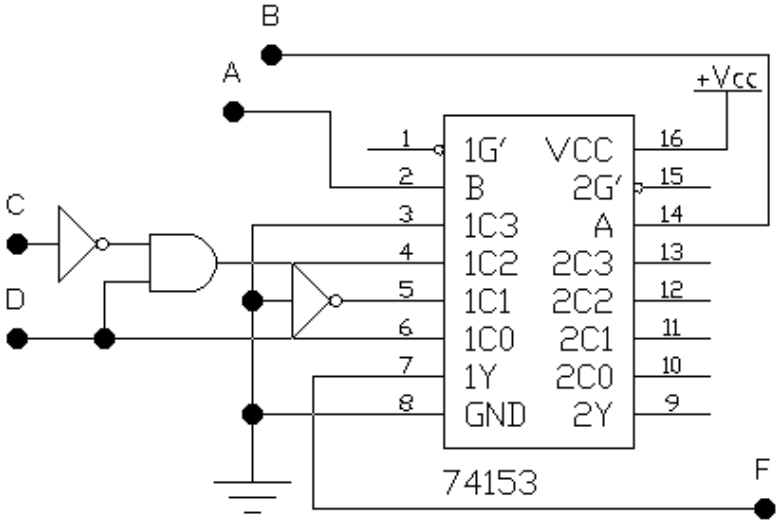

Figure 4.4

4.3 Activities

4.3.1 Create the circuit of one of the devices provided in the table on the basis of 74138 decoder. Analyze the behavior of the circuit obtained, adding the required elements for conducting the investigation.

4.3.2 Create the circuit of one of the devices provided in the table on the basis of 74153 multiplexer. Analyze the behavior of the circuit obtained, adding the required elements for conducting the investigation.

4.4 Review questions

1. Explain decoder's principles of operation.
2. How can we obtain a circuit that performs the function of a decoder with 8 outputs with the help of a decoder with 4 outputs (microchip 74138)?
3. Explain multiplexer's principles of operation.
4. How can we obtain a circuit that implements the function of a multiplexer with 8 outputs with the help of a multiplexer with 4 outputs (microchip 74153)?
5. How can we get the scheme that implements the combination function with the help of a decoder with direct outputs?
6. How can we obtain a circuit that implements a combinational function with help of a decoder with inverse outputs?
7. Explain the method of tuning the ULM when the number of variables of the function being implemented is equal to the number of address inputs of the multiplexer
8. Explain the method of tuning the ULM when the number of variables of the function being implemented is greater than the number of address inputs of the multiplexer

5 Laboratory work №5. Memory Elements

Learning Objectives**:**

– mastering the principles of construction and the operation modes of edge-triggered flip-flops;

– mastering the principles of construction and the operation modes of edge-triggered flip-flops with static control;

– studying modes of operation for the edge-triggered flip-flops with dynamic control.

5.1 Equipment

Computer, Electronic Workbench environment.

5.2 Activities

5.2.1 Assemble the circuits of edge-triggered RS flip-flops on the baisis of NOR and NAND elements (figure 5.1) and study their modes of operation by measuring the levels of output signals.

In this and the following activities the source of logic levels is assembled into a unit titled "0_1".

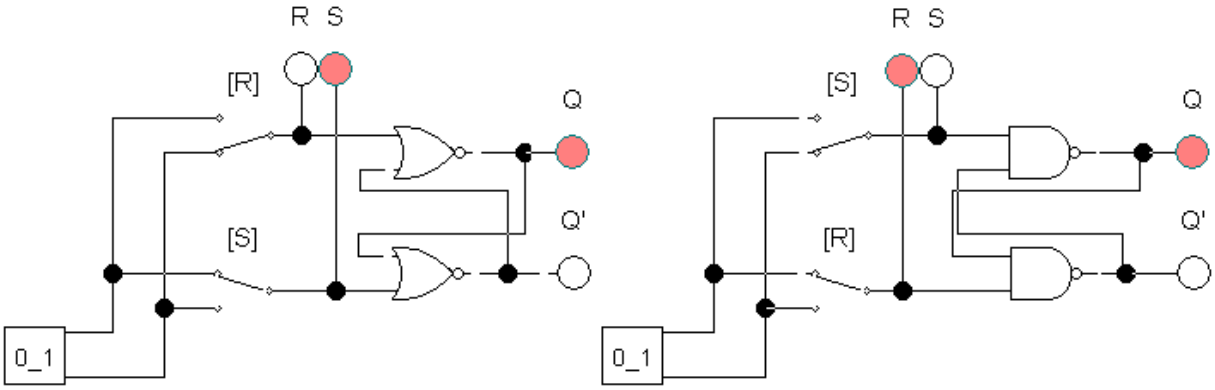

Figure 5.1

5.2.2 Assemble the circuits of edge-triggered RS flip-flops on the baisis of NOR and NAND elements (figure 5.1) and study its modes of operation.

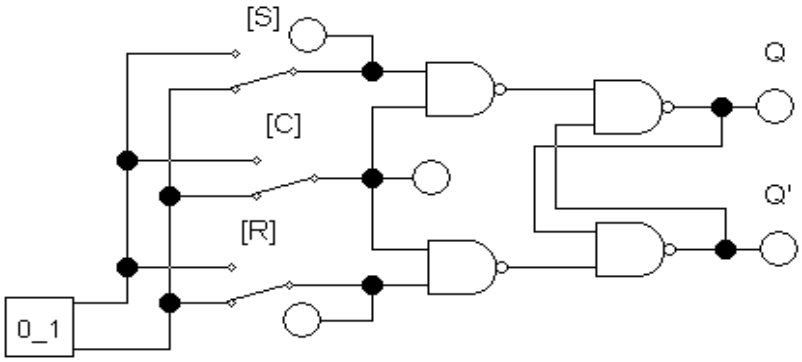

Figure 5.2

5.2.3 Investigate modes of operation of one of D flip-flops in the 7474 microchip (figure 5.3) with the help of logic analyzer.

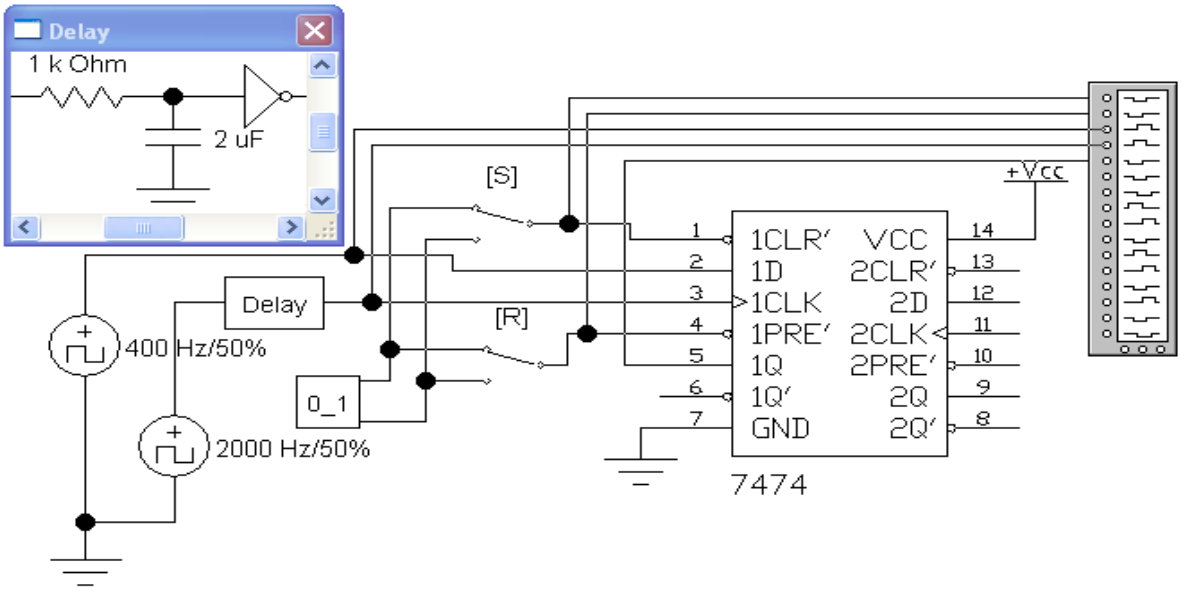

Figure 5.3

At the beginning of the analysis the R and S switches are to be set in “set” state. During the analysis by rapidly double-clicking on one of the buttons corresponding to these switches, implement asynchronous setting or clearing of the flip-flop.

5.2.4 Analyze the operation of a D-trigger in counter mode, for which you need to connect its inverse output to its information input (figure 5.4).

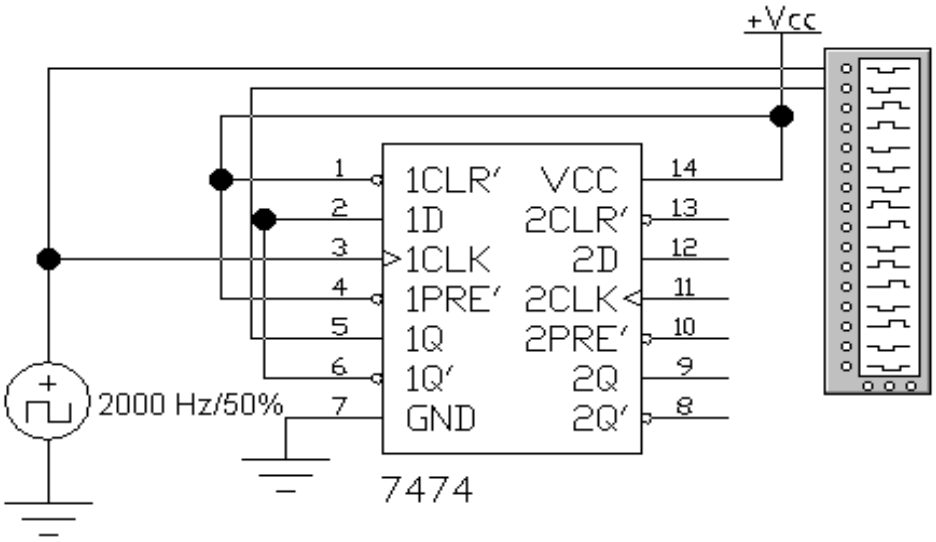

Figure 5.4

5.2.5 Investigate the modes of operation of a JK flip-flop (figure 5.5).

Installation and resetting up of the flip-flop are to be performed with a short change of signal levels on the R and S inputs during the analysis.

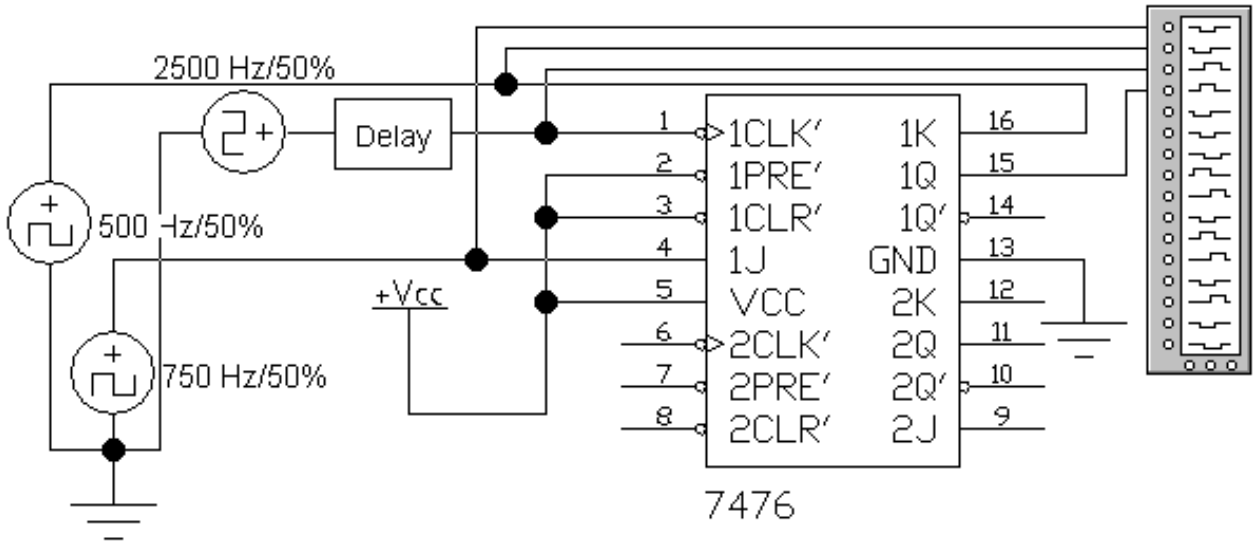

Figure 5.5

5.3 Review questions

1. Explain the structure and modes of operation of edge-triggered RS flip-flops with direct inputs.
2. Explain the structure and modes of operation of edge-triggered RS flip-flops with inverse inputs.
3. Explain the structure and modes of operation of edge-triggered RS flip-flops with static control.
4. What is the difference between static and dynamic controls of flip-flops?
5. How does one use a D flip-flop as a counter?
6. Explain the modes of operation of edge-triggered JK flip-flops.
7. How does one use a JK flip-flop as a counter?
8. What functions do signals on R and S inputs of D- and JK- flip-flops perform?

6 Laboratory work №6. Sequential devices

Learning Objectives**:**

– mastering the principles of operation and structural differences summing up and substracting counters;

– mastering the structural principles of altering the counting modulo of a counter;

– mastering different modes of entering information into the registers.

6.1 Equipment

– computer, electronics workbench environment.

6.2 Activities

6.2.1 Create units (sub-circuits) of a dynamic data source and single pulse generatoros (j and t) with corresponding polarity to use them in further experiments. (figure 6.1).

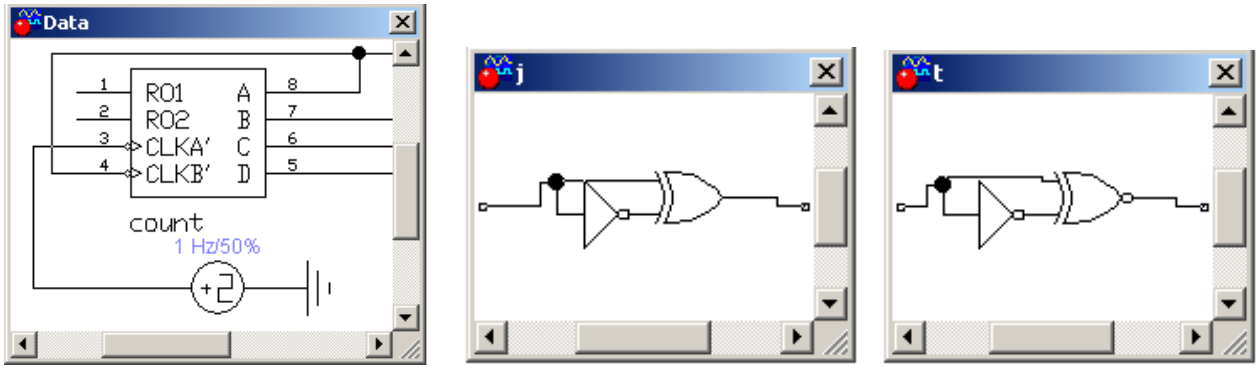

Figure 6.1

6.2.2 Assemble the circuit for investigating the 7493 counter (figure 6.2). Verify the operation of the counter by supplying clock pulse signal with the help of 'C' key on the keyboard. Verify signal assignment on R01 and R02 pins of the flip-flop's microchip.

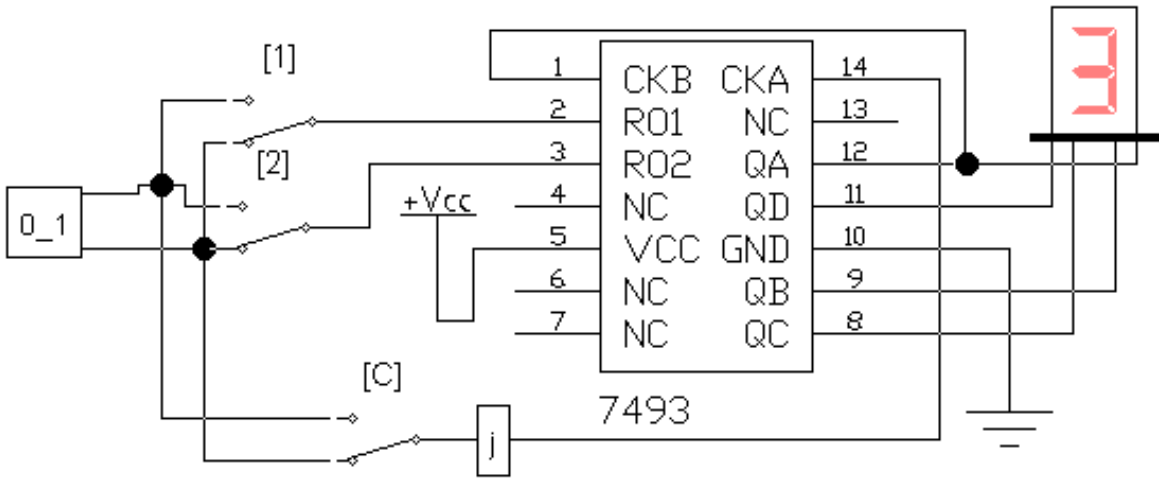

Figure 6.2

6.2.3 Verify operation of the counter with a counting modulo of 10 (figure 6.3).

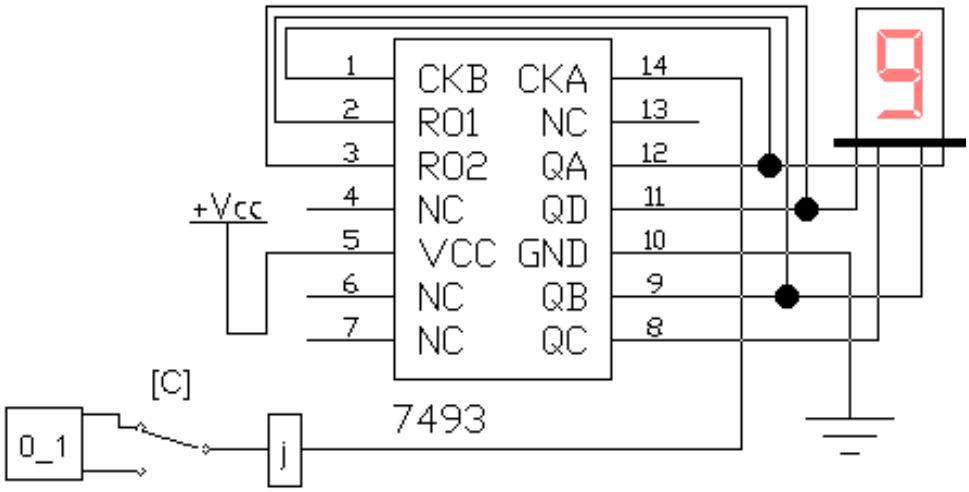

Figure 6.3

6.2.4 Alter the counting modulo of the counter and verify its operation.

6.2.5 Assemble the circuit for investigating the reverse counter 74169 (figure 6.4) and analyze its modes of operation.

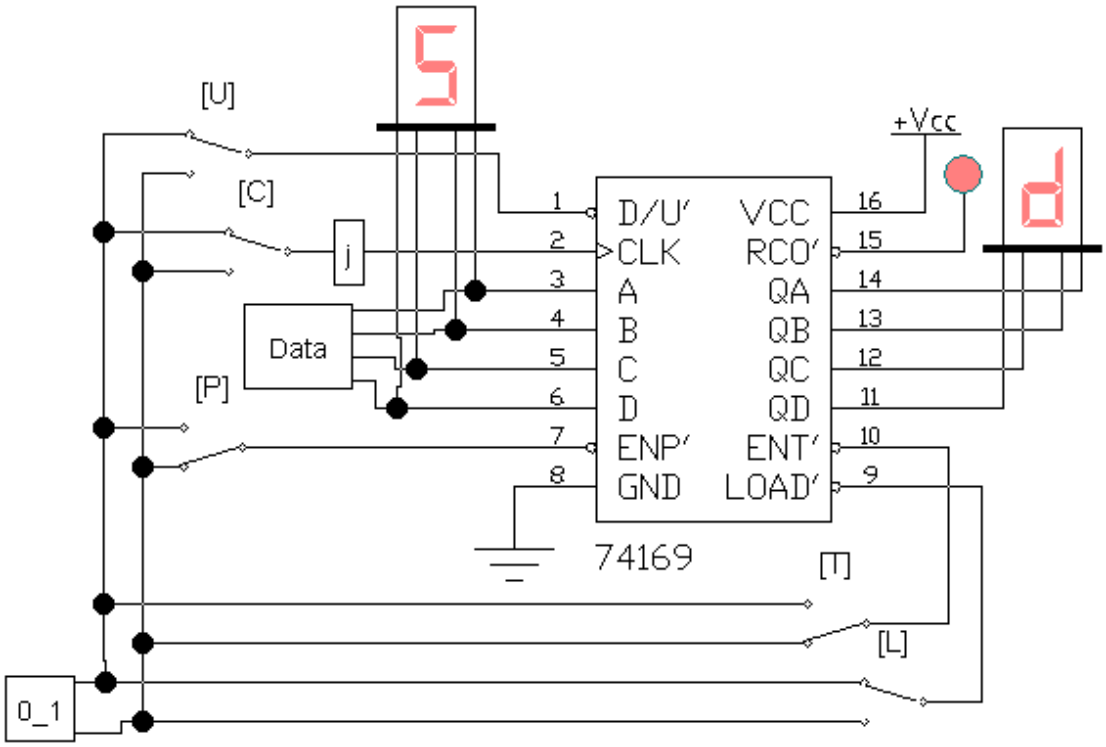

Figure 6.4

6.2.6 Assemble the circuit for investigating the reverse counter 74194 (figure 6.5) and analyze its modes of operation.

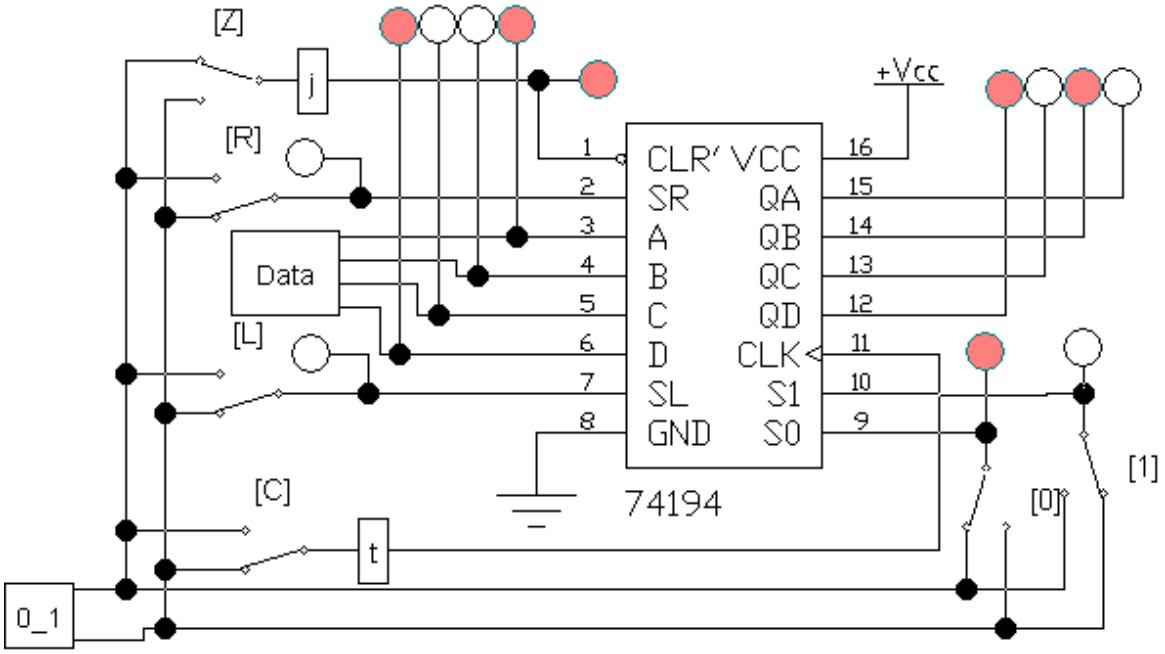

Figure 6.5

6.3 Review questions

1. Explain the principles of sequential and parallel inputs of information to the register.
2. How can we obtain a counter with counting modulo of 13 on the basis of 7493 microchip?
3. How can we obtain the 8-bit counter on the basis of 7493 microchip?
4. Create the circuit of a counter with a calculating modulo of 175 on the basis of 7493 microchip
5. Create the circuit of a counter with a calculating modulo of 100 on the basis pf 7493 microchip
6. Explain the modes of operation of a 74169 counter.
7. What function does the 74169 counter's RCO' output perform?
8. Explain the modes of operation of a 74194 register.

7 Laboratory work №7. Memory devices

Learning Objectives:

- studing the principles of building ROM;
- acquiring the skills of programming PROM;
- studing basic structures of Memory Storage Devices and principles of their construction.

7.1 Equipment and files

Computer, Electronics Workbench Professional.

File ROM(M) 16x7.ewb.
File PROM 16x7.ewb.
File RAM 2D_8x4.ewb.
File RAM_3D_16x4.ewb.
File RAM 2DM_32x4.ewb.

7.2 Practical information and methodical recommendations

The first assignment of the laboratory work is devoted to the study of the structure and principles of Read-Only Memory (ROM) with inalterable information to display hexadecimal symbols on the 7-segment indicator (figure 7.1).

ROM unit (subcircuit ROM_16x7) consists of a decoder with 16 outputs (74154 chip) and a memory matrix with the 16x7 organization, designed in the form of subcircuits with corresponding titles. The enable signal for the decoder's operation is provided by a negative signal level from the key E (Enable). Addresses of memory lines (7-bit cells) are formed by a 4-bit counter (subcircuit add4).

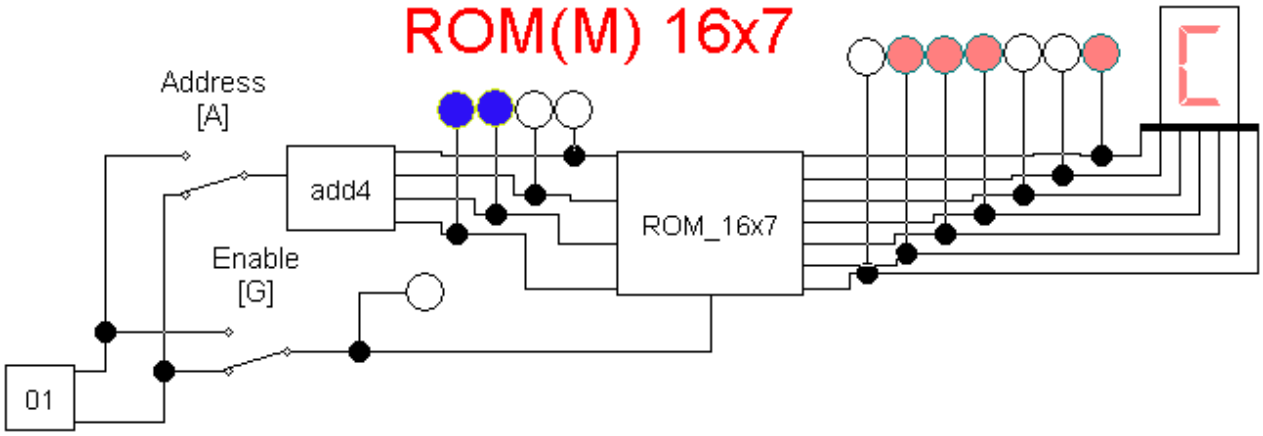

Figure 7.1

The second activity is devoted to the study of the structure (figure 7.2) and programming principles of a microchip (in our case, the subcircuit chip 16x7), which in its initial state has all the switches at the intersections of sixteen horizontal lines (output lines of the decoder) and seven vertical lines (bit lines).

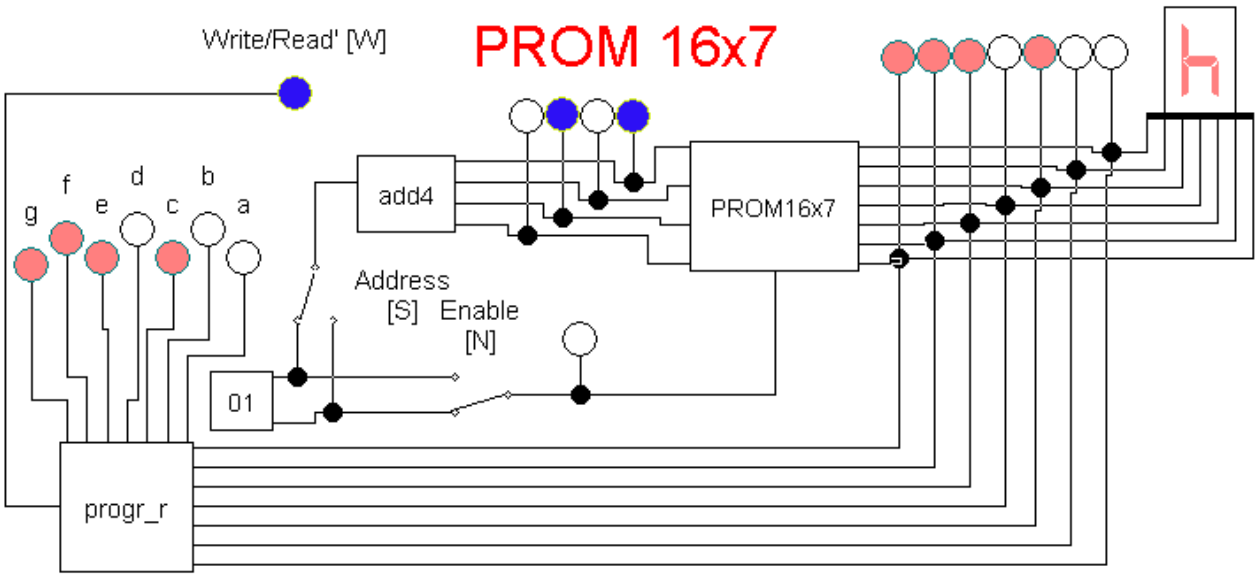

Figure 7.2

Programming of the microchip is performed in the following order:

– activate the circuit by pressing the start button (activate simulation);

– set the device to the programming mode by pressing W (Write/Read) (the corresponding LED illuminates);

– set the address of a programming row in the memory matrix;

– prepare the programming information by pressing the keys on the keyboard, corresponding to the disposable segments of the symbol which is to be set on this line;

– provide programming of the line mode with a short press of the “P” key (information weaving);

– restore the initial states of segment keys;

– set device to the read mode by pressing W (Write/Read) and verify the information programmed to the lines of the memory matrix.

The following activities of the laboratory work devoted to the study of different memory system structures (2D, 3D, 2DM) and their principles operation, are to be run on the respective models (figures 7.3-7.5).

The investigation circuit for the memory system according to 2D structure (figure 7.3) consists of the following units, presented in the form of the respective subcircuits:

– 3-bit address driver (add3) on the basis a counter;

– 4-bit data source on the basis of a counter;

– RAM_8x4 memory system, which consists of a decoder with 8 outputs (dc8) and memory matrix cells (8x4), containing 8 cells (rg4) based on registers. Each of the cells of the memory matrix is provided with an indicator that allows you to monitor selection of the memory cell during experimentation.

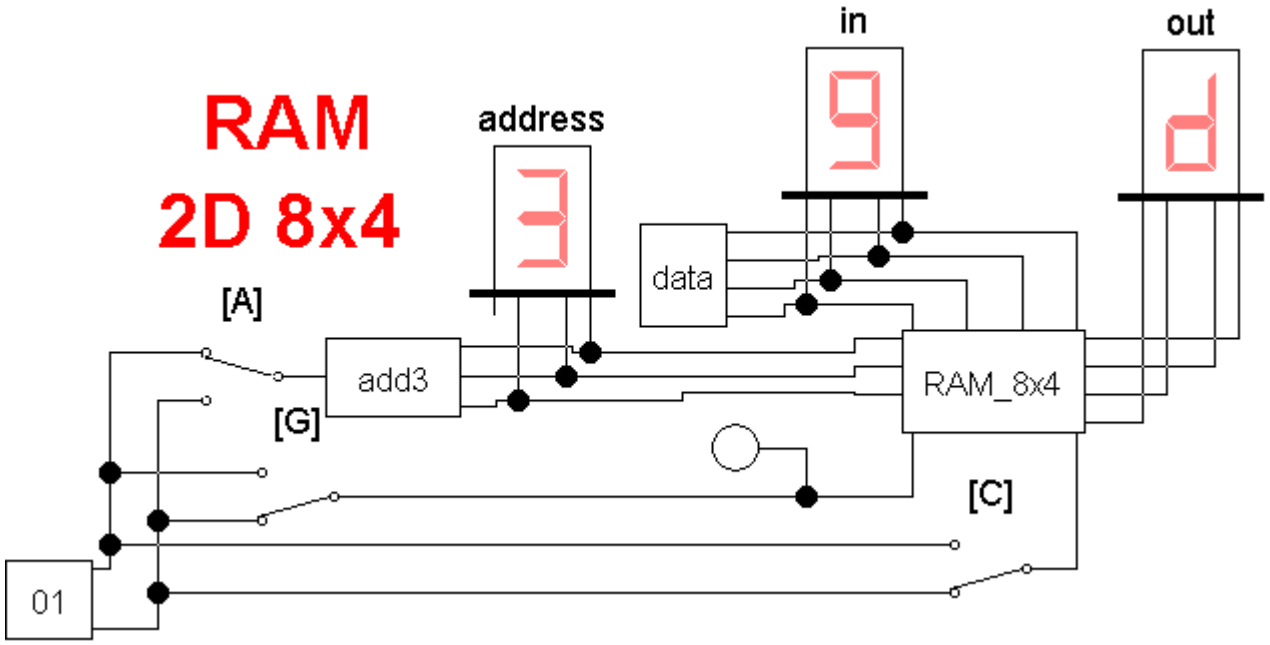

Figure 7.3

The investigation circuit for the memory system on 3D structure (figure 7.4) consists of the following units, presented in the form of the respective subcircuits

- 4-bit address driver (add4) on the basis of a counter;
- 4-bit data source on the basis of a counter;
- RAM_16x4 memory system, which consists of a double decoder (2dc4) and a matrix of memory cells (16x4), containing 16 cells (rg4) on the basis of registers.

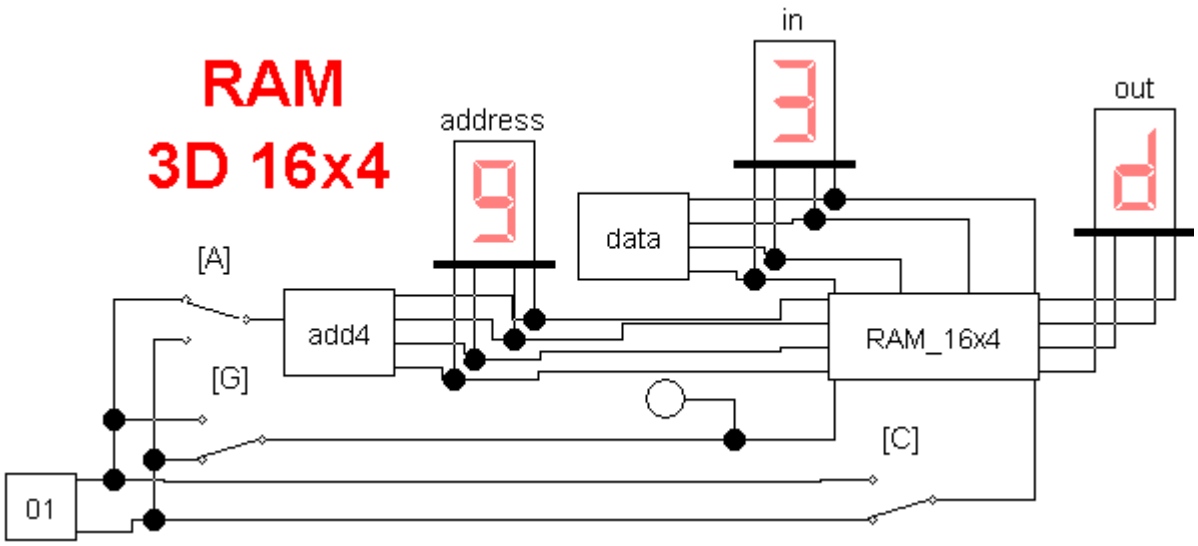

Figure 7.4

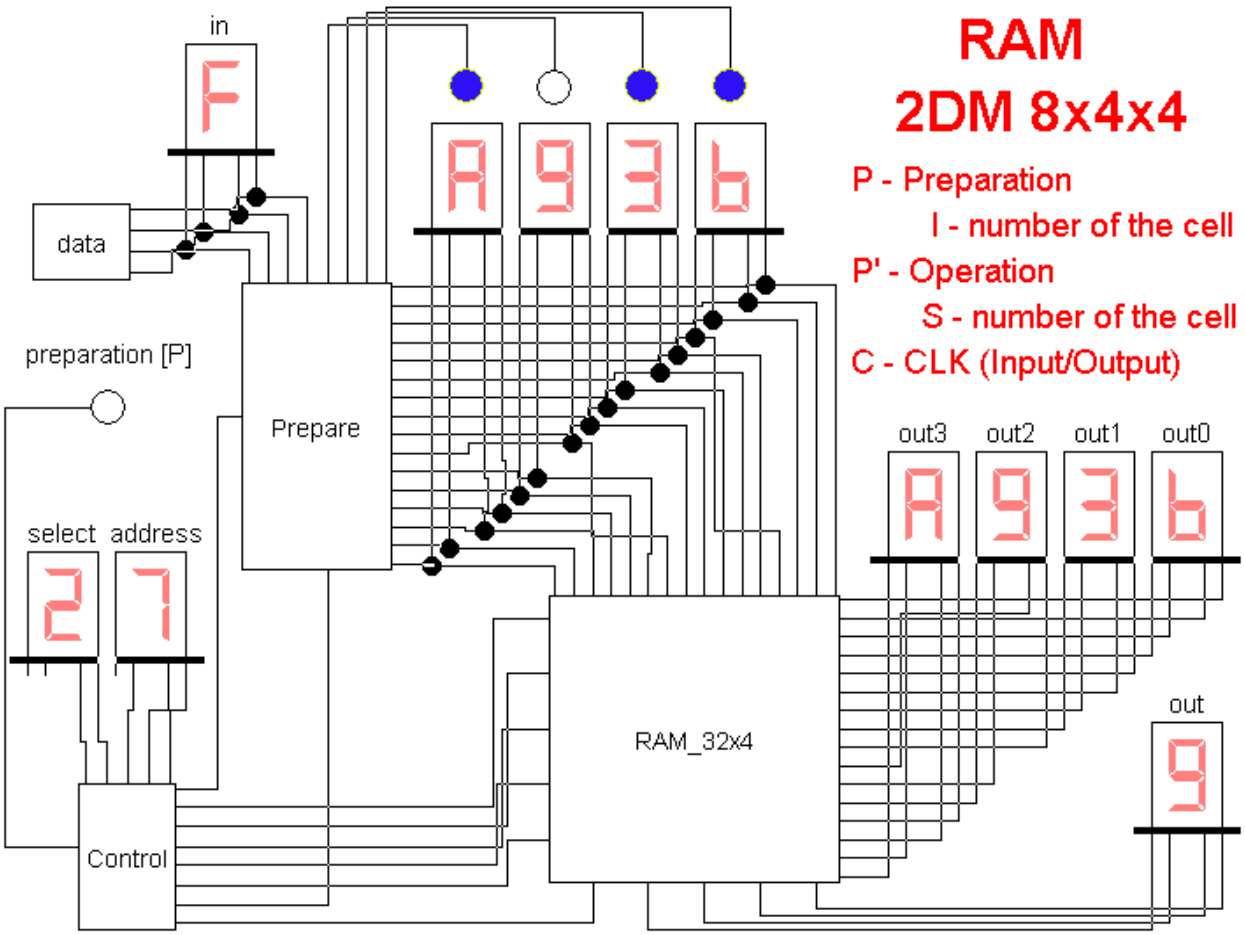

Figure 7.5

The investigation circuit for the memory system on 2DM structure (figure 7.5) consists of the following units:

– RAM_32x4 memory unit on 2DM structure;
– prepare data preparation unit;
– control unit.

RAM_32x4 memory unit consists of following sub-units:

– 32x4 memory matrix;
– dc8 line decoding unit;
– mux4 unit for selecting a cell from the row (while reading).

32x4 memory matrix (logically) represents 8 lines, each of which contains a cell_4x4 unit consisting of four 4-bit memory cells rg4. Physically, the memory unit 32x4 implemented (for the purposes of convenience, connectivity and review) as the union of two 16x4 units composed of four cell_4x4 units. Each of the internal memory array units 32x4 is provided with an indicator that allows you to monitor the row and memory cell selection during experimentation.

The dc8 unit, dedicated to selection of the memory row, consists of a decoder with 8 outputs.

Mux4 unit, assembled using two double 4-input multiplexers, allows selection of a cell from the activated row.

Prepare unit is dedicated to preliminary preparation of four 4-bit words that are subject to simultaneous writing into one of the rows of the memory matrix. During the preparation stage these words are temporarily saved in the inner block in_4, which is comprised of four 4-digit cells of "in" memory. Selection of these cells is done with the help of the decoding unit dc4 and its addressing unit 0_3.

Control unit contains united control keys and additional elements, providing the device operation modes. The device can operate in 3 different modes – preparation of data to be written, write and read modes.

Data preparation mode is performed in the following way:

1) Using the P (Prepare) key, transition the device into data preparation mode (the corresponding LED illuminates).

2) Using the I (In) key, select one of the four cells of the in_4 unit (corresponding LED illuminates).

3) Using the C (CLK) key, write one of the numbers, generated by the data unit into the selected cell.

Transitioning into the read/write mode is performed by pressing the P key again (the indicator stops illuminating).

In the write mode:

1) Using the A key, select the required row of the memory matrix (pointed to by a corresponding 7-segment indicator).

2) Using the C (CLK) key, write the prepared data to the selected row of the memory matrix.

In read mode:

1) Using the A key, select the required row of the memory matrix (pointed to by a corresponding 7-segment indicator).

2) Using the S (Select) key, select the proper cell from the activated row of the memory matrix (pointed to by a corresponding 7-segment indicator).

7.3 Activities

7.3.1 Study the principles of building ROM and its operation procedure:

1) Open the file ROM(M) 16x7.ewb, open the sub-circuits of different units (figure 7.1) and study their structure.

2) By changing the addresses of a memory matrix cell with the help of A (Address) key, verify the information written into the corresponding rows of memory matrix.

7.3.2 Study the principles of building PROM and the procedure for its programming:

1) Open the file PROM 16x7.ewb (figure 7.2), study the structure of the matrix for chip16x7 and the structure of the programming tool, by opening the relevant sub-circuits.

2) Program the rows of the memory matrix to show the symbols from the following list: L, J, H, P, S, U, –, _, e, n, q, r etc.

7.3.3 Study the principles of building 2D structure memory:

1) Open the file RAM 2D_8x4.ewb, where the circuit for studying the 2D structure memory system is located and study the principles of its operation (Figure 7.3). Open the sub-circuits of units and study their structure.

2) investigate the operation of the memory system presented, writing the information to the cells and verifying that it has been written.

7.3.4 Study the principles of building the 3D structure memory system:

1) Open the RAM 3D_16x4.ewb file, where the circuit for studying the 3D structure memory system is located and study the principles of its operation (Figure 7.4). Open the unit sub-circuits and study their structure.

2) Investigate the operation of the memory system presented, writing the information to the cells and verifying that it has been written.

7.3.5 Study the principles of building the 2DM structure memory system:

1) Open the RAM 2DM_32x4.ewb file, where the circuit for studying the 2DM structure memory system is located, study the principles of its operation (Figure 7.5) and study its building blocks by opening them and studying the explanatory text.

2) Study the operation of the RAM_2DM device, by writing the information to all of the rows of the matrix and then verifying the possibility of reading from a specified memory cell.

7.4 Review questions

1. Name the main parameters of memory devices.
2. What is the purpose of ROM?
3. What memory elements are used in different types of ROM?
4. Who and how programs the PROM
5. What is the difference between EPROM and EEPROM?
6. Which circuit design of the memory elements provides maximum RAM performance?
7. What is the principle of creating RAM with one- and two-dimensional addressing?
8. Compare advantages and disadvantages of 2D and 3D structures
9. Explain the principles of implementing 2DM memory structure and its advantages compared to other structures.

8 Laboratory work №8. Model of a microprocessor system

Learning Objectives:

– get familiar with the structure of the microprocessor system (MPS model;

– master the MPS operation procedure;
– studing the actions of different instructions;
– mastering the principles of organizing a repetitive routine structure on an MPS model;
– mastering the techniques of working with data array;
– studing the program for calculating a sum of several numbers;
– mastering the techniques of searching for a single bit in specific data bits.

8.1 Equipment and the file

Computer, Electronics Workbench Professional.
File MP4M.ewb.

8.2 Practical information and methodical recommendations

The model of a 4-bit microprocessor system containing a microprocessor (MP4M), an input device (In), a memory system (Mem), and a system operation mode selection unit are shown in figure 8.1.

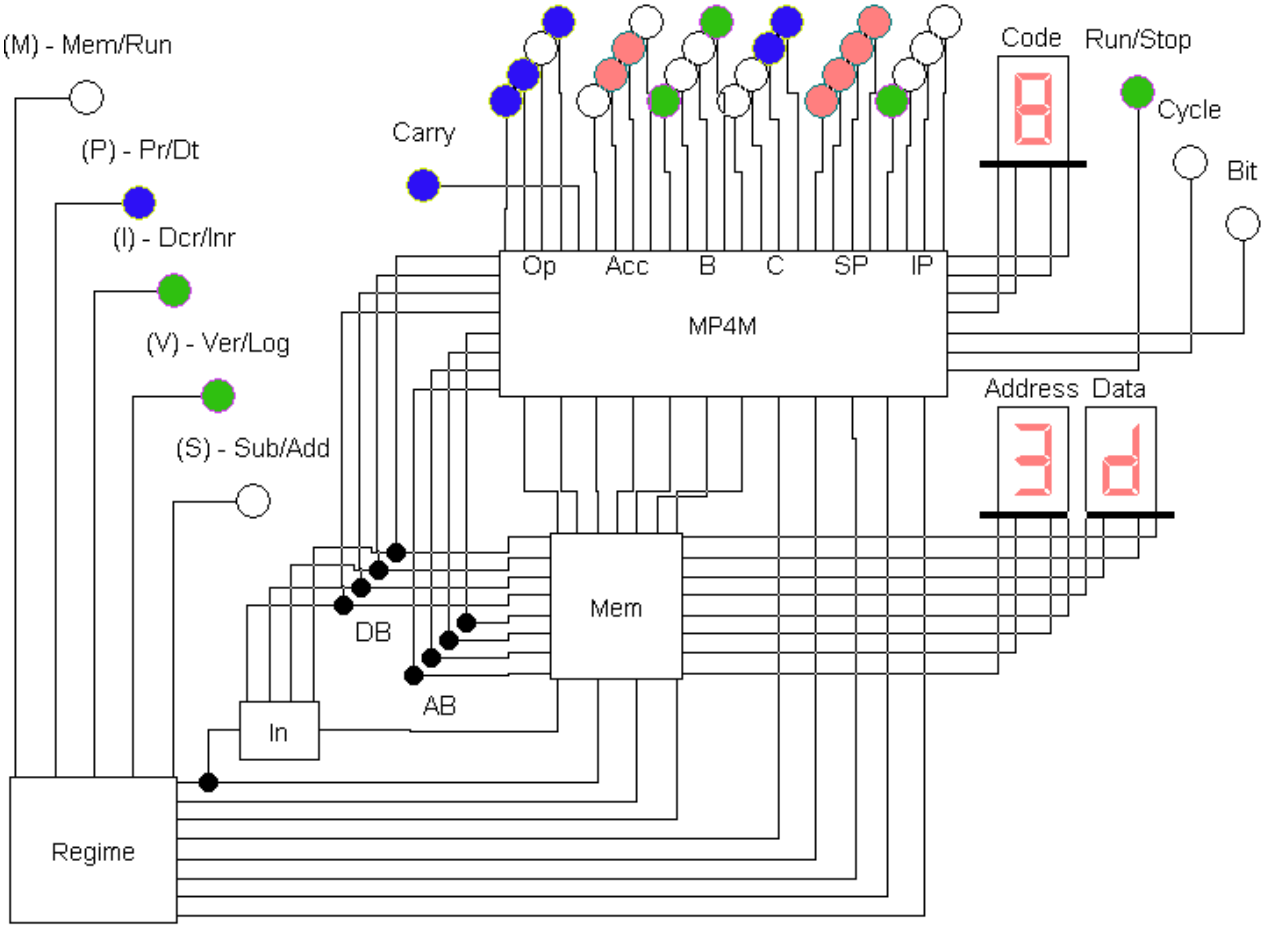

Figure 8.1

The microprocessor consists of following units:

1) Set of specialized registers (RGs, Registers).
2) Arithmetic logic unit (ALU, Arithmetic-Logic Unit).
3) Instruction register (IR, Instruction Register).
4) Instruction decoder (DC_16, Decoder 16-bit).
5) Timer (CLK, Clock).
6) Firmware device (Combi, Combinational Circuit).

The set of registers (RGs) contain the following registers in its structure, each of which is designed to perform certain functions:

1) Register A performs the Accumulator function, where either one of the operands participating in the execution of arithmetic and logical operations or the result of the operation is stored. The accumulator is loaded from memory (instruction LD A) or from ALU (after performing arithmetic or logical operation).

2) Register A1 is used to accumulate carry-overs when organizing loop summation. It is reset when the program is started and when the accumulator is loaded (register A) with the help of the instruction LD A. Display of its contents on the indicators (accumulator) is performed by briefly pressing the Space key.

3) Register B performs the address register function for the move instructions between the memory and the internal registers of the microprocessor (LD C, LD IP, LD SP, LD A). After executing the instructions LD C, LD SP, LD A, the contents of register B are incremented. This register is loaded by the LD B command, which uses register C as an address register. It is reset when the program starts.

4) Register C performs the function of the address register for the second operand involved in performing arithmetic and logical operations. After the corresponding instructions are executed, the contents of register C are incremented, but unlike register B, the increment sign of register C can be changed, it is set by the key I. Its state is shown by the indicator (I) - Dcr / Inr: for I = 0 – Positive (Increment), And for I = 1 - negative (Decrement) increment. Its contents can be left in an unaltered state, which will be specified when describing the operating modes of the system. This register is loaded by the command LD C. When the program is started, this register is set to 1111 (using XOR elements). In view of the fact that any program contains the move instructions, the first instruction of the program must be load register B (instruction LD B) to the data contained in the data memory cell with address 1111. The register C includes a circuit for determining the zero state of the register, upon which a conditional jump is carried out in the organization of loop operations.

5) The SP register performs the function of a stack pointer (Stack Pointer), i.e it performs the function of the address register when executing the PUSH and POP instructions: before executing the PUSH command, its contents are decremented (Predecrement) and after the execution of the POP command - positively incremented (Postincrement). Loading of this register is carried out by the command LD SP. When the program starts, this register is set to 1111.

6) The IP register acts as a Instruction pointer (Instuction Pointer). After selecting the code of the current instruction and saving it in the instruction register, its

contents are incremented. This register is loaded by the LD IP instruction. Its contents are altered when jump conditions defined in some alternative modes of the system operation are met. It is reset when the program starts.

The inputs of all registers are connected to the internal data bus, and the outputs are connected to the data bus or to the address bus (through the tri-state buffer elements), respectively, according to the functions they perform.

The ALU, Arithmetic-Logic Unit performs arithmetic and logical operations and contains various sub-units (sum, not, and, or, xor) intended for the implementation of the corresponding operations, and buffer registers for temporary storage of operands and the result of the performed operation.

The instruction decoder (DC_16), depending on the instruction code, includes the corresponding microprogram controlling the execution of the current instruction, i.e it determines the shape of generating signals that control the operation of the devices involved in its execution.

The timer (CLK) determines the time distribution of control signals transmitted to devices participating in the execution of instructions.

A microprogram device of a combinational type (Combi, Combinational Circuit), generates spatial and temporal control signals that ensure the execution of the current instruction.

Devices, similar to the ones considered earlier, are used as a basis for the input device and the memory system.

The memory control unit (MCon, Memory Control) ensures selection of one of the memory (code or data) units and its operation mode (input, write, read) depending on the control signals coming from the input device or microprocessor.

The system can operate in several modes, which are determined depending on the state of the keys in the mode selection unit. The status of these keys is shown by corresponding indicators. Let us describe these modes.

Loading memory from the input device is performed at M = 1. If, in this case, P = 1, then the write is done into the program memory (the program must end with a halt instruction containing the code F), conversely, with P = 0 - into the data memory. The memory contents are verified by changing the address of the memory location by briefly pressing the Space key. When the P key is switched, the memory address is reset.

Run the program by switching the M key to the M = 0 state. If, in this case, P = 1, then when the arithmetic and logical instructions are executed, the contents of register C (corresponding to the value of I) change, and for P = 0 its contents remain unchanged.

For S = 0, ADS and ACS instructions perform addition operations, and for S = 1, subtraction operations.

For V = 0, with the AND instruction the usual operation of logical multiplication between two operands are performed, and under V = 1, a bit operation is performed for searching for a single bit in a certain bit string of data, for the purposes of which a number containing a 1 in the desired bit must be loaded into the accumu-

lator. The described bit operation simulates the search a request signal from an external device or acknowledgment in real application systems.

The transfer of the system into one of the described above modes is carried out before the program runs, but it can be implemented while the program is running, which expands the software capabilities of the system. For example, during execution process you can modify the addition operations to subtraction and vice versa. Similarly, the conjunction can be changed to a bit operation.

The instructions of the MP4M microprocessor and their characteristics are shown in table 8.1, and order of setting the operating modes of the system is shown in table 8.2.

Table 8.1 – The commands of microprocessor MP4M

Mnem.	Code	Operation			
	Hex	V=0	P	S	V=1; I=1
LD B	0	B←[(C)]; C←(C)+1			
LD C	1	C←[(B)]; B←(B)+1			
LD IP	2	IP←[(B)]			
LD SP	3	SP←[(B)]; B←(B)+1			
LD A	4	A←[(B)]; B←(B)+1			
ST	5	[(B)]←(A);B←(B)+1			
INR	6	A←(A)+1			
NOT	7	A←$\overline{(A)}$			
ADS	8	A←(A)±[(C)]; T_C	P=1: C←var(I) P=0: C = invar	S=1: SUB (–) S=0:ADD,ADC (+)	Zc = 0: IP←IP+1
ACS	9	A←(A)±[(C)±T_C; T_C			
AND	A	A←(A)∧[(C)]			Zc∧Za=0: IP←IP+1
OR	B	A←(A)∨[(C)]			
XOR	C	A←(A)⊕[(C)]			
PUSH	D	SP←(SP)–1; (SP)]←(A)			
POP	E	A←(SP)]; SP←(SP)+1			
HLT	F	IP = invar			

Table 8.2 – Setup of the operating modes of the MP4M microprocessor

Keys and indicators	M/R = 1		M/R → 0: Running: A,B,IP ← 0; C,SP ← F	
			P=1: C←var(I);	P=0: C = invar
S = 0	P/D = 1: MemPr←IN	P/D = 0: MemD←IN	Addition (+)	A ← (A) + [(C)]
S = 1			Subtraction (–)	A ← (A) – [(C)]
I = 1			PostDecrement C←(C) –1	C←(C)–1; (C)=0: Zc ← 0
I = 0			PostIncrement C←(C) +1	C ← (C) + 1
V = 0			Logical Operation	A ← (A) {∧ / ∨ / ⊕} [(C)]
V = 1			Verifying of Byte	(A)=0010…b_i = 1: Za ← 0

8.3 Activities

8.3.1 Study the move instructions:

1) Open the program EWB5PRO and select file MP4M.ewb. Expanding the subcircuits of the main units of the system (Figure 6.1) and their internal components, study the practical information about the system given above, familiarize yourselves with its structure and principles of operation.

2) Activate the EWB5PRO program with the  button and then it should not be turned off until the end of the work in order to avoid loss of the recorded information.

3) Prepare the system for the program loading mode by setting the keys M and P to state M=1 and P=1. Load the instruction codes of the following program into memory.

Address	Commands	Code	Address	Commands	Code
0	LD B	0	5	POP	E
1	LD A	4	6	ST	5
2	LD SP	3	7	PUSH	D
3	LD C	1	8	PUSH	D
4	POP	E	9	HLT	F

4) Switch the system into the data loading mode by setting the P key to state P=0. Load data into memory (for example, sequential values of numbers, starting with a specific value) and write them into a workbook.

5) Check the correctness of the entered information (program and data) in memory by selecting the required memory type with the P key and briefly pressing the Space key.

6) Run the loaded program by setting the P key to the state P=1 and switch M key to state M=0. Observing the status of the indicators, verify the operation of the program. If necessary, you can pause the program by pressing the 

button or restart the program by double-clicking M (not very fast).

7) After the program finishes, switch the system to the input mode and check the contents of the data memory cells. Find out the reason of the changes.

8) After restoring the previously entered data and changing the contents of the address cell F, run the program and verify its operation.

8.3.2 Study the arithmetic and logical instructions

1. Prepare the system for the program loading mode (M=1, P=1) and load into memory the instruction codes of the programs listed below.

Address	Commands	Code	Address	Commands	Code
0	LD B	0	7	ADS	8
1	LD C	1	8	ACS	9
2	LD A	4	9	ACS	9
3	INR	6	A	AND	A
4	INR	6	B	OR	B
5	ADS	8	C	XOR	C

6	ADS	8	D	HLT	F

2. Switch the system to the data load mode (M=1, P=0). Load data into memory (for example, sequential values of numbers, starting with a specific value) and write them down into a workbook.

3. Run the loaded program (P=1, then M=0). Observing the status of the indicators, verify the operation of the program. If necessary, you can use the pause button 

.

4. Having finished the programm, check the contents of the A1 register with a brief press of the Space key (this can also be done during the program's operation). Explain the result. If there is any difficulty in this, restart the program (by double-pressing key M) and monitor more closely the emerging shifts in addition operations.

5. Switch the system into the program load mode, change the state of the key S to S=1, as a result, the addition program will transform to the subtraction program, i.e. when the ADC and ACS commands are executed, subtraction operations are performed (respectively – without and with carry).

6. Run the loaded program. Observing the status of the indicators, verify the operation of the program. If necessary, you can restart the program.

7. Considering that arithmetic operations are performed in the additional code, manually perform in binary code all subtraction operations carried out in the program and compare the results with the results obtained in the system.

8.3.3 Study the principles of the organization of looping program structures on the MPS model:

– prepare the system for program loading mode and load into the memory instruction codes of the program given below and the data for the program for determining the sum of N numbers (in this case N=5);

Address	Commands	Code	Address	Data	Address	Data
0	LD B	0	0	…	…	…
1	LD C	1	1	A	7	5
2	ADS	8	2	B	8	2
3	LD IP	2	3	C	…	…
4	HLT	F	4	D		
			5	E	F	7

– run the loaded program with P=1, S=0, I=1. After the program is finished, the first 4 least significant bits of the addition result will be in the accumulator (A register), the four most significant bits, which are contained in the A1 register, can be found by briefly pressing the Space key;

– manually determine the sum of numbers located in the cells with addresses 1…5, and compare it with the result obtained in the system;

– switch the system into boot mode, load into the memory the instruction codes of the following program for detecting a 1 bit in the third bit among the data

located in cells with addresses 1…5 (in this case - number D, located in cell with address 2);

Address	Commands	Code	Address	Data	Address	Data
0	LD B	0	0	6	…	…
1	LD C	1	1	8	7	5
2	LD A	4	2	D	8	4
3	AND	A	3	A	9	3
4	LD IP	2	4	9	…	…
5	HLT	F	5	B	F	7

– run the loaded program. Observing the status of the indicators, check the operation of the program. When a required number is found, the Bit indicator lights up and the program finishes its work;

– put the system into loading mode and change value of D, located in cell with address 2, to 3 and restart the program. In this case, the program will check the data in cells with addresses 1…5 and, not finding among them the required number, will finish its work.

8.4 Review questions

1. Explain the structure of the model of the microprocessor system and its functional units.
2. Explain the internal structure of the microprocessor and the functions of its constituent components.
3. Explain the move instructions used in the examined program.
4. Explain arithmetic and logical instructions used in the program. Explain the order of the corresponding operations with examples.
5. Explain the principles of organizing a looping program structure on the training model.
6. Explain how to work with the array of data.
7. Explain the structure of the program for calculating the sum of numbers.
8. Explain the structure of the program for unit detection in a particular bit in the data array.

References

1. Shanaev O.T. System modeling of Electronic Workbench / on Kazakh and Russian languages. – Almaty: AUPET, 2003.

2. Ugrumov E. O. Digital circuit design. – СПб.: БХВ-Petersburg, 2004.

3. Khamaher K. I др. Organisation ECM. Trans. from. eng. – СПб.: Petersburg, 2003.

4. Sugano T. J., Uydmer H. C. Digital system. Theory and practice: Trans. from eng. – M.: Publishing house «Williams», 2004.

Contents