F.A. Gubarev,
M.E. Nakonechnaya

DIGITAL DEVICES

TOMSK POLYTECHNIC UNIVERSITY

F.A. Gubarev, M.E. Nakonechnaya

DIGITAL DEVICES

Practical Course

Recommended for publishing as a study aid
by the Editorial Board of Tomsk Polytechnic University

Tomsk Polytechnic University Publishing House
2013

UDC 004.31(075.8)
BBC 32.973.2я73
G93

Gubarev F.A.

G93 Digital Devices. Practical Course: study aid / F.A. Gubarev, M.E. Nakonechnaya; Tomsk Polytechnic University. – Tomsk: TPU Publishing House, 2013. – 112 pp.

In the Practical Course the main operation principles of digital automatics units are discussed. The complex of practical exercises is offered students to consolidate the theory and skills of electronic circuits design on the basis of digital devices. It contains nine practical works accompanied by methodical instructions and required theoretical knowledge.

This study aid is intended for the students, studying courses "Digital Devices" and "Microprocessor Technique". It also can be helpful for students of all specialities related to Electrophysics and Electronic Equipment.

UDC 004.31(075.8)
BBC 32.973.2я73

Reviewers

PhD, associated professor of Tomsk State Pedagogical University
Dean of the Faculty of Foreign Languages
I.E. Vysotova

PhD, associated professor of the Tomsk Polytechnic University
D.N. Ogorodnikov

PhD, senior staff scientist of V.E. Zuev Institute of Atmospheric Optics
D.V. Apeksimov

INTRODUCTION

Nowadays it is hard to imagine our lives without modern digital devices. In the morning digital alarm-clocks wake people up, digital watches inform us about the beginning of a new day and are even able to play brisk melodies. A microwave oven warms our breakfast up measuring the time till its ready. When we go to our work we often use elevators, which let us easily and quickly reach the required floor. A great many of traffic lights make it safe to move around the city.

It is also impossible to imagine modern enterprises without personal computers and workstations. Here huge quantities of information which can be provided for a consumer on demand are stored. This is only a small number of applications of digital circuits – logic gates, registers, counters, timers, switches, decoders, adders, converters, etc. They make up all the devices considered above – digital watches, timers, lift and traffic lights control circuits, computers.

The first and the most essential factor for the constructed circuits to operate without fail and provide our lives with comfort is to know the physical principles of the simplest digital devices functioning and the methods of design of complex systems made on their basis. Digital electronics plays an important role in providing high reliability of the created automatic and computer-aided systems, which control objects, processes and manufacturing systems.

It is obvious that the modern digital devices are more and more seldom designed using simple discrete integrated circuits (IC) – ordinary logic gates, counters, registers, and flip-flop. At the same time all these elements being the components of more complicated circuits form the basis of programmable logic device (PLD), microcontrollers, and microprocessors. Therefore, the knowledge of fundamentals of digital technique is necessary when studying the disciplines connected with microprocessor-based system.

This teaching aid is a laboratory practical course for the discipline named ‘Digital devices’. The book contains methodical guidelines for nine lab works covering three main course units: basic logic gates, combinational logic devices (multiplexers, decoders, arithmetic units), and sequential logic devices (flip-flops, counters, registers). Every lab work is supplied with theoretical background. If a student wants to check his knowledge of the studied material, he is offered some questions at the end of each work.

Methodical instructions presented here and logic circuits recommended for studying require a laboratory bench for circuits assembling and connecting of the required pins of ICs. The commutation process on the

pinboard can be done either manually or by means of computer-aided systems.

The practical classes in ‘Digital devices’ are given at Industrial and Medical Electronics department of Tomsk Polytechnic University using a full-scale modelling system of electronic circuits, which was designed and produced at the department [1, 2]. The laboratory module UIK-1 has a set of slots for placing DIP14 и DIP16 integrated circuits, and also a kit of built-in elements (pulse generator, counter, seven-segment display, light-emitting diodes, key buttons, and pull-up and pull-down resistors). The module is connected with the personal computer through USB. The electronic circuit is represented at the screen of the personal computer graphically with the help of the relevant software. The circuit synthesis is executed directly in the module. Such modelling environment let us observe signals at any pin of a real IC with an oscilloscope.

The given teaching aid is based on the lab work instructions [3] which were worked out for the course ‘Digital devices’ previously. The laboratory course was extended for working with the module UIK-1. The description and pin configuration of the ICs used in the lab works are taken from the reference books [4–7] and manufacturer free access datasheets [8]. Some information about the operational principle of discussed devices you can find in [9–12].

Lab 1

ELECTRONIC CIRCUITS SYNTHESIS ACCORDING TO THE GIVEN FUNCTION

1.1 OBJECTIVES

The lab work acquaints students with basic logic elements and their functions. A widespread series ICs KR1533 (analog SN74ALS) is chosen as an example. The work also helps to develop the skills in minimizing logic functions and circuit synthesis in various element bases.

1.2 PRE-LAB TASKS

1. Acquaint with the operational principles and pin configuration of the ICs KR1533LA1 (SN74ALS20AN), KR1533LA3 (SN74ALS00AN), KR1533LA4 (SN74ALS10AN), KR1533LE4 (SN74ALS27AN), and KR1533LE1 (SN74ALS02N).
2. Study the aspects of simplifying Boolean functions (BF).
3. Conduct BF minimization by Karnaugh map method according to the lecturer's task. Obtain the minimized algebraic expression in the disjunctive normal form (DNF).
4. Implement the obtained function using NAND universal gates.
5. Present a voltage diagram for the given function; show the time intervals when critical races are possible to emerge.
6. Implement the function on the basis of NOR universal gates.
7. Present voltage diagrams for the given function; show the time intervals when critical races are possible to emerge.

1.3 BASIC THEORY

Any digital computer consists of logic circuits. These are circuits which can be only in two states – either 'logic zero' or 'logic one'. Any expression (including a word) which can be characterized as 'true' or 'false' is assumed as logic zero or logic one. In electronic engineering logic 0 and 1 are the certain states of electric circuits. For example, if we talk about logic elements and circuits carried on TTL-technology (transistor-transistor logic), logic 0 will mean voltage range from 0 to +0.4 V, and logic 1 – from +2.4 to +5 V.

The operation of logic circuits is characterized by a special mathematical tool called Boolean algebra or logical algebra. Boolean algebra was developed by an English mathematician and logician George Boole (1815–

1864) and is the fundamental of all methods of logical expressions simplification.

1.3.1 Basic postulates of Boolean algebra

Laws for a single variable

$$X+0=X \qquad X\times X=X$$
$$X\times 0=0 \qquad X+\overline{X}=1$$
$$X+1=1 \qquad X\times\overline{X}=0$$
$$X\times 1=X \qquad \overline{\overline{X}}=X$$
$$X+X=X$$

Laws for two and more variables

1. Commutative law:

$$X+Y+Z=Y+X+Z=Z+X+Y,$$
$$X\times Y\times Z=Y\times X\times Z=Z\times X\times Y.$$

2. Associative law:

$$X+Y+Z=(X+Y)+Z=(Z+Y)+X,$$
$$X\times Y\times Z=(X\times Y)Z=(Z\times Y)X.$$

3. Distributive law:

$$X(Y+Z)=X\times Y+X\times Z,$$
$$X+(Y\times Z)=(X+Y)(X+Z).$$

4. Absorption law:

$$X+X\times Y=X(1+Y)=X,$$
$$X(X+Y)=X+XY=X.$$

5. Sewing law:

$$X\times Y+\overline{X}\times Y=Y(X+\overline{X})=Y,$$
$$(X+Y)(\overline{X}+Y)=\overline{X}\times X+X\times Y+Y\times\overline{X}+Y\times Y=Y.$$

6. Law of dualization (de Morgan´s Theorem):

$$\overline{X+Y}=\overline{X}\times\overline{Y},$$
$$X+Y=\overline{\overline{X}\times\overline{Y}},$$
$$\overline{X\times Y}=\overline{X}+\overline{Y},$$
$$X\times Y=\overline{\overline{X}+\overline{Y}}.$$

1.3.2 Karnaugh map method

Karnaugh maps are used to facilitate the simplification of Boolean algebra functions. Karnaugh map is a type of truth table representation. Every

cell of the Karnaugh map corresponds to the row of the truth table. Along each axis variable combinations are plotted, and inside the map the value of the function is given.

The purpose of the Karnaugh map is to find out the logical sum of the direct and inverse variable values. For any variable, for example *a,* the logical sum equals $a + \bar{a} = 1$. If $a = 0$, we have 0+1=1; if a=1, we get 1+0=1. Therefore, when we put this expression in brackets,

$$abc + \bar{a}bc = bc(a + \bar{a}) = bc \cdot 1 = bc$$

the sum $a + \bar{a} = 1$ can be neglected, and the result of the formula will remain the same. This example shows how the simplification of the logical expression on the basis of Karnaugh map works. Tables 1.1–1.3 present Karnaugh maps of different ranks.

Table 1.1

2-variable Karnaugh map

B/A	*0*	*1*
0	$\bar{A}\bar{B}$	$A\bar{B}$
1	$\bar{A}B$	AB

Table 1.2

3-variable Karnaugh map

C/AB	*00*	*01*	*11*	*10*
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Table 1.3

4-variable Karnaugh map

CD/AB	*00*	*01*	*11*	*10*
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$AB\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
01	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}B\bar{C}D$	$AB\bar{C}D$	$A\bar{B}\bar{C}D$
11	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$ABCD$	$A\bar{B}CD$
10	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}BC\bar{D}$	$ABC\bar{D}$	$A\bar{B}C\bar{D}$

How to simplify Boolean functions with the help of Karnaugh map

1. Algebraic expression in the disjunctive normal form (as a rule, full DNF) is put down.
2. According to the number of variables a corresponding rank of Karnaugh map is chosen.
3. 1 is put into those cells of the map, whose summands are present in the algebraic expression of the given function.

4. Then 2^n neighbouring 1 are joined by the common contour (every contour can only consist of 1, 2, 4…2^n logic ones)
5. In the obtained contours complementary variables are excluded.
6. The simplified Boolean function is written down.

Example 1. Function simplification and synthesis in the NAND basis

E.g. BF $F = AB\overline{C}D + AB\overline{C}\overline{D} + \overline{A}\overline{B}CD + A\overline{B}CD + ABCD$ is given. Table 1.4 presents the dependence of *F* on variables *A*, *B*, *C* and *D*.

Table 1.4

Truth table for F-function

Decimal number	*A*	*B*	*C*	*D*	*F*
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	1

The 4-variable Karnaugh map for the given BF is shown in Fig. 1.1.

First contour: $AB\overline{C}\overline{D} + AB\overline{C}D = AB\overline{C}\left(D + \overline{D}\right) = AB\overline{C}$.

Second contour: $AB\overline{C}D + ABCD = ABD\left(C + \overline{C}\right) = ABD$.

Third contour: $\overline{A}\overline{B}CD + A\overline{B}CD = \overline{B}CD\left(\overline{A} + A\right) = \overline{B}CD$.

Finally, we obtain the simplified expression of the given BF:

$F = AB\overline{C} + ABD + \overline{B}CD$.

According to the Law of dualization (de Morgan´s Theorem) the function in the NAND basis can be presented as:

$F = AB\overline{C} + ABD + \overline{B}CD = \overline{\overline{AB\overline{C}} \cdot \overline{ABD} \cdot \overline{\overline{B}CD}}$.

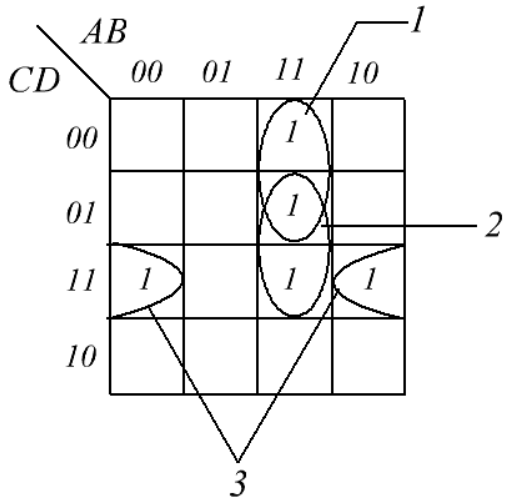


Fig. 1.1. Karnaugh map for the function in the 1st example

Example 2. Function simplification and synthesis in the NOR basis

E.g. Boolean function $F = ABC\overline{D} + AB\overline{C}\,\overline{D} + AB\overline{C}D + \overline{A}\,\overline{B}CD + \overline{A}\overline{B}CD$ is given. Table 1.5 presents the dependence of F on variables A, B, C and D. The four-rank Karnaugh map for the given BF is shown in Fig. 1.2.

Table 1.5

Truth table for F-function

Decimal number	*A*	*B*	*C*	*D*	*F*
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

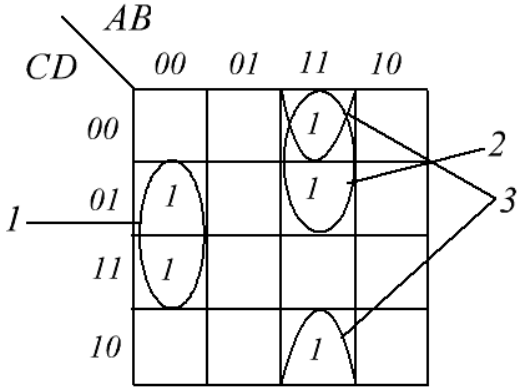


Fig. 1.2. Karnaugh map for the function in the 2nd example

First contour: $\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD = \overline{A}\overline{B}D$.

Second contour: $AB\overline{C}\overline{D} + AB\overline{C}D = AB\overline{C}$.

Third contour: $AB\overline{C}\overline{D} + ABC\overline{D} = AB\overline{D}$.

Finally, we obtain the simplified expression of the given BF:

$F = \overline{A}\overline{B}D + AB\overline{C} + AB\overline{D}.$

According to the de Morgan´s Theorem the function in the NOR basic can be presented as

$$F = \overline{A}\overline{B}D + AB\overline{C} + AB\overline{D} = \overline{\overline{\overline{A}\overline{B}D}} + \overline{\overline{AB\overline{C}}} + \overline{\overline{AB\overline{D}}} = \overline{\overline{A + B + \overline{D}} + \overline{\overline{A} + \overline{B} + C} + \overline{\overline{A} + \overline{B} + D}}.$$

1.3.3 Integrated circuits used in the lab work

IC KR1533LA1 (SN74ALS20AN) is two 4NAND elements. Fig. 1.3 shows its logic diagram and pin configuration. The operation of one section of the IC is described in table 1.6.

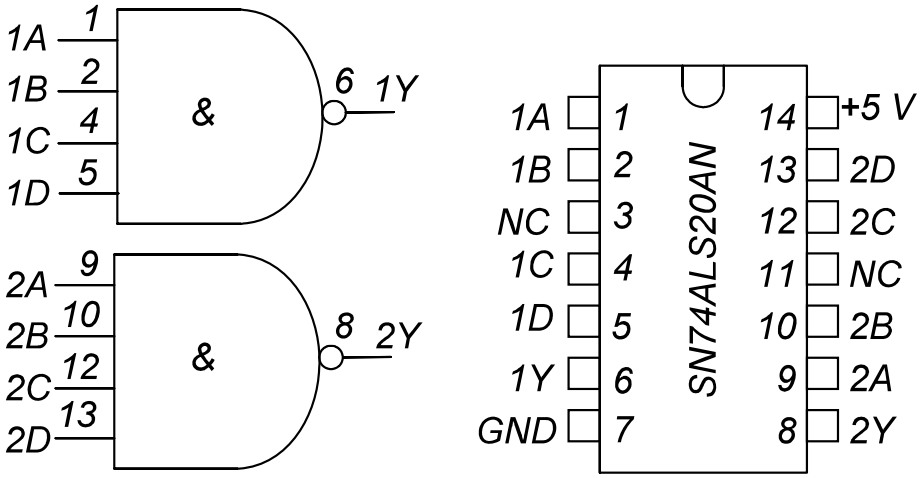


Fig. 1.3. IC KR1533LA1 logic diagram and pin configuration

Table 1.6

Truth table for one section of IC KR1533LA1

Inputs				Outputs
A1	*B1*	*C1*	*D1*	*F1*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

IC KR1533LA3 (SN74ALS00AN) contains four 2NAND elements. Fig. 1.4 shows its logic diagram and pin configuration. The operation of one section of the IC is described in table 1.7.

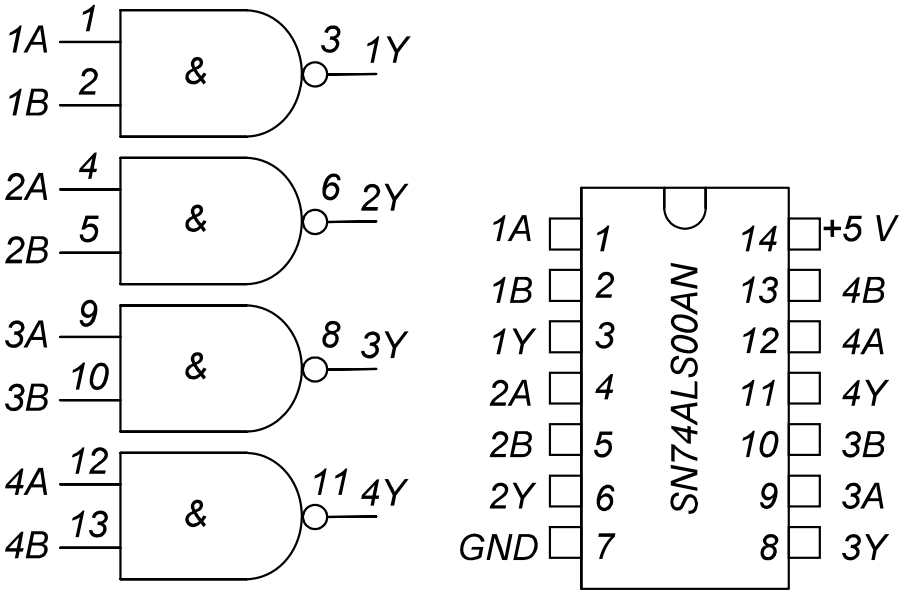


Fig. 1.4. IC KR1533LA3 logic diagram and pin configuration

Table 1.7

Truth table for one section of IC KR1533LA3

Inputs		Output
A1	*B1*	*F1*
0	0	1
0	1	1
1	0	1
1	1	0

IC KR1533LA4 (SN74ALS10AN) consists of three 3NAND elements. Fig. 1.5 shows its logic diagram and pin configuration. The operation of one section of the IC is described in table 1.8.

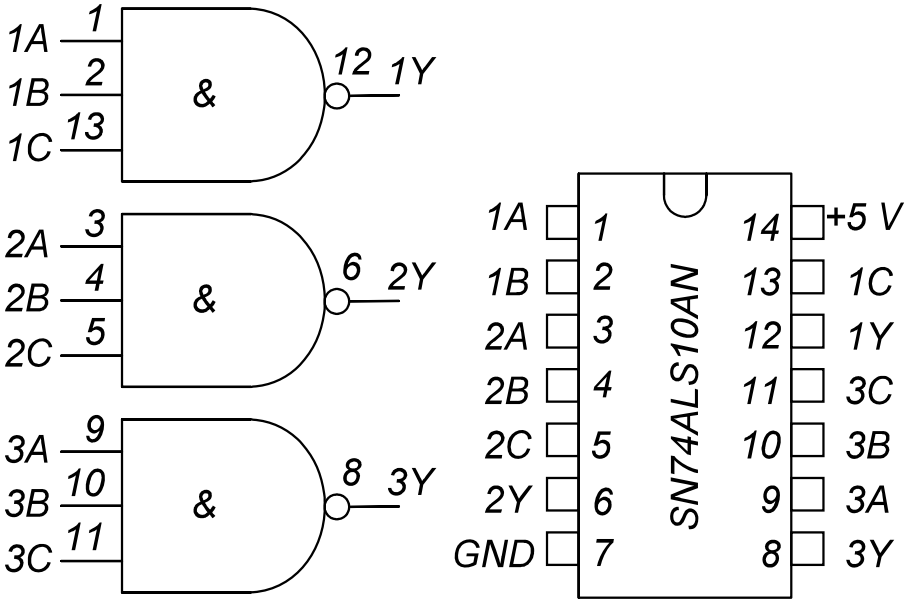


Fig. 1.5. IC KR1533LA4 logic diagram and pin configuration

Table 1.8

Truth table for one section of IC KR1533LA4

Inputs			Output
A1	*B1*	*C1*	*F1*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

IC KR1533LN1 (SN74ALS04AN) contains six independent inverters (NOT gates). Fig. 1.6 shows its logic diagram and pin configuration.

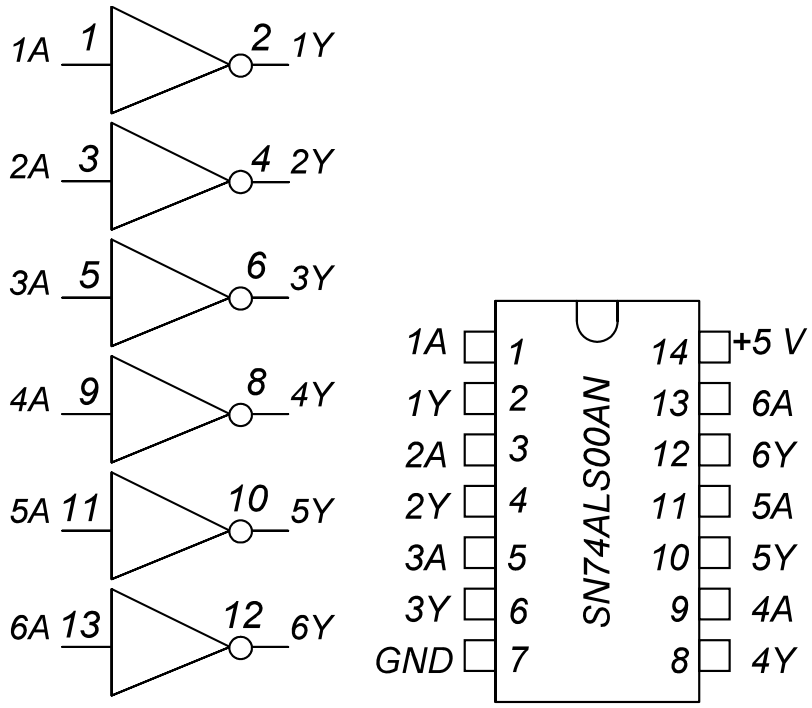


Fig. 1.6. IC KR1533LN1 logic diagram and pin configuration

IC KR1533LE1 (SN74ALS02N) is four 2NOR elements. Fig. 1.7 shows its logic diagram and pin configuration. The operation of one section of the IC is described in table 1.9.

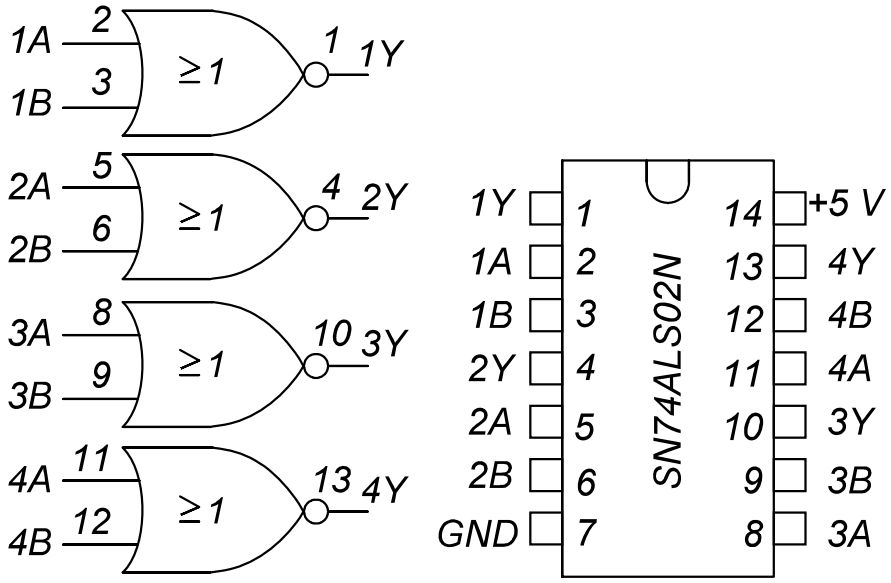


Fig. 1.7. IC KR1533LE1 logic diagram and pin configuration

Table 1.9

Truth table for one section of IC KR1533LE1

Inputs		Output
A1	*B1*	*F1*
0	0	1
0	1	0
1	0	0
1	1	0

IC KR1533LE4 (SN74ALS27AN) consists of three 3NOR elements. Fig. 1.8 shows its logic diagram and pin configuration. The operation of one section of the IC is described in table 1.10.

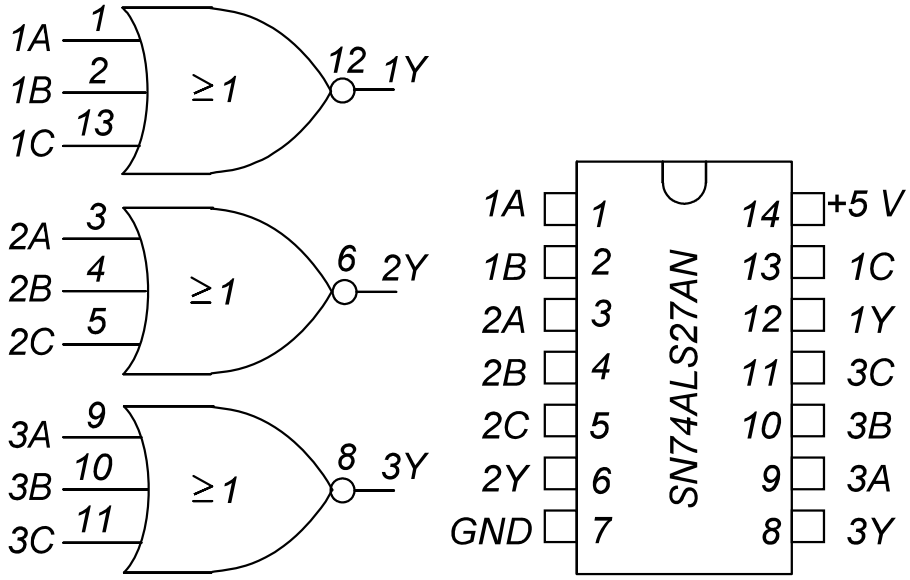


Fig. 1.8. IC KR1533LE4 logic diagram and pin configuration

Table 1.10

Truth table for one section of IC KR1533LE4

Inputs			Output
A1	*B1*	*C1*	*F1*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

IC KR1533IE7 (SN74ALS193N) is 4-bit up-down counter (will be considered in the Lab work No 7).

1.3.4 Examples of BF implementation

NAND elements based circuit

Fig. 1.9 shows BF circuit implementation (example 1) on the basis of logical elements NAND.

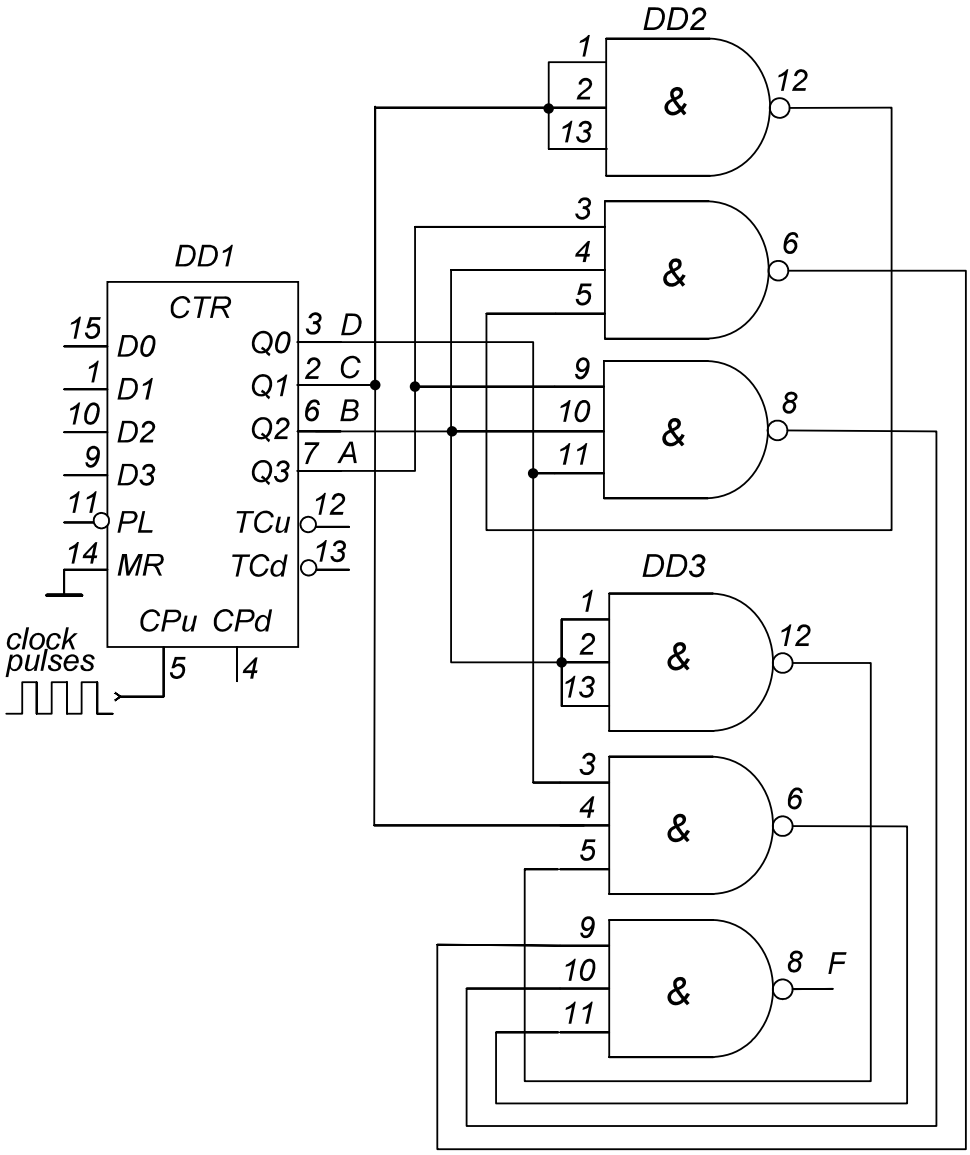


Fig. 1.9. BF implementation on the basis of logical elements NAND

The formation of the variables *A*, *B*, *C* and *D* is performed via the 4-bit up-down counter KR1533IE7. For this purpose the signals from the built-in frequency divider (embodied into the IC KR1533IE19) or built-in pulse generator output are supplied to the pin +1 of the counter. The number of the

frequency divider output is chosen on account of stable operation of a circuit. For the successful performance of the counter KR1533IE7 input *R* must be grounded (connected to the common bus).

Fig. 1.10 presents voltage diagrams for the circuit in Fig. 1.9. Table 1.11 shows circuit pins connection to the common point and power supply.

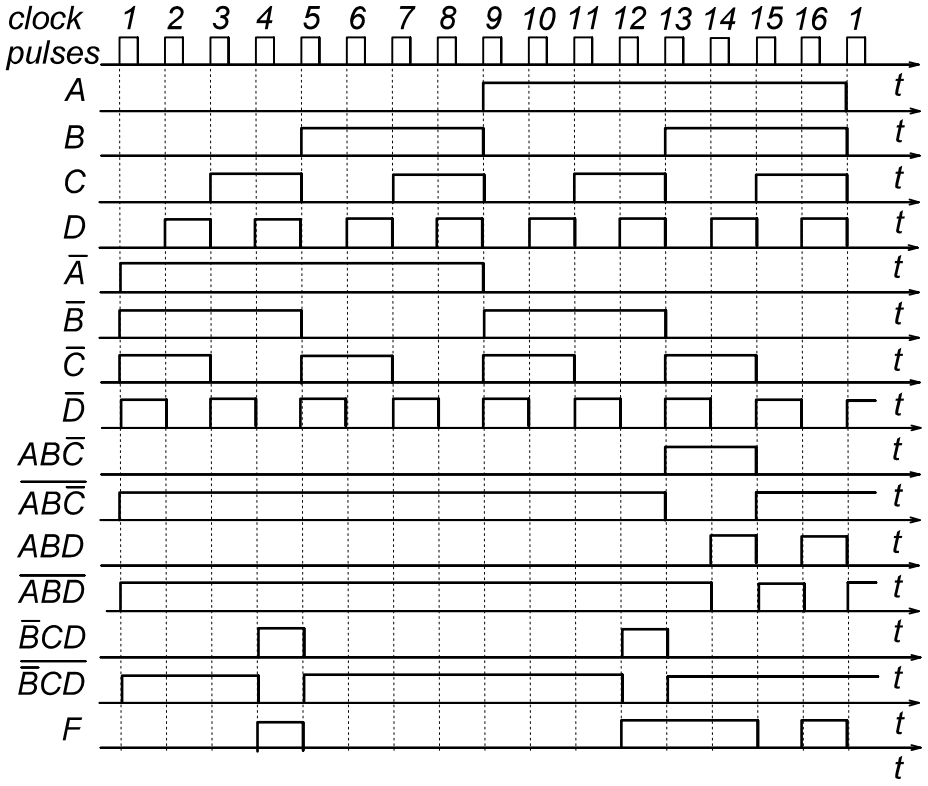


Fig. 1.10. Voltage diagrams for the circuit in Fig. 1.8

Table 1.11

ICs for the circuit in Fig. 1.9

Type of IC	KR1533IE7	KR1533LA4
Circuitry symbol	DD1	DD2, DD3
Common	8	7
+ 5 V	16	14

NOR elements based circuit

Fig. 1.11 shows BF implementation (example 2) on the basis of logical elements NOR. The formation of the variables *A*, *B*, *C* and *D* is performed in the same manner as it was described above.

Fig. 1.12 presents voltage diagrams for the given circuit. Table 1.12 shows circuit pins connection to the common bus and power supply.

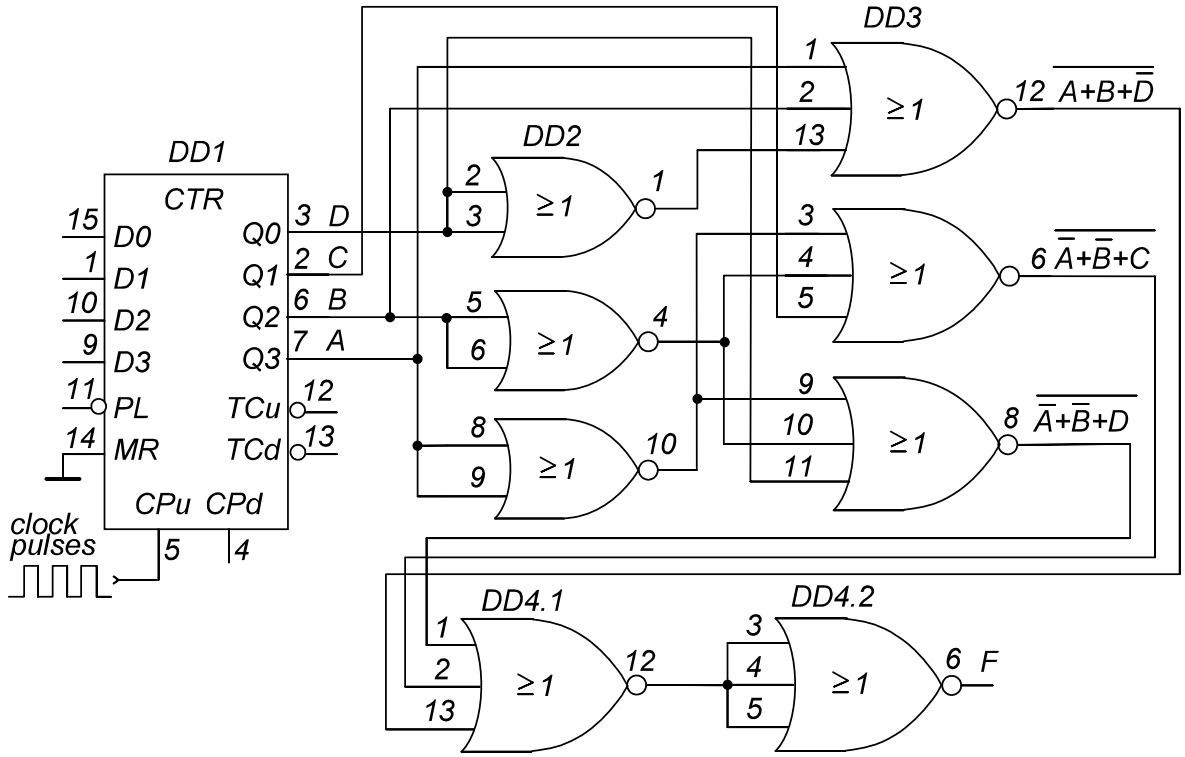


Fig. 1.11. BF implementation on the basis of logical elements NOR

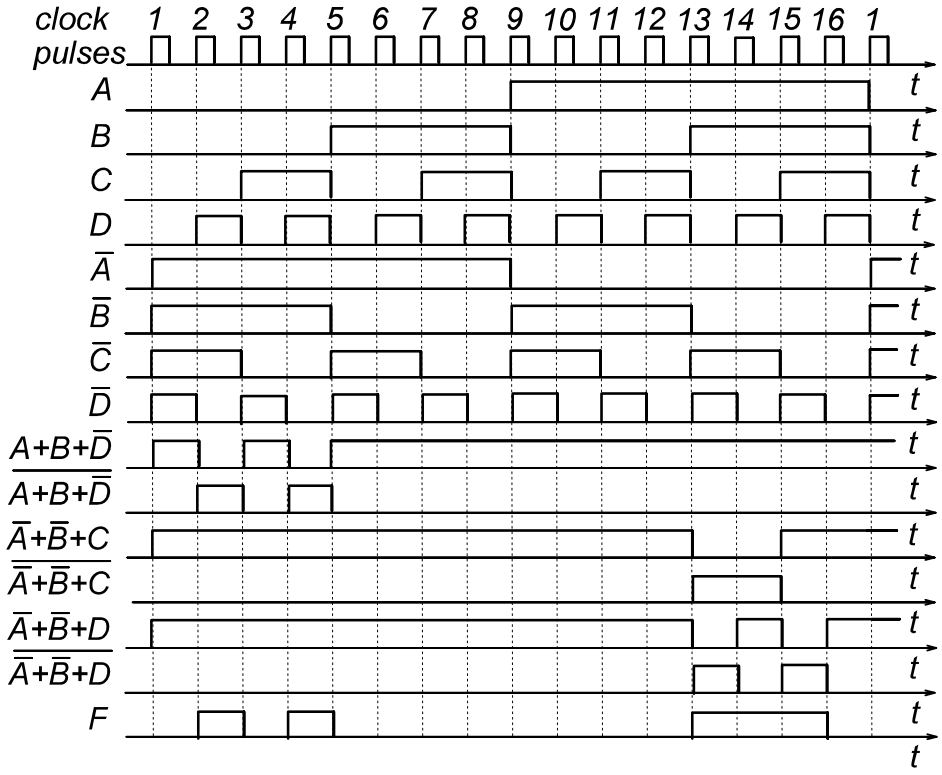


Fig. 1.12. Voltage diagrams for the circuit in Fig. 1.11

Table 1.12

ICs for the circuit in Fig. 1.11

Type of IC	KR1533IE7	KR1533LE1	KR1533LE4
Circuitry symbol	DD1	DD2	DD3, DD4
Common	8	7	7
+ 5 V	16	14	14

1.4 EQUIPMENT

In the lab work the module UIK-1 with the relevant software is used. Before starting the work with the module the students are recommended to study carefully its configuration and the software. They are also supposed to understand electronic circuit design principle using the module and the command assignment in the PC application window.

The module is equipped with special ZIF sockets (zero-insertion-force socket) for mounting and fixing microcircuits: 4 sockets for ICs with 14 pins and 4 sockets for IC with 16 pins. To the left and to the right of the connectors we can see a number of plated holes which are connected to the corresponding clips for the IC pins and are intended for oscilloscope probes placing. ***Notice***: there are no plated holes for the IC power supply pins (7, 14 or 8, 16).

The module is equipped with a built-in clock pulse generator, frequency divider (on the basis of a binary counter KR1533IE19), seven-segment display, two light-emitting diodes with pull-up resistor, two key buttons connected to the common bus and power supply through a resistor, two 1 kΩ resistors for setting logic 1 state, and two terminals for interconnecting external devices (generators, oscilloscopes, voltmeters, etc.), which can be added to any part of the circuit.

By default, a built-in generator (integrated circuit SG51P) is not displayed in the program window. It should be added by putting a tick '√' in the menu 'Show/Hide circuit elements'. In the menu you can choose what devices built in the module will be displayed on the pinboard.

How to work with the module:

1. Connect the module to the computer via USB-interface.
2. Connect the module to the mains ~220 V. A light-emitting diode will glow green informing that there is power supply of ICs.
3. Open the file *ddevice.lnk* at the Windows desktop and choose "Digital devices" option in the pop-up window.
4. Make sure that the device is ready to work: the sign 'Device is found' appears. If the sign 'Device is not found' or 'Device is preparing'

appears, shut down the programme, switch the module off and repeat steps 1–3. Sometimes a computer should be reloaded.

5. Insert the IMC into the slots according to the number of pins. Do not embed microcircuits with 14 pins into the slots with 16 clips. Microcircuits must be put into the slots with a key upward. The module has a built-in wiring of the common and power supply buses. Therefore, when the module is on, there is a supply voltage +5 V between left bottom and right upper clips at each socket. **Be careful when mounting microcircuits!**

6. Assemble the required circuit in the program window. **When connecting the elements pay attention to the outputs of the IC. They shouldn't be connected together as well as connected to the power supply or common bus.**

7. It is highly recommended to save data (button 'Save' at the program command panel) in order not to lose the results of the work.

8. Press the button 'Download data' at the side panel. Only when you push it, the changes made in the circuit are transferred to the module and switching occurs.

To register the waveform and parameters of the output function double-channel oscilloscope is used.

To fulfil the first laboratory work you'll need a set of simple logic ICs: KR1533LA1, KR1533LA3, KR1533LA4, KR1533LE4, and KR1533LE1 or their analogs. To form the variables *A*, *B*, *C*, *D* binary counter KR1533IE7 is used.

1.5 IN-LAB TASKS

1. Connect the counter KR1533IE7. With the help of the built-in divider set minimum clock pulse repetition frequency, make sure that the signals *A*, *B*, *C* и *D* exist.

2. Assemble a NAND circuit at the laboratory bench according to the pre-lab task.

3. Be sure that the logic circuit operation of the given function is correct. Compare the observed waveforms of the functions at the output of the circuit and intermediate points with the pre-lab task results.

4. If there is no coincidence with the results, find the errors which were made while assembling ICs or when simplifying BF. Eliminate the errors and succeed in matching practical and ideal diagrams.

5. Increase clock pulse repetition frequency 4–8 times; compare the diagrams of the circuit operation with the earlier ones. If false function value occurs, find the cause and try to dispose of it.

6. Repeat steps 2–5 for NOR circuit.

1.6 QUESTIONS

1. What does ‘irredundant form of the function’ mean?
2. Explain the occurrence of critical races on the example of the circuit which you’ve designed.
3. Draw a truth table for three arguments $X_1X_2X_3$, when the function possesses the value 1 if X_2 equals 1. Give Boolean expression for the function having the value 1.
4. How are the ICs with the open collector output connected to each other?
5. What is ‘Z-state logic gate’? Give an example.
6. How are outputs of Z-state ICs connected? Give an example.
7. What is usually done with unused inputs of TTL circuits at practice?
8. What is the range of logic 1 level of the gates of CMOS and TTL familys?
9. How can the level of logic 1 (of logic 0) be set at the inputs of TTL/CMOS circuits?
10. When is it possible to connect several outputs of logic gates?

Lab 2

MULTIPLEXER STUDY AND CIRCUIT DESIGN ON ITS BASE

2.1 OBJECTIVES

The aim of the lab work is to study the functional power of multiplexers in digital devices, and to reinforce skills of performance control of multiplexers in various operational modes during the research work.

2.2 PRE-LAB TASKS

1. Study functional principle, operation modes and pin configuration of ICs KR1533KP2 (SN74ALS153N) and KR1533KP7 (SN74ALS151N).
2. Study the methods of multiplexer capacity increasing.
3. Make clear the fundamental difference between TTL and CMOS multiplexers.
4. Study the implementation principle of logic functions with the help of multiplexers.
5. Design circuits realizing logic functions from the lab work 1 with the use of one IC KR1533KP2, one IC KR1533KP7, two ICs KR1533KP7.

2.3 BASIC THEORY

Multiplexers are digital multi-positional switches. Multiplexers are able to select a definite path of data transmission. Therefore, they are also called data selectors. Sometimes duplicate name is used: ‘selector/multiplexer’. Multiplexers vary in the number of data and address inputs, availability of enable inputs and inverted outputs, amount of elements in one IC.

Let us consider ICs KR1533KP7and KR1533KP2. You are offered to analyze their functional power in the lab work.

IC KR1533KP7 is a multiplexer which makes possible to carry data from 8 inputs to one output line, which is presented in a noninverting and inverting format. Fig. 2.1 shows the IC logic symbol and pin configuration.

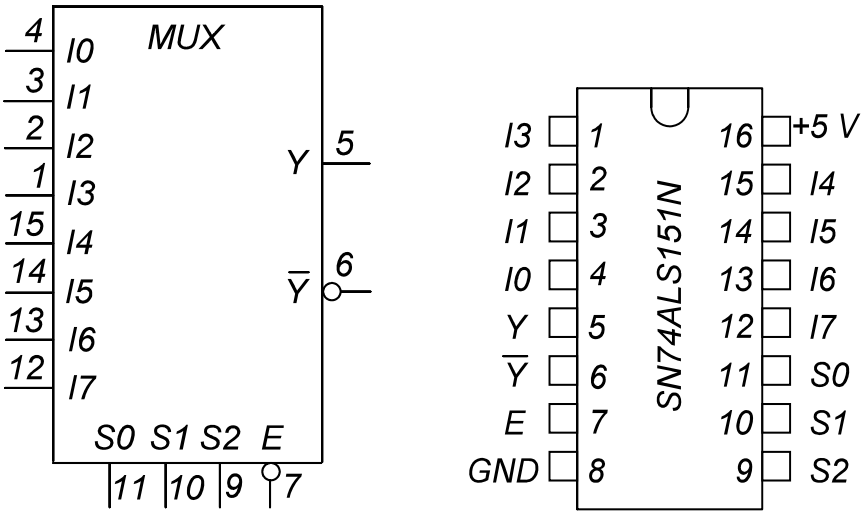


Fig. 2.1. IC KR1533KP7logic symbol and pin configuration

Logic function, carried out by IC KR1533KP7, is given by (in a noninverting output):

$$F = \overline{V}(\overline{A2}\,\overline{A1}\,\overline{A0}D0 + \overline{A2}\,\overline{A1}A0D1 + \overline{A2}A1\overline{A0}D2 + ... + A2A1A0D7).$$

Table 2.1 characterizes IC KR1533KP7 operation principle.

Table 2.1

Multiplexer KR1533KP7 truth table

Inputs				Outputs	
A2	*A1*	*A0*	$\overline{V}$	*F*	$\overline{F}$
x	x	x	1	0	1
0	0	0	0	*D0*	$\overline{D0}$
0	0	1	0	*D1*	$\overline{D1}$
0	1	0	0	*D2*	$\overline{D2}$
0	1	1	0	*D3*	$\overline{D3}$
1	0	0	0	*D4*	$\overline{D4}$
1	0	1	0	*D5*	$\overline{D5}$
1	1	0	0	*D6*	$\overline{D6}$
1	1	1	0	*D7*	$\overline{D7}$

Digit combination at the address inputs (*A2*, *A1*, *A0*) determines from which data input the signals will be transferred to the outputs in the noninverting format (output 5) and which – in the inverting one (output 6). The enable input $\overline{V}$ (input 7) must be in a logic 0 state. Logic 1 at the enable input $\overline{V}$ disables IC operation. In this mode *F* output is set to a logic 0 state

($\overline{F}$ output is set to logic 1 state) at any combination of address and data signals.

Fig. 2.2 shows BF $F = \overline{B}CD + ACD + AB\overline{C} + ABD$ implementation circuit on the basis of IC KR1533KP7 (see also table 2.2). In the circuit the most significant variable A lets DD3 and DD4 multiplexers operate one by one. Only half of the truth table values of the function is embodied at each of them (table 1.4), i.e. $F = F1$ if $A = 0$, $F = F2$ if $A = 1$. Logical summing the variables *F1* и *F2* up we obtain the following resulting function *F*:

$$F = F1 + F2 = \overline{\overline{F1} \times \overline{F2}}$$

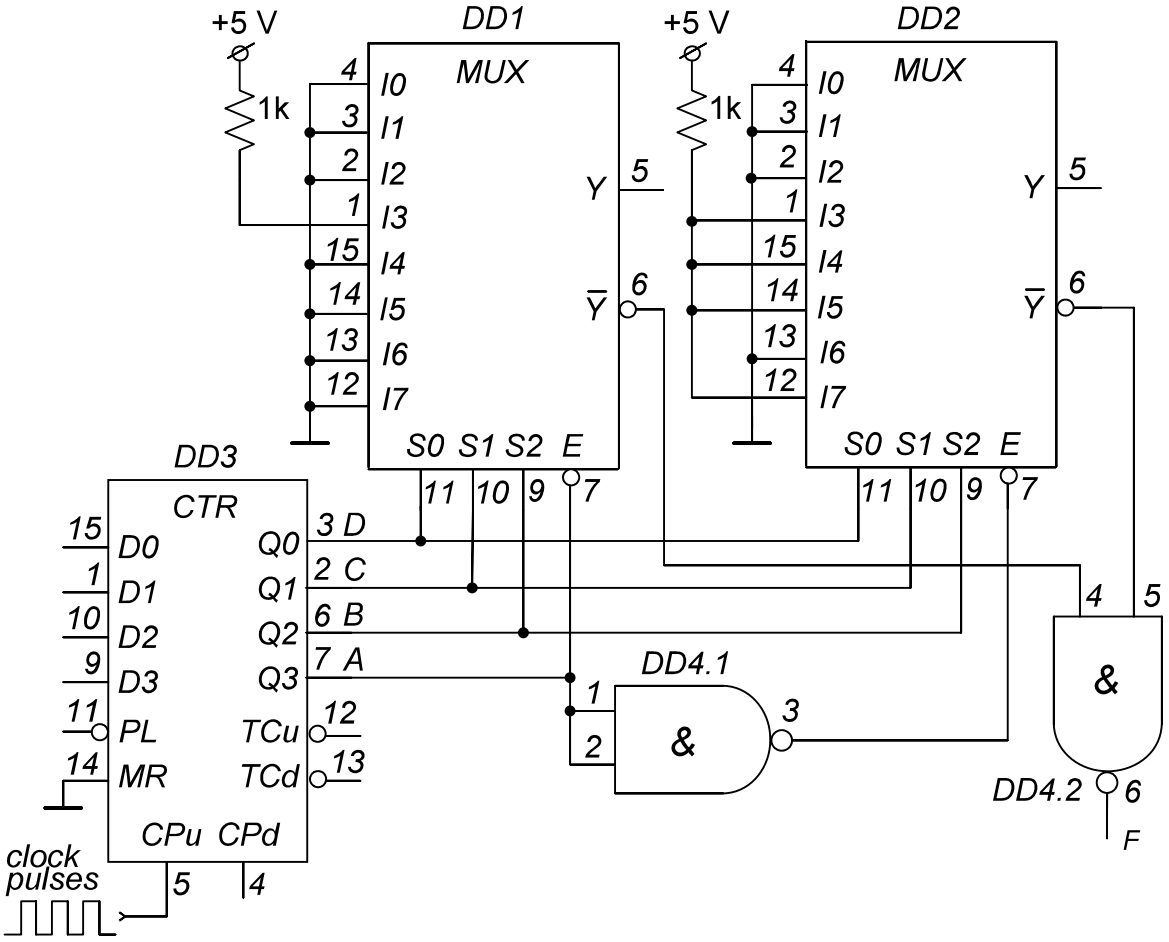


Fig. 2.2. BF implementation on the basis of two ICs KR1533KP7

Table 2.2

ICs for the circuit in Fig. 2.2

IC type Circuitry symbol	KR1533IE7 DD1	KR1533LA3 DD2	KR1533KP7 DD3, DD4
Common	8	7	8
+ 5 V	16	14	16

Fig. 2.3 shows voltage diagrams for the circuit presented in Fig. 2.2. Here we can see when multiplexers DD3 (*F1*) and DD4 (*F2*) operate. As it follows from the diagrams, the *resulting function F* is the same as in the Lab work 1 (Fig. 1.9).

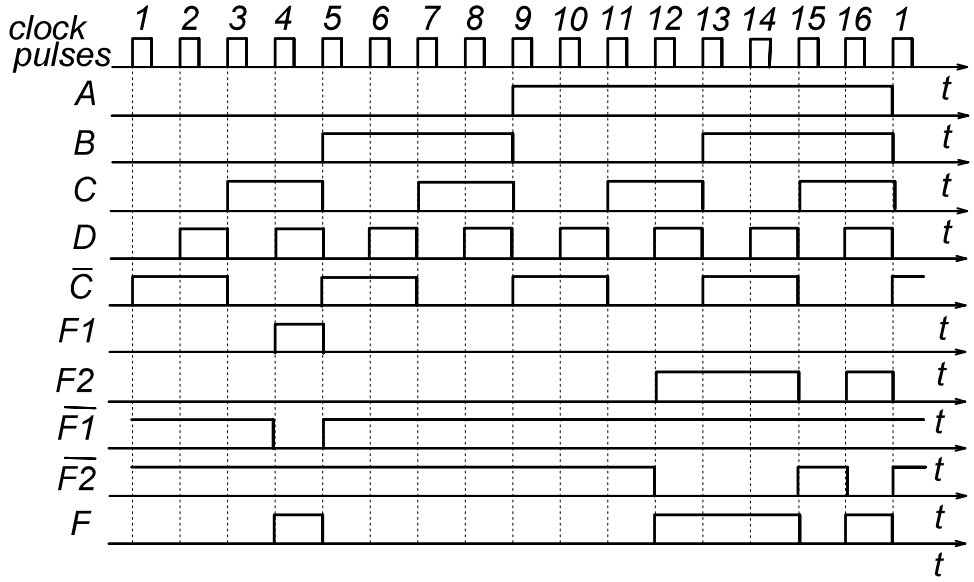


Fig. 2.3. Voltage diagrams for the circuit in Fig. 2.2

The considered example of BF implementation by means of two multiplexers demonstrates the easiest way of multiplexer digit capacity increase. Lab work 3 will be devoted to the more detailed discussion of the methods of data selectors capacity increase.

If we want to implement BF by means of one IC KR1533KP7 or KR1533KP2, the given function *F* should be presented in the table form and linked with one of the four variables: *A*, *B*, *C* or *D*. Tables 2.3–2.6 and Fig. 2.4 shows how the function *F* (the values are taken from the lab work 1, see table 1.4) can be linked with various variables and which signals should be fed to the multiplexer inputs in each case. Table 2.3 indicates how it is possible to link the function *F* and the variable *A*; table 2.4 – variable *B*; table 2.5 – variable *C*; and table 2.6 – variable *D*.

Fig. 2.5 illustrates the BF implementation method, when the output function is linked with one of the input variables. The circuit is carried into operation by using one IC KR1533KP7. Here the signals formed by the counter KR1533IE7 are used as address variables, and the given function is linked with the variable *D* (see table 2.6 and Fig. 2.4, d). The application of the given method allows us to reduce the number of multiplexer address inputs by one, i.e. to use microcircuits with the lower number of data inputs.

Table 2.3

Function F linked with variable A

decimal number	*B*	*C*	*D*	*A*	*F*	
0	0	0	0	**0**	**0**	}**0**
8	0	0	0	**1**	**0**	
1	0	0	1	**0**	**0**	}**0**
9	0	0	1	**1**	**0**	
2	0	1	0	**0**	**0**	}**0**
10	0	1	0	**1**	**0**	
3	0	1	1	**0**	**1**	}**1**
11	0	1	1	**1**	**1**	
4	1	0	0	**0**	**0**	}***A***
12	1	0	0	**1**	**1**	
5	1	0	1	**0**	**0**	}***A***
13	1	0	1	**1**	**1**	
6	1	1	0	**0**	**0**	}**0**
14	1	1	0	**1**	**0**	
7	1	1	1	**0**	**0**	}***A***
15	1	1	1	**1**	**1**	

⇒

decimal number	*B*	*C*	*D*	*A*	*F*
0	0	0	0	**0**	**0**
8	0	0	0	**1**	**0**
1	0	0	1	**0**	**0**
9	0	0	1	**1**	**0**
2	0	1	0	**0**	**0**
10	0	1	0	**1**	**0**
3	0	1	1	**0**	**1**
11	0	1	1	**1**	**1**
4	1	0	0	**0**	***A***
12	1	0	0	**1**	***A***
5	1	0	1	**0**	***A***
13	1	0	1	**1**	***A***
6	1	1	0	**0**	**0**
14	1	1	0	**1**	**0**
7	1	1	1	**0**	***A***
15	1	1	1	**1**	***A***

Table 2.4

Function F linked with variable B

decimal number	*A*	*C*	*D*	*B*	*F*	
0	0	0	0	**0**	**0**	}**0**
4	0	0	0	**1**	**0**	
1	0	0	1	**0**	**0**	}**0**
5	0	0	1	**1**	**0**	
2	0	1	0	**0**	**0**	}**0**
6	0	1	0	**1**	**0**	
3	0	1	1	**0**	**1**	}$\overline{\boldsymbol{B}}$
7	0	1	1	**1**	**0**	
8	1	0	0	**0**	**0**	}***B***
12	1	0	0	**1**	**1**	
9	1	0	1	**0**	**0**	}***B***
13	1	0	1	**1**	**1**	
10	1	1	0	**0**	**0**	}**0**
14	1	1	0	**1**	**0**	
11	1	1	1	**0**	**1**	}**1**
15	1	1	1	**1**	**1**	

⇒

decimal number	*A*	*C*	*D*	*B*	*F*
0	0	0	0	**0**	**0**
4	0	0	0	**1**	**0**
1	0	0	1	**0**	**0**
5	0	0	1	**1**	**0**
2	0	1	0	**0**	**0**
6	0	1	0	**1**	**0**
3	0	1	1	**0**	$\overline{\boldsymbol{B}}$
7	0	1	1	**1**	$\overline{\boldsymbol{B}}$
8	1	0	0	**0**	***B***
12	1	0	0	**1**	***B***
9	1	0	1	**0**	***B***
13	1	0	1	**1**	***B***
10	1	1	0	**0**	**0**
14	1	1	0	**1**	**0**
11	1	1	1	**0**	**1**
15	1	1	1	**1**	**1**

Table 2.5

Function F linked with variable C

decimal number	*A*	*B*	*D*	*C*	*F*	
0	0	0	0	**0**	**0**	}**0**
2	0	0	0	**1**	**0**	
1	0	0	1	**0**	**0**	}***C***
3	0	0	1	**1**	**1**	
4	0	1	0	**0**	**0**	}**0**
6	0	1	0	**1**	**0**	
5	0	1	1	**0**	**0**	}**0**
7	0	1	1	**1**	**0**	
8	1	0	0	**0**	**0**	}**0**
10	1	0	0	**1**	**0**	
9	1	0	1	**0**	**0**	}***C***
11	1	0	1	**1**	**1**	
12	1	1	0	**0**	**1**	}$\overline{\boldsymbol{C}}$
14	1	1	0	**1**	**0**	
13	1	1	1	**0**	**1**	}**1**
15	1	1	1	**1**	**1**	

⇒

decimal number	*A*	*B*	*D*	*C*	*F*
0	0	0	0	**0**	**0**
2	0	0	0	**1**	**0**
1	0	0	1	**0**	***C***
3	0	0	1	**1**	***C***
4	0	1	0	**0**	**0**
6	0	1	0	**1**	**0**
5	0	1	1	**0**	**0**
7	0	1	1	**1**	**0**
8	1	0	0	**0**	**0**
10	1	0	0	**1**	**0**
9	1	0	1	**0**	***C***
11	1	0	1	**1**	***C***
12	1	1	0	**0**	$\overline{\boldsymbol{C}}$
14	1	1	0	**1**	$\overline{\boldsymbol{C}}$
13	1	1	1	**0**	**1**
15	1	1	1	**1**	**1**

Table 2.6

Function F and variable D connection

decimal number	*A*	*B*	*C*	*D*	*F*	
0	0	0	0	**0**	**0**	}***0***
1	0	0	0	**1**	**0**	
2	0	0	1	**0**	**0**	}***D***
3	0	0	1	**1**	**1**	
4	0	1	0	**0**	**0**	}***0***
5	0	1	0	**1**	**0**	
6	0	1	1	**0**	**0**	}***0***
7	0	1	1	**1**	**0**	
8	1	0	0	**0**	**0**	}***0***
9	1	0	0	**1**	**0**	
10	1	0	1	**0**	**0**	}***D***
11	1	0	1	**1**	**1**	
12	1	1	0	**0**	**1**	}***1***
13	1	1	0	**1**	**1**	
14	1	1	1	**0**	**0**	}***D***
15	1	1	1	**1**	**1**	

⇒

decimal number	*A*	*B*	*C*	*D*	*F*
0	0	0	0	**0**	***0***
1	0	0	0	**1**	***0***
2	0	0	1	**0**	***D***
3	0	0	1	**1**	***D***
4	0	1	0	**0**	***0***
5	0	1	0	**1**	***0***
6	0	1	1	**0**	***0***
7	0	1	1	**1**	***0***
8	1	0	0	**0**	***0***
9	1	0	0	**1**	***0***
10	1	0	1	**0**	***D***
11	1	0	1	**1**	***D***
12	1	1	0	**0**	***1***
13	1	1	0	**1**	***1***
14	1	1	1	**0**	***D***
15	1	1	1	**1**	***D***

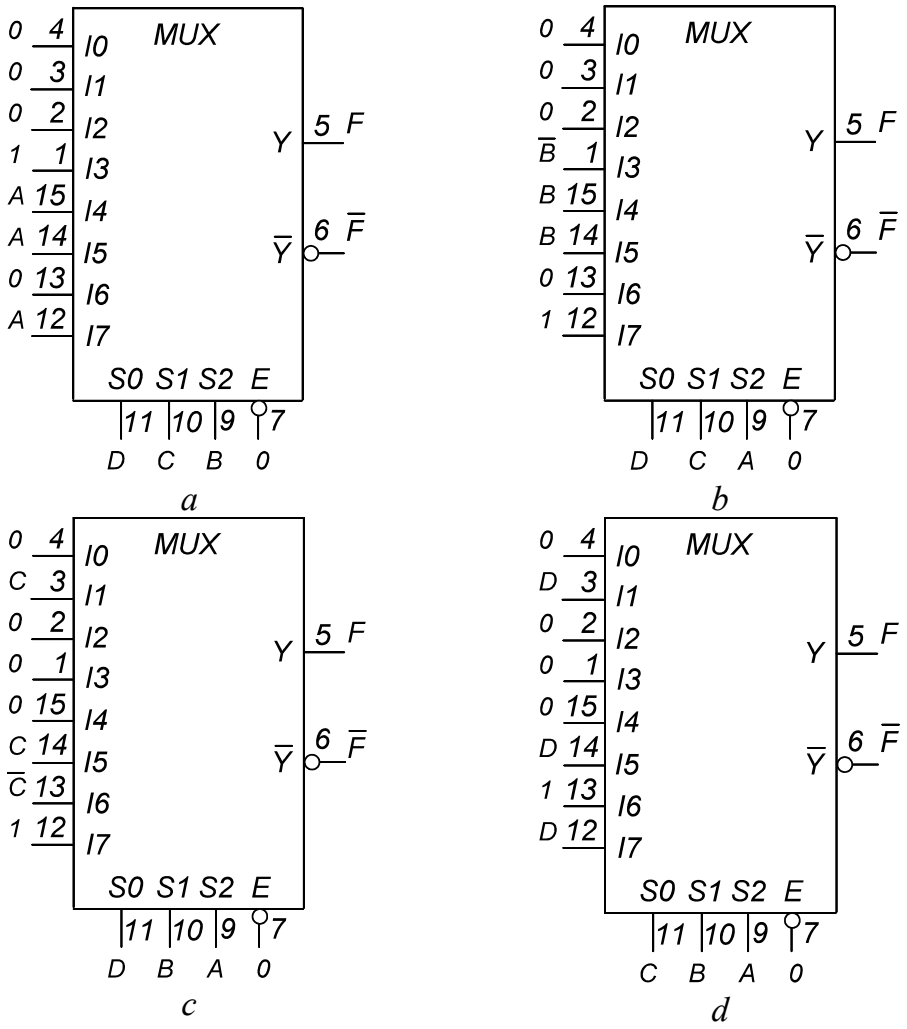


Fig. 2.4. BF implementation via linking with the variables A (a), B (b), C (c) and D (d), according to the table 2.3–2.6

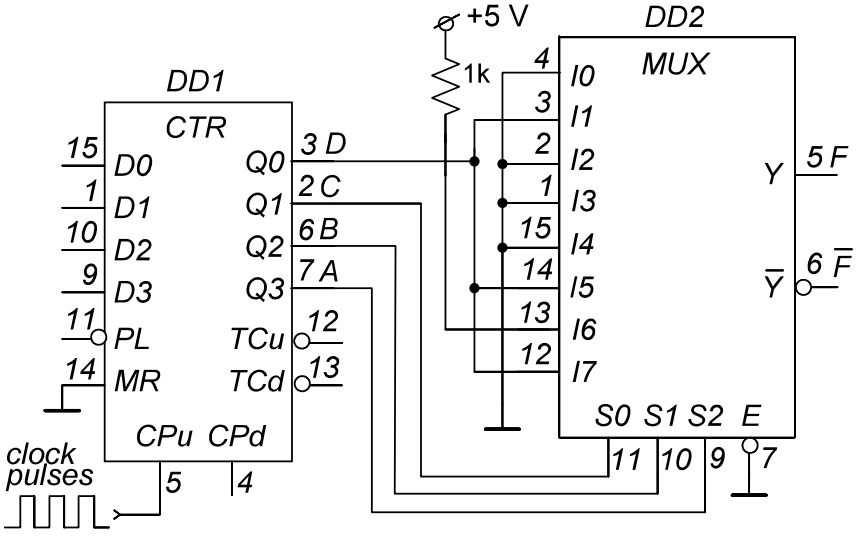


Fig. 2.5. BF implementation by using one IC KR1533KP7 (DD1 – KR1533IE7, DD2 – KR1533KP7)

IC KR1533KP2 is a two 4-input multiplexers, having two address inputs *A1* and *A0*, which are common for both multiplexers; $\overline{V1}$ и $\overline{V2}$ – enable inputs (active level is low). Two independent outputs transmit the signal levels presented at the multiplexers data inputs selected with the help of the address. Fig. 2.6 shows the IC KR1533KP2 logic symbol and pin configuration.

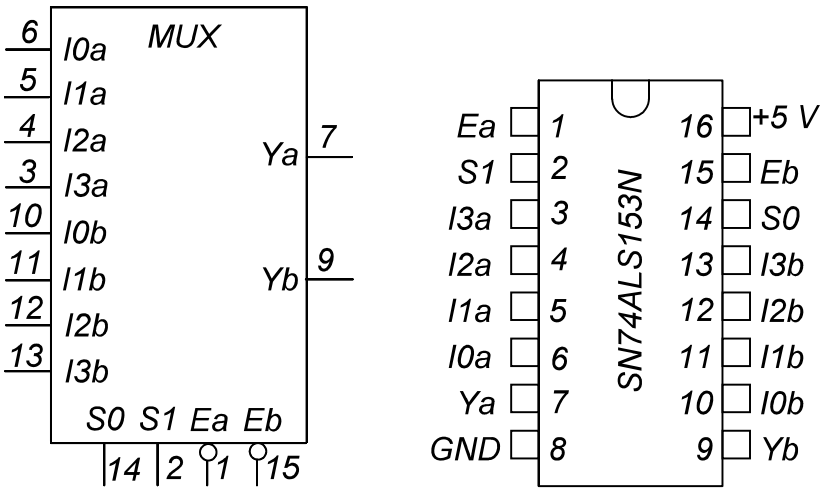


Fig. 2.6. IC KR1533KP2 logic symbol and pin configuration

The logic function realised by each part of the IC KR1533KP2 has the following form:

$$F = \overline{V}(\overline{A1}\,\overline{A0}D0 + \overline{A1}A0D1 + A1\overline{A0}D2 + A1A0D3).$$

The successful operation of the section occurs when the potential of the enable input $\overline{V}$ ($\overline{V1}$ or $\overline{V2}$) is equal to 0.

When $\overline{V1} = 1$ or $\overline{V2} = 1$, the blocking of the corresponded multiplexer section takes place, and logic 0 rate is set at its output despite the data inputs *D0–D3* state. The operation of the IC KR1533KP2 is presented in table 2.7.

Table 2.7

Truth table for KR1533KP2 multiplexer

Inputs							Output
$\overline{V}$	*A1*	*A0*	*D0*	*D1*	*D2*	*D3*	*F*
0	0	0	1/0	x	x	x	*D0*
0	0	1	x	1/0	x	x	*D1*
0	1	0	x	x	1/0	x	*D2*
0	1	1	x	x	x	1/0	*D3*
1	x	x	x	x	x	x	0

Fig. 2.7 realises the BF $F = \overline{B}CD + ACD + AB\overline{C} + ABD$ implementation circuit on the basis of IC KR1533KP2 (see also table 2.8), where the prescribed function is linked with the variable *D*. As each part of the multiplexer has only 4 inputs, it is required to increase capacity twice. The signals *C* и $\overline{C}$ are employed as the signals permitting multiplexer functioning. Thus, to the circuit output the signal takes turns in coming from the multiplexer first and second input, i.e. firstly, from *1D0*, then *2D0*, *1D1*, *2D1*, etc. Fig. 2.8 depicts voltage diagrams in the circuit scheme 2.7 and shows the order of signals selection.

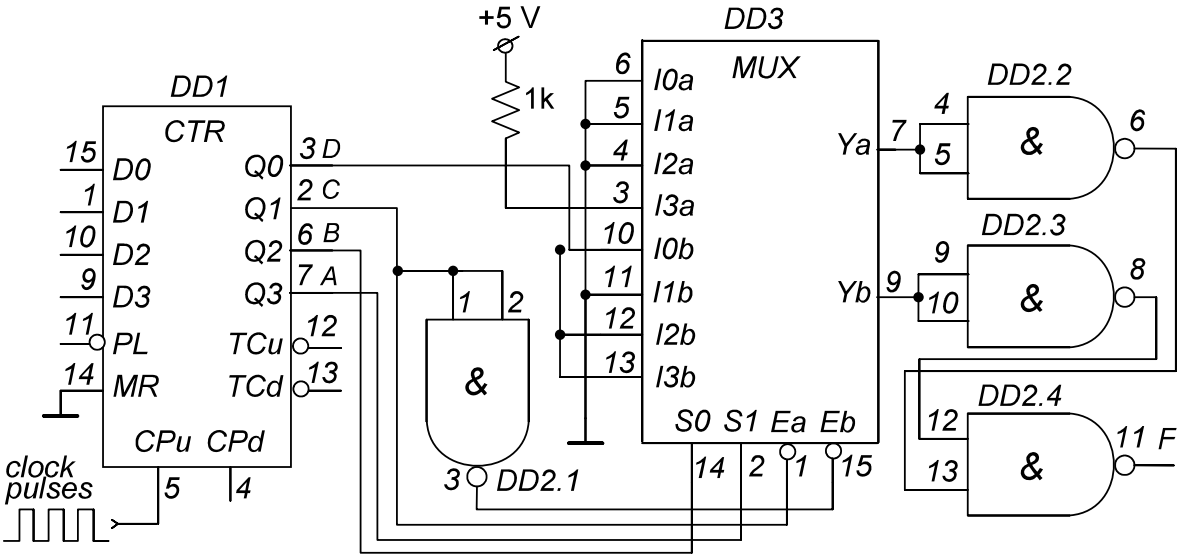


Fig. 2.7. BF implementation by means of IC KR1533KP2

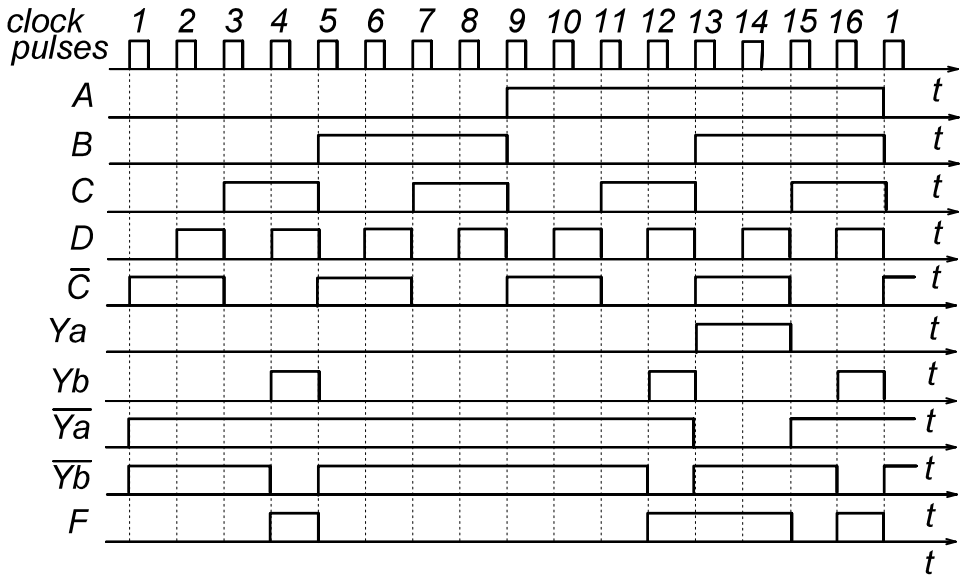


Fig. 2.8. Voltage diagram for the circuit in Fig. 2.7

As you can see from the diagrams (Fig. 2.8), function *F* is the same as presented in Fig. 2.3 and in the lab work 1 (Fig. 1.9). At the same time the number and mode of intermediate signals is defined by the selected method of the function synthesis.

Table 2.8

ICs in the circuit in Fig. 2.7

IC type Circuitry symbol	KR1533LA3 DD2	KR1533IE7 DD1	KR1533KP2 DD3
Common	7	8	8
+5 V	14	16	16

2.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4.

In order to carry out the lab work you'll need a set of microcircuits: KR1533LA3, KR1533KP2, KR1533KP7, KR1533IE7 (to form the variables *A*, *B*, *C* and *D*).

2.5 IN-LAB TASKS

1. Check the operating principle of the multiplexer in the steady-state mode. To do this you'll need to provide a random combination of input data to the data and address inputs.
2. Study the functioning of a multiplexer as a converter of a parallel binary code to the serial one. In order to do this you should set the given code to the multiplexer KR1533KP7 data input according to the lecturer's task. The address inputs should be supplied with the counter KR1533IE7 code bit-by-bit. The counter should operate in accumulating mode (*+1*). Take the waveforms of voltages at the multiplexer inputs and output. Make sure that the code is converted.
3. On the base of one or two KR1533KP7 multiplexers implement the function of 4 variables, which is prescribed in lab work 1 and fulfilled by means of simple logic gates. Be sure that the multiplexer acts as a logical function generator.
4. Implement the function of 4 variables using IC KR1533KP2. Take the waveforms of voltages at the address inputs, enable inputs, multiplexer outputs and whole circuit output.
5. Compare the waveforms from sections 3 и 4 with the results of the lab work 1. Draw the conclusion.

2.6 QUESTIONS

1. Why multiplexers are sometimes called selectors? Give the definition of the term ‘multiplexer’.
2. What are the differences between TTL and CMOS multiplexers? Do they differ in operating principle or functionality?
3. What are the differences of TTL family multiplexers?
4. What is the output function in the circuit presented in Fig. 2.5 if there is a failure of the pin $\overline{V1}$ (or $\overline{V2}$)?
5. What does ‘Z-state of a multiplexer’ mean?
6. Where can multiplexers are applied?
7. Explain the multiplexer operating principle.
8. Give the definition of the term ‘demultiplexer’. Describe its operating principle.
9. What are the differences between TTL and CMOS demultiplexers?
10. How is it possible to perform the conversion from a parallel binary code to the sequential one?

Lab 3

MULTIPLEXER AND DECODER CAPACITY INCREASE. SEVEN-SEGMENT DECODERS STUDY

3.1 OBJECTIVES

The aim of the lab work is to revise the main principles of access devices capacity increase on the basis of multiplexers and decoders. The peculiarities of seven-segment decoder, which is widely used in data display systems, are also studied here.

3.2 PRE-TASKS

1. Study the functional principle, operation modes and pin configuration of the ICs KR1533KP2 (SN74ALS153N), KR1554ID14 (74AC139N) and 533ID18 (SN54LS247), 555ID18 (74LS247N).

2. Study the methods of decoders and multiplexers capacity increase.

3. Get to know the connection circuit of the 7-segment display and the functions described the states of the 7-segment decoder outputs. Write the functions of 4 variables for any three segments.

3.3 BASIC THEORY

The functional principle, logic symbol and the purpose of the IC outputs of the multiplexers (particularly, KR1533KP2 and KR1533KP7) are considered in details in paragraph 2.3. Let us now examine the functional principle of another access device – a decoder.

Decoders (DC – decoder) are the devices used to convert binary input code to the positional output one. At any given time only one of *n*-outputs is turned on. The selection is performed by means of control signals coming to the proper inputs.

IC KR1554ID14 consists of a pair of two-input decoders. Each of them possesses an enable input and its active state is defined by the low level of the voltage ‘0’. Decoder’s outputs are inverted, i.e. active level also equals ‘0’. Fig. 3.1 shows the IC KR1554ID14 logic symbol and its pin configuration.

Logic function accomplished by the IC KR1554ID14 is:

$$F0 = X1 + X0 + E0,$$

therefore,

$$F1 = X1 + \overline{X0} + E0$$

etc.

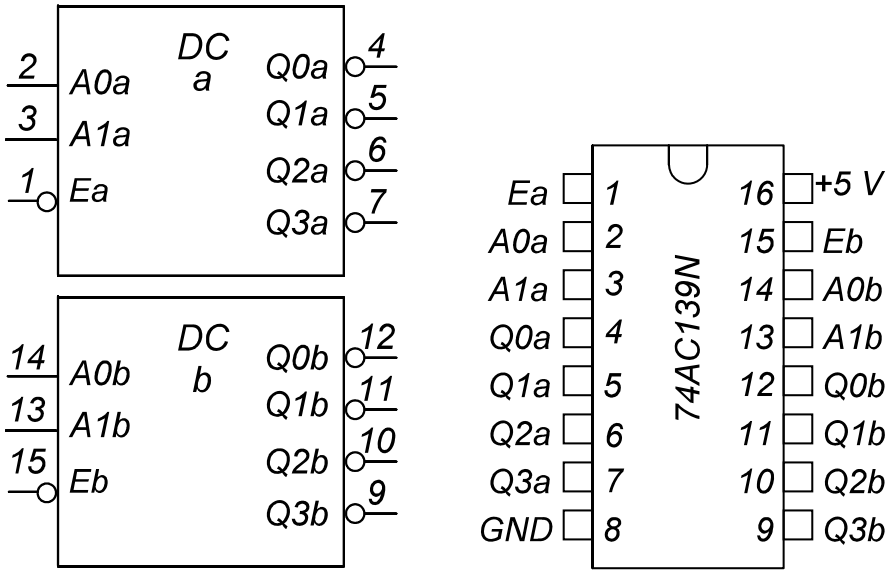


Fig. 3.1. IC KR1554ID14 logic symbol and its pin configuration

Table 3.1 represents the operation of one section of the IC KR1554ID14.

Table 3.1

Truth table for KR1554ID14

Input code			Output state			
E0	*X1*	*X0*	*F0*	*F1*	*F2*	*F3*
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

In the lab work the decoder under discussion is going to be used for multiplexer capacity growth. The capacity increase is applied when it is necessary to switch more lines than the given multiplexer allows. The following main methods of multiplexer capacity increase are known.

Cascade multiplexer connection (series connection)

Fig. 3.2 shows the implementation of cascade multiplexer connection. Cascading is performed using 4-input multiplexers.

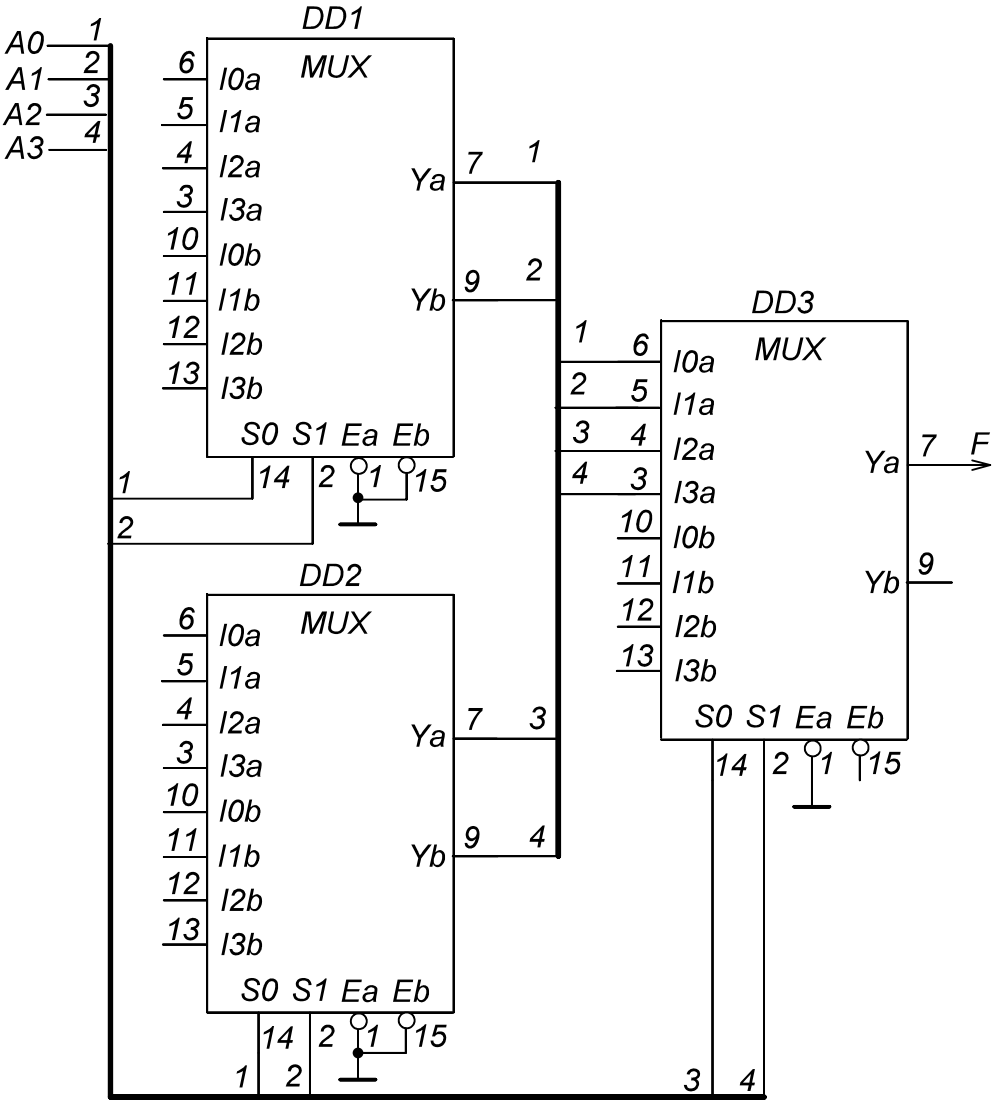


Fig. 3.2. Cascade multiplexer connection (4-to-1)
(DD1, DD2, DD3 – KR1533KP2)

In order to select one out of 16 data inputs it's necessary to have 4 address input lines: *A3*, *A2*, *A1* and *A0*. Four basic multiplexers '4-to-1' assure the select according to the code provided to *A1* and *A0*, respectively: the first – one of *D0*, *D1*, *D2* or *D3* signals, the second – *D4*, *D5*, *D6* or *D7* signals, the third – *D8*, *D9*, *D10* or *D11* signals, the fourth –*D12*, *D13*, *D14* or *D15* signals. The fifth multiplexer guarantees the select of one of the formerly selected signals (from the output of the four multiplexers of the first cascade) depending on the code supplied to *A3* и *A2*. The active input level (in case KR1533KP2 is low) must be supplied to the enable inputs $\overline{V}$ of all multiplexers. In case of such connection multiplexers without enable input can be used.

Let's define which signal *A3A2A1A0* must be provided at the address inputs in order to transmit the signal *D10*, coming to the input *1D2* of the multiplexer DD2, to the circuit output *F1*. The input *1D2* corresponds to 2_{10}, therefore, the signal 10_2 must be supplied to the address inputs *A0A1*. In the first cascade of DD1, DD2 multiplexers the signals *D2*, *D6*, *D10*, and *D14* will be selected. According to the diagram (Fig. 3.2), DD3 multiplexer must connect pin 4 (input *1D2*) to the output, where the signal comes from the first section output of DD2 multiplexer. *1D2* input code of DD3 multiplexer equals 2_{10}, thus, the signal 10_2 must be supplied to the address inputs *A2A3*. So, if we want *D10* to transmit to the output of the whole circuit, the code 1010_2 must occur at address lines *A3A2A1A0*.

Parallel multiplexer connection

Parallel multiplexer connection is realised with the help of a decoder. An illustration of this implementation is shown in Fig. 3.3 (see also table 3.2). In this case the decoder KR1554ID14 is used. To make it work we should supply logic zero to the enable input $\overline{E0}$. To select one multiplexer data input out of 16, 4 input address lines: *A3*, *A2*, *A1* и *A0* are required. The most significant bits of the addresses *A3* и *A2* are supplied to the decoder data inputs. The corresponding DC outputs are connected to the enable inputs $\overline{V}$ of the multiplexers (DD2, DD3). DC selects which multiplexer – DD2 or DD3 – will operate. Regardless of the variable *D* value there will be '0' at the outputs of other multiplexers. In accordance with the code supplied to *A1* и *A0* the active multiplexer selects respectively: the first – one of *D0*, *D1*, *D2* or *D3* signals, the second – *D4*, *D5*, *D6* or *D7* signals, the third – *D8*, *D9*, *D10* or *D11* signals, the fourth – *D12*, *D13*, *D14* or *D15* signals. The signals received from the multiplexer outputs are summed up.

For example, if we supply *A3A2A1A0* = 0010_2 = 2_{10} to the DC data inputs, we'll obtain the code *A3A2*=00_2 =0_{10}, therefore, only the first multiplexer will be allowed to operate. There will be logic 0, i.e. *F2=F3=F4*=0 at the rest of multiplexers outputs. If a binary code *A1A0* = 10_2 = 2_{10} is delivered to all multiplexers address inputs, then the information from *1D2* input (i.e. the value of the variable *D2*) will be transferred to the *F1* DD2 output. After being summed, *F=F1+F2+F3+F4* = 0 + 0 + 0 + *D2* = *D2* will be at the output of the whole circuit.

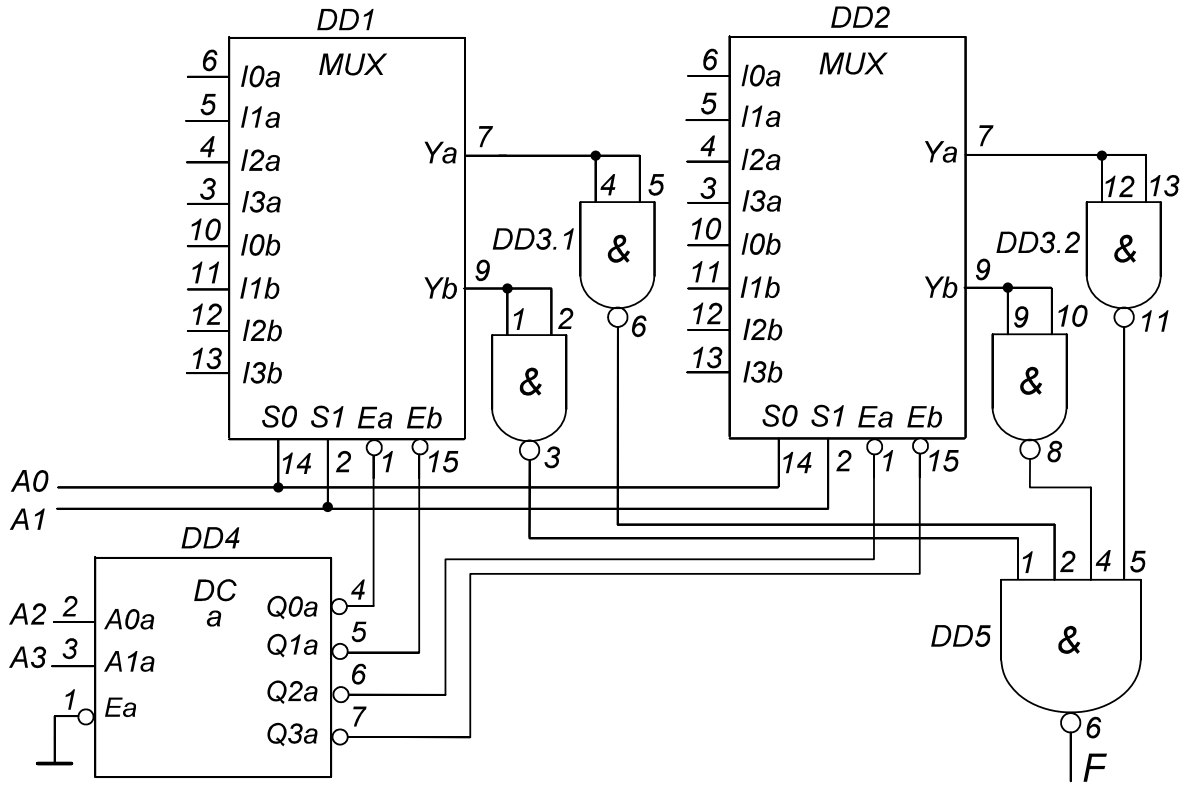


Fig. 3.3. The circuit of multiplexer capacity increase with the help of a decoder

Table 3.2

IC in the circuit in Fig. 3.3

IC type Circuitry symbol	KR1533KP2 DD1, DD2	KR1554ID14 DD4	KR1533LA3 DD3	KR1533LA1 DD5
Common	8	8	7	7
+5 V	16	16	14	14

Decoder capacity increase

It is often necessary to provide an access to devices which amount exceeds the digit capacity of given decoder. The enable input allows us to increase the decoder capacity. Fig. 3.4 shows the simplest scheme of decoder capacity increase up to 8 outputs on the basic of dual 4-output decoder KR1554ID14 and an inverter. The least significant bits *X1X0* of the *X2X1X0* input code are supplied to the similar data inputs of both decoders, high-order bit *X2* – to the enable input. Moreover, one of the decoders is supplied with inverted *X2* signal. Depending on the variable *X2* value, only one section of KR1554ID14 will function. At the same time, there are inactive signals at all the outputs of the second section. The operation of the decoder capacity increase circuit is presented in table 3.4.

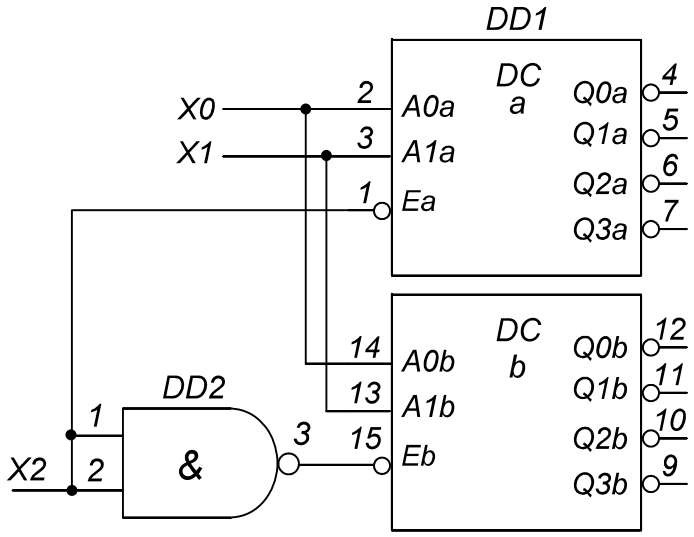


Fig. 3.4. Decoder capacity increase circuit on the IC KR1554ID14 basis (DD1 – KR1554ID14, DD2 – KR1533LA3)

Table 3.4

Decoder capacity increase (Fig. 3.4)

Inputs			Outputs							
X2	*X1*	*X0*	*F0*	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Decoder capacity can be enhanced by analogy with the parallel method of multiplexer capacity increase (Fig. 3.3). For example, if it is necessary to implement a 16-output decoder, you'll need five 4-output decoders with enable inputs. In this case the first decoder – DC1 (high-order bits of the input code are supplied to its inputs) selects which decoder DC2–DC5 will function. Therefore, the signals from DC1 outputs are connected to DC2–DC5 enable inputs, and data inputs of the same name *D0* and *D1* are joined together and connected to the least significant bits of the input code.

Seven-segment decoder study

7-segment decoders are applied for converting binary code to the 7-segment one. 7-segment code is essential for displaying figures from 0 to 9 at a digital indicator. The code is called 7-segment because the figures are

depicted in the form of the so called ‘segments’ (Fig. 3.5, *a*). The display often has one more additional segment – a dot. Actually, 7-segment decoders are the elements with the open collector (inverted output) or an open emitter (non- inverted output). Along with such decoders the displays with common anodes or cathodes are used.

In the lab work the 7-segment decoder 533ID18 (SN54LS247) (555ID18 (74LS247N) is also possible) with open collector outputs is used. Fig. 3.5 shows the IC 533ID18 logic symbol and pin configuration.

IC 533ID18 is a decoder-driver for converting binary-decimal code to the 7-segment one. It is designed for operating with the common anode indicator, which is connected to the outputs *a–g* (active low level). The inputs *X0–X3* are supplied with a number in a binary code. Being excited by a low-level signal, the lighting test input (*LT*) makes the outputs *a – g* active. When the same excitation signal is supplied to the blanking input (*B1*), high level appears at all outputs, and causes the previous test values of the indicator to be reset. Successive blanking input (*RB1*) is intended for IC 533ID18 operation composed of multidigit display. When *RB1* input is excited by a low-level signal, the indicator blanking takes place only if zero is currently displayed. When the input *RB1* becomes active, for some time the output *B1/RB0* becomes the output of the successive blanking signal (*RB0*) and the high-level signal is changed by the low-level signal. It should be noted here that ‘blanking’ means emission cease of all indicator LEDs.

Fig. 3.6 presents the connection circuit of the 7-segment display with a common anode to the decoder 533ID18. The display SA04-12 from ‘Kingbright’ is used in our lab work.

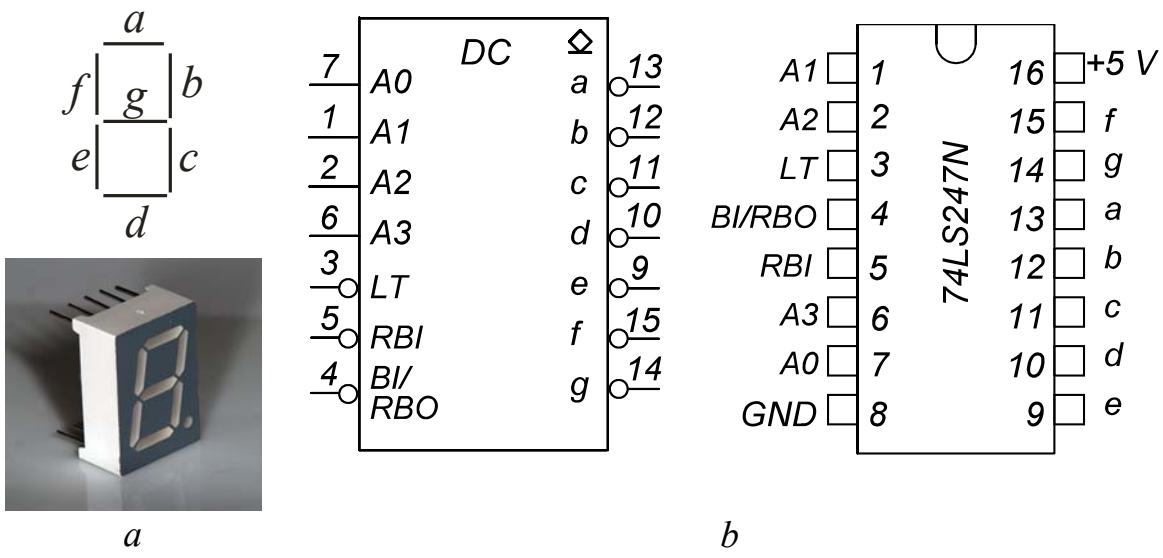


Fig. 3.5. Exterior view of a seven-segment display (a), the IC 533ID18 logic symbol and pin configuration (b)

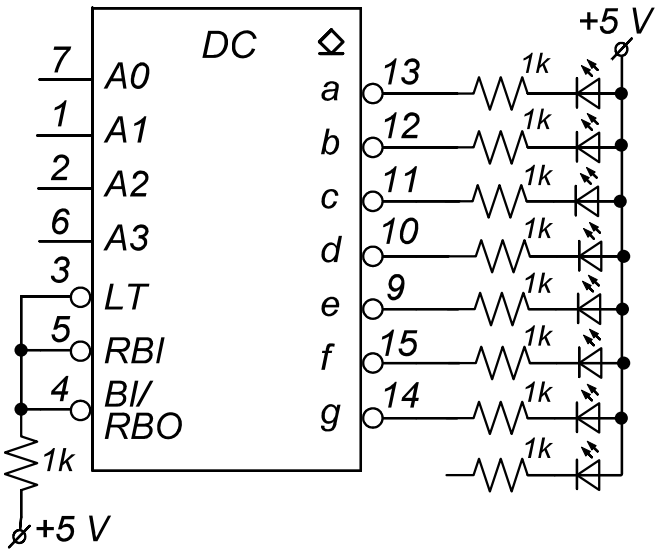


Fig. 3.6. Connection circuit of the 7-segment display to the decoder 533ID18 (555ID18)

Table 3.5

Truth table for the decoder 533ID18

Decimal numbers/ functionality	Input code						*B1/ RB0*	Outputs						
dec	*LT*	*RB1*	*X3*	*X2*	*X1*	*X0*		*a*	*b*	*c*	*d*	*e*	*f*	*G*
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	x	0	0	0	1	1	1	0	0	1	1	1	1
2	1	x	0	0	1	0	1	0	0	1	0	0	1	0
3	1	x	0	0	1	1	1	0	0	0	0	1	1	0
4	1	x	0	1	0	0	1	1	0	0	1	1	0	0
5	1	x	0	1	0	1	1	0	1	0	0	1	0	0
6	1	x	0	1	1	0	1	1	1	0	0	0	0	0
7	1	x	0	1	1	1	1	0	0	0	1	1	1	1
8	1	x	1	0	0	0	1	0	0	0	0	0	0	0
9	1	x	1	0	0	1	1	0	0	0	0	1	0	0
10	1	x	1	0	1	0	1	1	1	1	0	0	1	0
11	1	x	1	0	1	1	1	1	1	0	0	1	1	0
12	1	x	1	1	0	0	1	1	0	1	1	1	0	0
13	1	x	1	1	0	1	1	0	1	1	0	1	0	0
14	1	x	1	1	1	0	1	1	1	1	0	0	0	0
15	1	x	1	1	1	1	1	1	1	1	1	1	1	1
B1	x	x	x	x	x	x	0	1	1	1	1	1	1	1
RB1	1	0	0	0	0	0	0	1	1	1	1	1	1	1
LT	0	x	x	x	x	x	1	0	0	0	0	0	0	0

Table 3.5 represents the operation of the decoder. ***Notice:***

- For lines 0–15: when it is desirable to display numbers from 0 to 15, the blanking input *B1* must be disconnected or have high logic level signal. The successive blanking input *RB1* must be disconnected or have high logic level, if decimal zero suppression is unwanted.
- For line *B1*: when low-level voltage comes directly to the blanking input *B1*, all segment outputs are turned off without regard to the level of any other input.
- For line *RB1*: when the voltage at the input *RB1* and the voltage at the inputs *X0–X3* bear low level, and the voltage is high at the *LT*, all segment outputs are turned off and the level *RB0* becomes low (trigger condition).
- For line *LT*: when the blanking input/output of the successive blanking (*B1/RB0*) is disconnected or bears high voltage level, and the input *LT* is provided with low voltage, all segment outputs are turned on.
- Symbol ‘x’ means non-operating inputs.

It should be noted that the signals at the input corresponded to the decimal numbers 10, 11, 12, 13, 14 and 15 make specific signals appear at the decoder 533ID18 outputs. These signals relate to the incomplete digital images at the display.

3.4 EQUIPMENT

In the lab work the ICs of the multiplexer KR1533KP2 and decoders KR1554ID14 and 533ID18 are studied. The laboratory bench has a built-in 7-segment display with pull-down resistors. The wires connection points are marked with the letters *a* – *g* for segments and *h* – point. The implementation of the required connections of the ICs is performed at the pinboard of the module UIK-1 with the assistance of ordinary equipment: an oscilloscope, and a personal computer.

3.5 IN-LAB TASKS

1. Increase multiplexer KR1533KP2 capacity by cascade connection (Fig. 3.2) and check if the circuit functions correctly by setting up the address given by the lecturer and by supplying a random pulse sequence to the proper input of one of the multiplexers.
2. Implement a logic function of 4 variables. The function which was discussed earlier in the lab works 1 and 2 is used. As a source of logic variables the counter KR1533IE7, which operates in the accumulating mode (*+1*), is employed. Take the waveforms of signals at the multiplexer output and address inputs.

3. Increase multiplexer KR1533KP2 capacity by parallel connection (Fig. 3.3) and check if the circuit functions correctly by setting up the address given by the lecturer and by supplying a random pulse sequence to the proper input of one of the multiplexers.

4. Implement a logic function of 4 variables by means of parallel connection. Compare the results achieved in paragraph 2 and previous lab works. Make a conclusion.

5. Verify the operating principle of the decoder KR1554ID14 in a static mode. To fulfil this task you should supply the code (given by the lecturer) to the data inputs.

6. Implement decoder capacity increase up to 8 using the circuit in Fig. 3.4. Check the circuit performance by supplying alternating binary code as *X3X2X1* signals. The code is formed by means of the counter KR1533IE7.

7. Display the given number at the built-in 7-segment display by connecting the proper inputs to the power supply or zero potential.

8. Connect the decoder 533ID18 to the 7-segment display. According to the lecturer's task set the given number on the display (Fig. 3.6).

9. Study how the decoder functions if the input code is more than 9. Make a conclusion.

3.6. QUESTIONS

1. Design a multiplexer 16-to-1 on the basis of the multiplexer KR1533KP7. Present a circuit in the lab work report.
2. Design a multiplexer 16-to-1 on the basis of the multiplexer KR1533KP2. Present a circuit in the lab work report.
3. Think of case studies of multiplexing. What is it applied for?
4. Give a definition of 'a decoder'. Describe its operating principle, and show the truth table. Speak about the application of decoders.
5. Is it possible to increase the decoder capacity if the decoders without enable input are used? How?
6. Describe a 7-segment display, its types and circuits.
7. Explain what a 7-segment code is and how is it formed?
8. Implement 8-output decoder capacity increase in order to obtain a 16-output decoder.
9. How is it possible to implement a 4-output demultiplexer on the basis of DC KR1554ID14.
10. Design a circuit of 8-bit demultiplexer on the basis of DC KR1554ID14 and basic logic gates.

Lab 4

CIRCUIT DESIGN AND STUDY OF HALF-ADDER AND FULL-ADDER ON THE BASIS OF MULTIPLEXERS AND LOGIC GATES

4.1 OBJECTIVES

The lab work acquaints students with the execution of arithmetic operations on binary numbers, on the one hand, and the implementation of elementary arithmetic units using multiplexers and simple logic gates, on the other hand.

4.2 PRE-TASKS

1. In terms of one-digit binary numbers addition be able to make up truth tables for a half-adder and a half-subtracter. Write down the equations.
2. Draw a half-adder circuit in NAND basis.
3. Study the half-adder/half-subtracter operating principle (Fig. 4.2) and logic structure.
4. Draw the diagrams of a half-adder and a half-subtracter operation.
5. Study the operating principle of one-digit and two-digit adders.
6. On the base of the table 4.10 construct the circuit for the one-digit subtracter using the IC KR1533KP2.

4.3 BASIC THEORY

In the lab work elementary arithmetic units are analysed. Their peculiarity lies in the fact that the signals are attributed arithmetic values 0 and 1, but not logic ones, and the operations made on them follow the binary arithmetic laws. The lab work is devoted to the design and study of the half-adder circuits based on the simple logic IC KR1533LA3, KR1533LP5, one-digit and two-digit adders and one-digit subtracter on the basis of IC KR1533KP2 multiplexer.

Adders are the electronic assemblies performing binary numbers addition. The adders are divided into two groups according to the mode of operation: combinational – having no memory elements and storage adders – those which store the results of previous calculations. Depending on the number processing technique each adder in its turn can be classified as series or parallel. Both of them are built on the basis of single-digit adding circuits. Therefore, in the lab work the fundamentals of the elementary half-adder circuit, single-digit full-adder and full-subtracter design are considered.

The addition of numbers in series adders is fulfilled bit-by-bit, step-by-step. In parallel adders all digits of multidigited numbers are added simultaneously.

The simplest adding element is a half-adder shown in Fig. 4.1. There are two inputs *A* and *B* for the summands and two outputs: *S* (sum) и *C* (carry) here. The half-adder can be used only if we need to add two single-digit numbers. Table 4.1 presents the combination logic device operating principle.

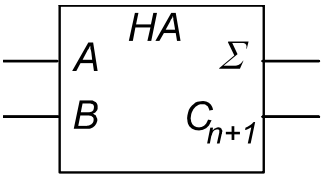


Fig. 4.1. Half-adder

Table 4.1

Truth table for a half-adder

A	*B*	*S*	*C*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

The sum *S* and carry *C* output functions can be defined in the following way:

$$S = \overline{A}B + A\overline{B},$$
$$C = AB.$$

The implementation of the universal half-adder/half-subtracter based on the logic gates KR1533LP5 (SN74ALS86N) – 'EXCLUSIVE OR', is shown in Fig. 4.2 (see also table 4.2). When summed up the outputs correspond to *Fs* и *Fc* (sum and carry, respectively), when subtracted – *Fd* и *Fb* (difference and borrow). Table 4.3 presents the circuit functioning. A pair of adjacent variables from the counter output is used as input variables.

Table 4.2

ICs of the circuit in Fig. 4.2

IC type Circuitry symbol	KR1533IE7 DD1	KR1533LP5 DD2	KR1533LA3 DD3
Common	8	7	7
+5 V	16	14	14

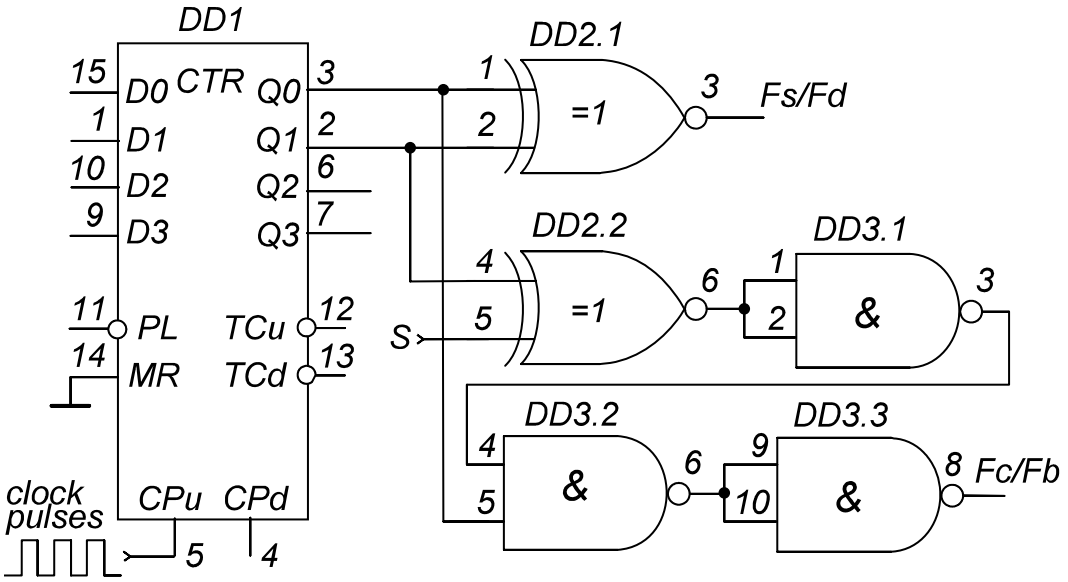


Fig. 4.2. Half-adder/half-subtracter circuit

Table 4.3

Truth table for the circuit in Fig. 4.2

S	Operation	*X1*	*X0*	*Fs/Fd*	*Fc/Fb*
0	} subtraction	0	0	0	0
0		0	1	1	1
0		1	0	1	0
0		1	1	0	0
1	} addition	0	0	0	0
1		0	1	1	0
1		1	0	1	0
1		1	1	0	1

Single-digit full-adder is shown in Fig. 4.3. There are three inputs: A and B for the summands and C_0 for the carry from the lower bit addition, and two outputs: S (sum) and C (carry). The sum S and carry C output functions can be defined in the following manner:

$$S = \overline{A}\overline{B}C_0 + \overline{A}B\overline{C}_0 + A\overline{B}\overline{C}_0 + ABP_0 = (A \oplus B)\overline{C}_0 + \overline{(A \oplus B)}C_0,$$

$$P = \overline{A}BC_0 + A\overline{B}C_0 + AB\overline{C}_0 + ABC_0 = AB + (A \oplus B)C_0.$$

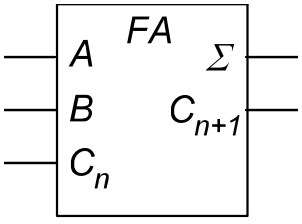


Fig. 4.3. Full-adder

Table 4.4 describes full single-digit adder functioning.

Table 4.4

Truth table for a full-adder

A	B	C_0	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

In the lab work it is required to implement a single-digit adder circuit on the basis of the multiplexer KR1533KP2. To perform the task you need 'to bind' the sum S and carry C functions with one of the IC KR1533IE7 output variables B, C or D. Table 4.5 illustrates full single-digit adder output function obtained by binding to the variable D, where S – the sum of the variables B, C и D=C_0, and C_1 – carry to the next digit by adding variables B, C and C_0.

Table 4.5

Binding between the full-adder output functions and the variable D

B	C	D	C_0	S	C_1	Data inputs KR1533KP2 (S)		Data inputs KR1533KP2 (C_1)	
0	0	0	0	0	0	*2D0*	D	*1D0*	0
0	0	1	1	1	0				
0	1	0	0	1	0	*2D1*	$\overline{D}$	*1D1*	D
0	1	1	1	0	1				
1	0	0	0	1	0	*2D2*	$\overline{D}$	*1D2*	D
1	0	1	1	0	1				
1	1	0	0	0	1	*2D3*	D	*1D3*	1
1	1	1	1	1	1				

Table 4.5 also shows what signals should be supplied to the IC KR1533KP2 data inputs (see the right four columns) in order to put a full single-digit adder into operation. At the same time the operation of both multiplexer sections must be permitted ($\overline{V1} = \overline{V2} = 0$). Fig. 4.4 depicts single-digit adder circuit on the basis of the IC KR1533KP2 (see also table 4.6). Figure 4.5 shows voltage diagrams for the given circuit.

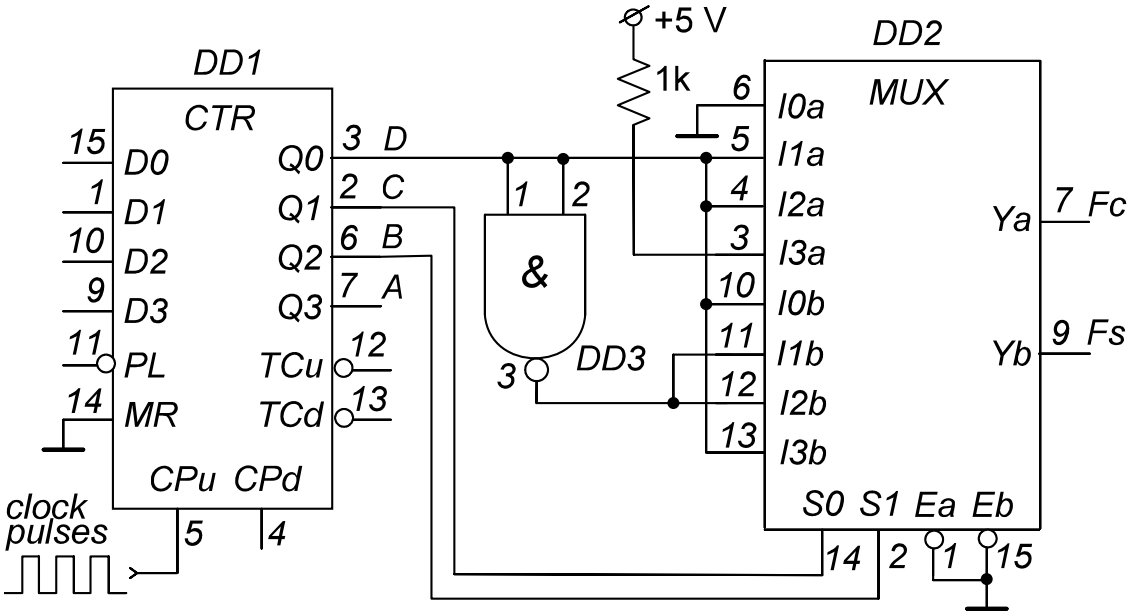


Fig. 4.4. Single-digit adder on the IC KR1533KP2 basis

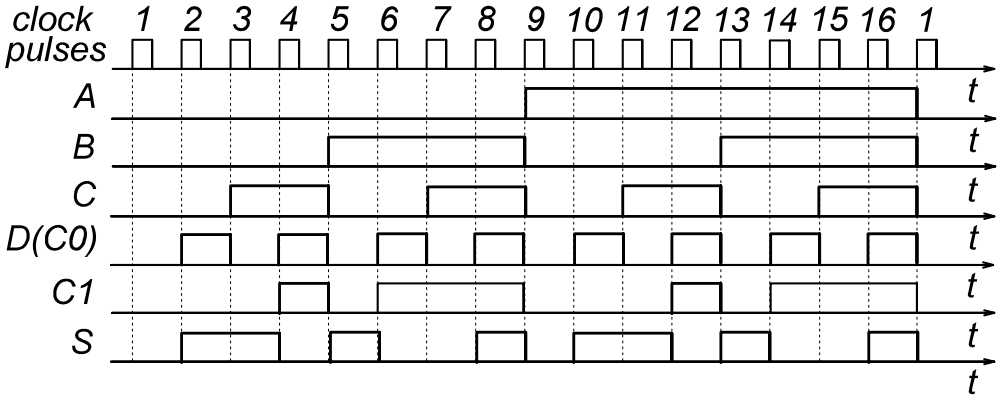


Fig. 4.5.Voltage diagrams for the circuit in Fig. 4.4

Table 4.6

ICs for the circuit in Fig. 4.4

IC type Circuitry symbol	KR1533IE7 DD1	KR1533KP2 DD2	KR1533LA3 DD3
Common	8	8	7
+5 V	16	16	14

In the lab work it is also necessary to implement a two-digit adder circuit (the truth table is shown in table 4.7) on the basis of the same multiplexer KR1533KP2, where *S1* is the sum of the variables *C* and *D* (low-order bits); *C1* – the carry to the next digit by adding the variables *C* and *D*; *S2* – the sum of the variables *A*, *B* (high-order bits) and *C1*; *C2* – the carry to the next digit by adding the variables *A*, *B* и *C1*. Table 4.7 also illustrates how to perform

the binding of the sum *S2* and carry *C2* with the variable *C1*. Table 4.8 shows what signals should be supplied (see columns *S1*, *C1*, *S2* и *C2*) to the IC KR1533KP2 data inputs in order to implement a two-digit adder.

Fig. 4.6 depicts the two-digit adder circuit on the IC KR1533KP2 basis (see also table 4.9). Fig. 4.7 shows voltage diagrams for the given circuit.

Table 4.7

Truth table for two-digit adder

C	*D*	*S1*	*C1*	*A*	*B*	*S2*		*C2*	
0	0	0	0	0	0	0	} *C1*	0	} *0*
0	1	1	0	0	0	0		0	
1	0	1	0	0	0	0		0	
1	1	0	1	0	0	1		0	
0	0	0	0	0	1	1	} $\overline{C1}$	0	} *C1*
0	1	1	0	0	1	1		0	
1	0	1	0	0	1	1		0	
1	1	0	1	0	1	0		1	
0	0	0	0	1	0	1	} $\overline{C1}$	0	} *C1*
0	1	1	0	1	0	1		0	
1	0	1	0	1	0	1		0	
1	1	0	1	1	0	0		1	
0	0	0	0	1	1	0	} *C1*	1	} *1*
0	1	1	0	1	1	0		1	
1	0	1	0	1	1	0		1	
1	1	0	1	1	1	1		1	

Table 4.8

Multiplexer input connection for the circuit in Fig. 4.6

Data inputs DD3 (*S1*)		Data inputs DD3 (*C1*)		Data inputs DD4 (*S2*)		Data inputs DD4 (*C2*)	
2D0	0	*1D0*	0	*1D0*	*C1*	*2D0*	0
2D1	1	*1D1*	0	*1D1*	$\overline{C1}$	*2D1*	*C1*
2D2	1	*1D2*	0	*1D2*	$\overline{C1}$	*2D2*	*C1*
2D3	0	*1D3*	1	*1D3*	*C1*	*2D3*	1

Table 4.9

ICs for the circuit in Fig. 4.6

IC type Circuitry symbol	KR1533KP2 DD1, DD4	KR1533LA3 DD2	KR1533IE7 DD3
Common	8	7	8
+5 V	16	14	16

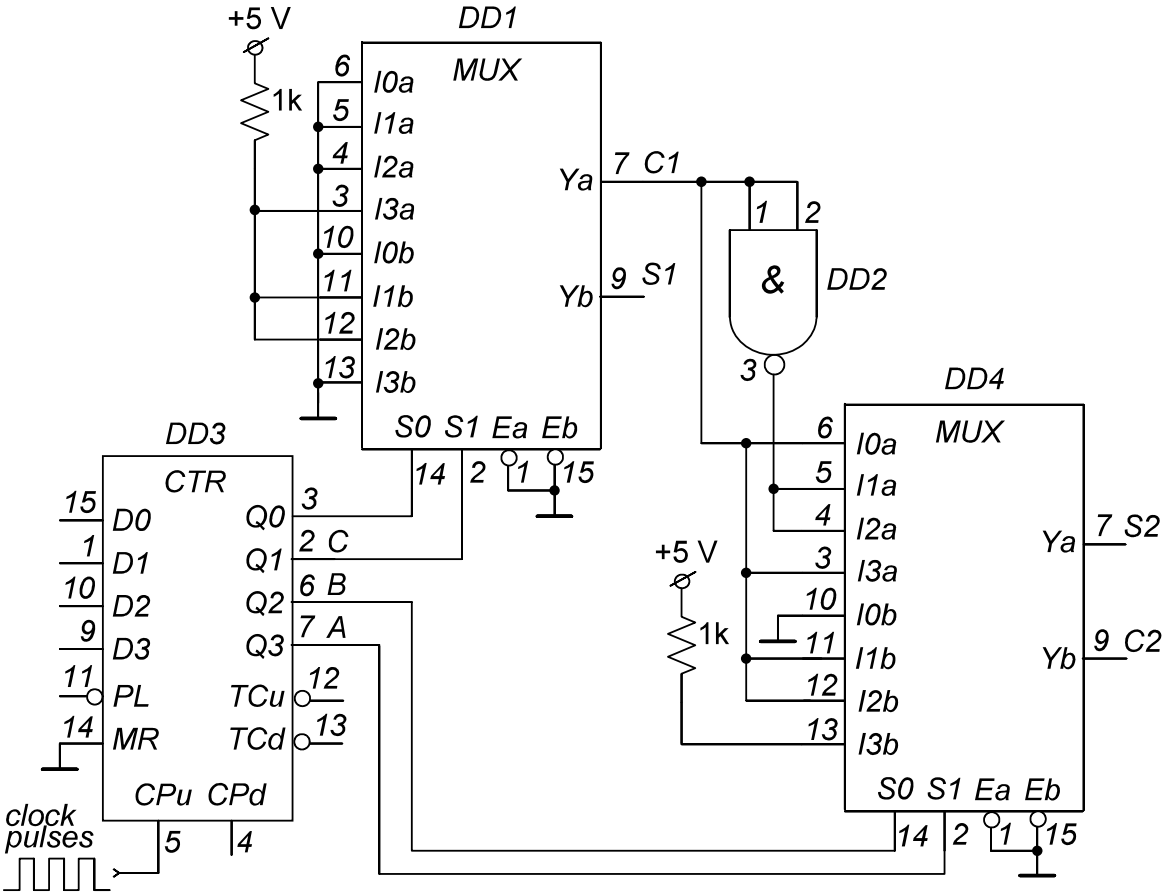


Fig. 4.6. Two-digit adder

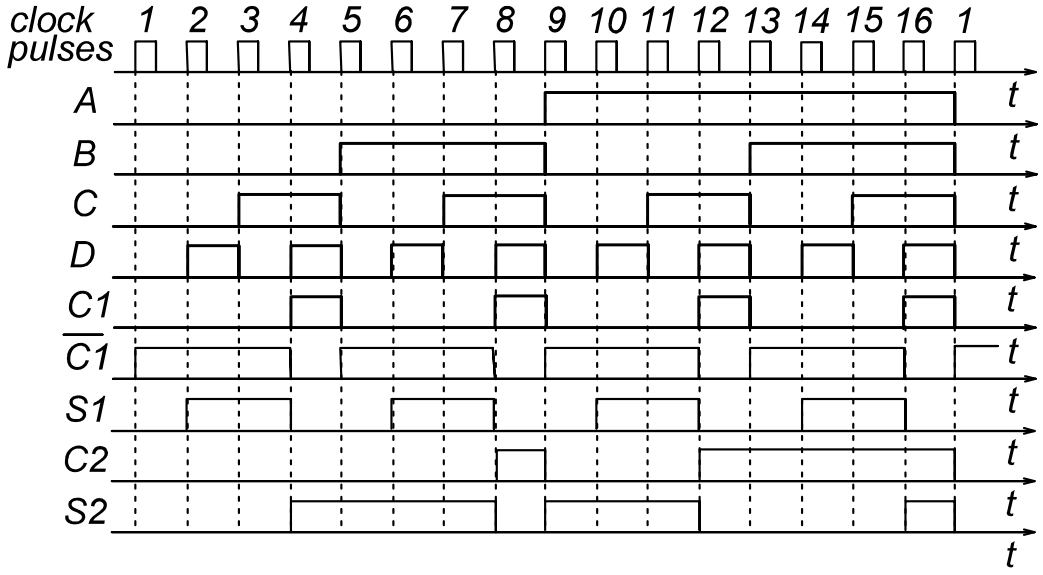


Fig. 4.7. Voltage diagrams in the circuit in Fig. 4.6

By means of the IC KR1533KP2 the circuit of the single-digit subtracter can also be accomplished. Let *Fd* be the difference of the variables *B* and *C* with the account of the borrowing *B*0=*D*, and *Fb* is the borrow from the high-order bit. In order to synthesize a circuit of subtracter, you should bind the output variables *Fd* and *Fb* with one of the input variables *B*, *C* or *D*. Table 4.10 introduces the single-digit subtracter output functions, and the signals which should be supplied to the IC KR1533KP2 data inputs in order to implement a full single-digit subtracter (binding with the variable *D*).

Table 4.10

Truth table for single-digit subtracter

B	*C*	*B0 =D*	*Fd*	*Fb*	Inputs KR1533KP2 (*Fd*)		Outputs KR1533KP2 (*Fb*)	
0	0	0	0	0	*2D0*	*D*	*1D0*	*D*
0	0	1	1	1				
0	1	0	1	1	*2D1*	$\overline{D}$	*1D1*	*1*
0	1	1	0	1				
1	0	0	1	0	*2D2*	$\overline{D}$	*1D2*	*0*
1	0	1	0	0				
1	1	0	0	0	*2D3*	*D*	*1D3*	*D*
1	1	1	1	1				

4.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for ‘Digital devices’ discipline is used. The procedure of the work is the same as described in section 1.4.

In order to carry out the laboratory work you’ll need a set of microcircuits: KR1533LA3 (SN74ALS00AN), KR1533LP5 (SN74ALS86N), KR1533KP2 (SN74ALS153N) and KR1533IE7 (SN74ALS193N) to form the variables *A*, *B*, *C* and *D*.

To record the form and parameters of the output functions the double-channel oscilloscope is used.

4.5 IN-LAB TASKS

1. Using table 4.1 synthesize the half-adder circuit on the base of simple logic gates (KR1533LA3 or KR1533LA4). Having assembled the circuit, verify its functioning principle according to the truth table.
2. Compare the sum column in the truth table with the truth table for the logic element ‘XOR’. Draw a conclusion.

3. Assemble the universal half-adder/half-subtracter circuit with the help of the microcircuits KR1533LA3 and KR1533LP5 (Fig. 4.2).

4. Study the circuit operation depending on the mode selection input value '*S*'.

5. Assemble a single-digit adder circuit on the IC KR1533KP2 basis at the pinboard (Fig. 4.4), and verify its functioning in accordance with the truth table.

6. Assemble a two-digit adder circuit on the IC KR1533LP5 basis at the pinboard (Fig. 4.6), and verify its functioning in accordance with the truth table.

7. Design a single-digit subtracter circuit on the IC KR1533KP2 basis and assemble it at the pinboard, and verify its functioning in accordance with the truth table (table 4.10).

4.6. QUESTIONS

1. What is the main peculiarity of the arithmetic units? Can they be regarded as logic structures?
2. Put down the function accomplished by one section of the IC KR1533LP5.
3. Is it possible to implement a single-digit adder by means of the KR1533KP7 multiplexer?
4. Draw the full single-digit adder circuit.
5. Draw the full single-digit subtracter circuit.
6. Draw the truth table for two-digit subtracter.
7. What is a complement code of a number and what is it used for?
8. Subtract the numbers 110011_2 and 111001_2 with the help of the complement code.
9. What is the correlation between addition operation time and adder digit capacity?
10. What is the difference between series and parallel adders?

Lab 5

ARITHMETIC INTEGRATED CIRCUITS STUDY

5.1 OBJECTIVES

The aim of the lab work is to study the functional power of adder and comparator circuits in digital devices.

5.2 PRE-TASKS

1. Study the typical structure of a multidigit adder implemented on the single-digit adder base.
2. Design a four-digit adder and subtracter circuits using the ICs K555IM5 (SN74LS183N).
3. Study the digital comparator functioning principle (comparing circuit). Acquaint with the IC K555SP1 (SN74LS85N).

5.3 BASIC THEORY

Let us look at the IC K555IM5, which functional power is suggested to be examined in the lab work.

IC K555IM5 (SN74LS183N) consists of two independent single-digit adders. With the help of several microcircuits of that sort it is possible to design a multidigit line of series addition. Fig. 5.1 presents the IC K555IM5 logic symbol and pin configuration. The functioning of one IC K555IM5 section is shown in table 5.1.

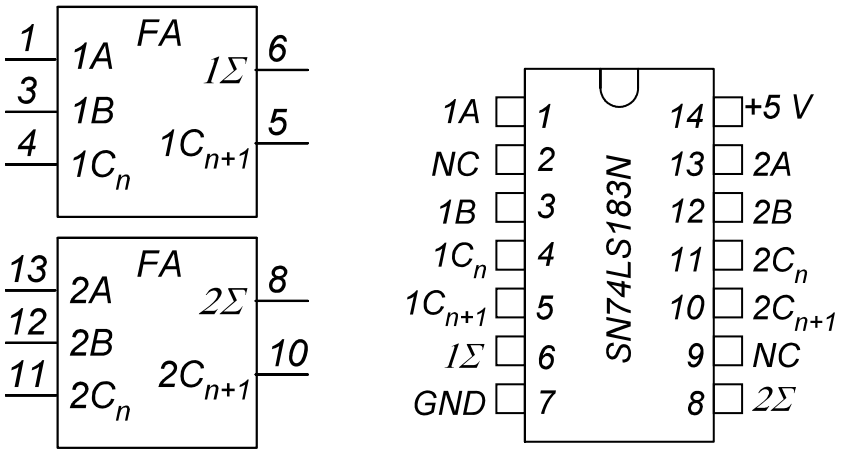


Fig. 5.1. Logic diagram of the IC K555IM5 and its pin configuration

Table 5.1

Truth table for the adder K555IM5

Input code			Output state	
C0	*B1*	*A1*	*S1*	*C1*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

In the lab work it is required to implement a two-digit adder circuit on the IC K555IM5 base. The circuit is shown in Fig. 5.2 (see also table 5.2), voltage diagrams are presented in Fig. 5.3.

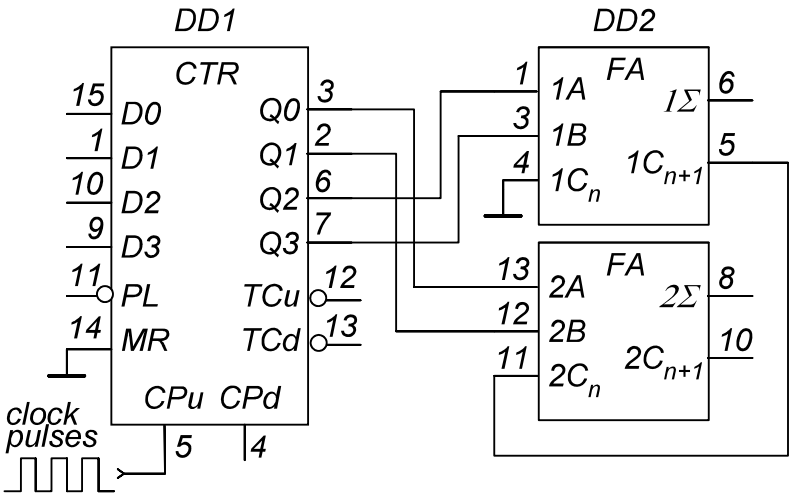


Fig. 5.2. Two-digit adder circuit on the IC K555IM5 base

Table 5.2

ICs for the circuit in Fig. 5.2

IMC type Circuitry symbol	KR1533IE7 DD1	K555IM5 DD2
Common	8	7
+5 V	16	14

The digits *A*, *B*, *C* and *D* of the summand codes are supplied to the corresponded adder inputs *A1*, *B1*, *A2* and *B2*. The signals are taken from the sum outputs *S1*, *S2* and carry output *C3*. At first, let's take into account that there wasn't any transfer *C0* from the previous order, therefore, we should

supply logic 0 to this input. The carry signal output of the first digit *C1* is provided to the carry input of the second digit *C2*. If we want to receive at the output a signal equal to the sum of input codes, the carry signal should be formed at the adder outputs of all digits step-by-step. Hence, in spite of the fact that a separate adder is used for adding in every digit, the real time of operation execution in the given circuit is equal to the sum transfer time of the carry signal from one order to another successively.

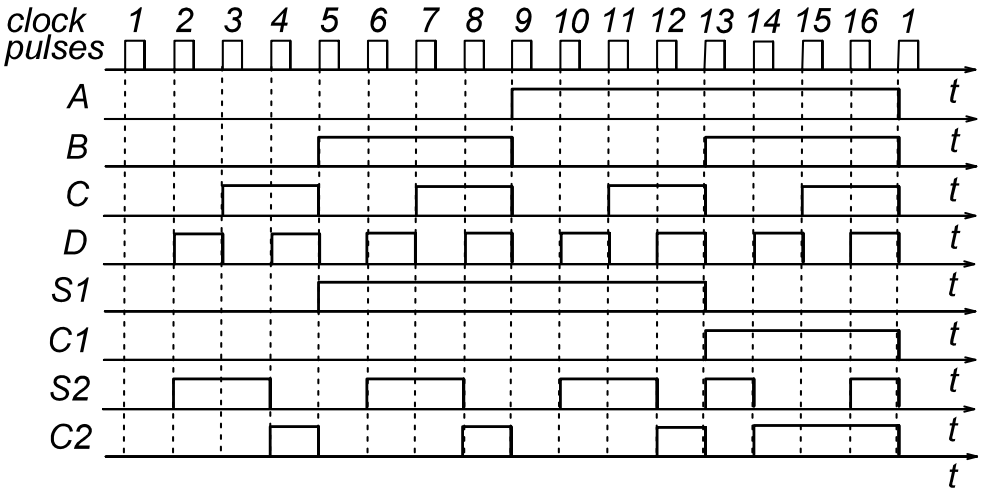


Fig. 5.3. Voltage diagrams for the circuit in Fig. 5.2

Let's now discuss the examples of a binary adder application for carrying out subtraction. At first, we'll design single-digit and two-digit subtracter circuits on the single-digit adder К555ИМ5 basis.

The subtraction of two numbers can be replaced by addition, if we substitute a subtrahend with the opposite sign number. Thus, for subtraction we can use adder circuits.

In the circuit in Fig. 5.4 (see also table 5.4) a single-digit subtracter on the IC K555IM5 basis is implemented. Here the subtrahend *D* is given in the complement code. The inverter (DD2) is used to form the inverse code of the number *D*. Number *C* is supplied to the adder DD3 input *A1*, the inverse code of the number *D* – to the adder input *B1*. The carry input *C0* is supplied with logic 1. This guarantees adding 'one' to the final result (i.e. result increment). So, $Fd = \overline{D} + C0 + C$. At the adder output *S1* we finally obtain the signal *Fd* – the difference between the numbers *C* and *D*, and at the output *C1* – number sign (0 – negative). Table 5.5 and Fig. 5.5 describe the functioning principle of this circuit.

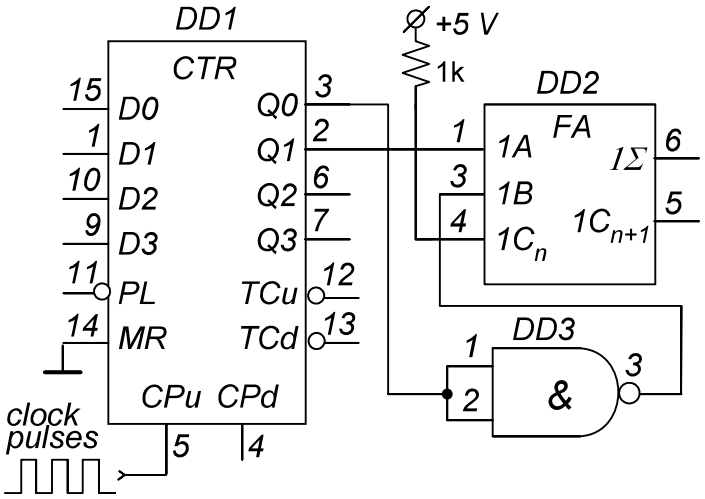


Fig. 5.4. Single-digit subtracter

Table 5.4

ICs for the circuit in Fig. 5.4

IC type Circuitry symbol	KR1533IE7 DD1	K555IM5 DD2	KR1533LA3 DD3
Common	8	7	7
+5 V	16	14	14

Table 5.5

Truth table for the circuit in Fig. 5.4

C0	*C*	*D*	$\overline{D}$	*Fd*
1	0	0	1	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	0

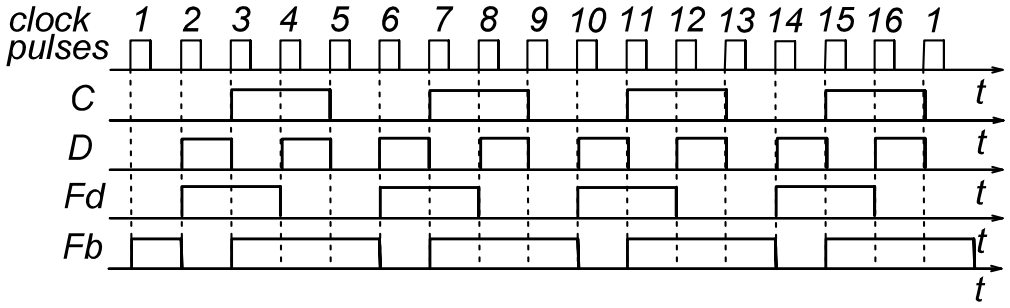


Fig. 5.5. Voltage diagrams of the circuit in Fig. 5.4

Let's now accomplish a two-digit subtracter circuit on the IC K555IM5 base, where the subtrahend is given in the complement code (Fig. 5.6, table 5.6). This is achieved as well as in the previous example by adding

logic 1 to the least significant bit of the subtrahend inverse code. The circuit subtracts signals *CD* from *AB*. Table 5.7 and diagrams in the Fig. 5.7 show its functioning.

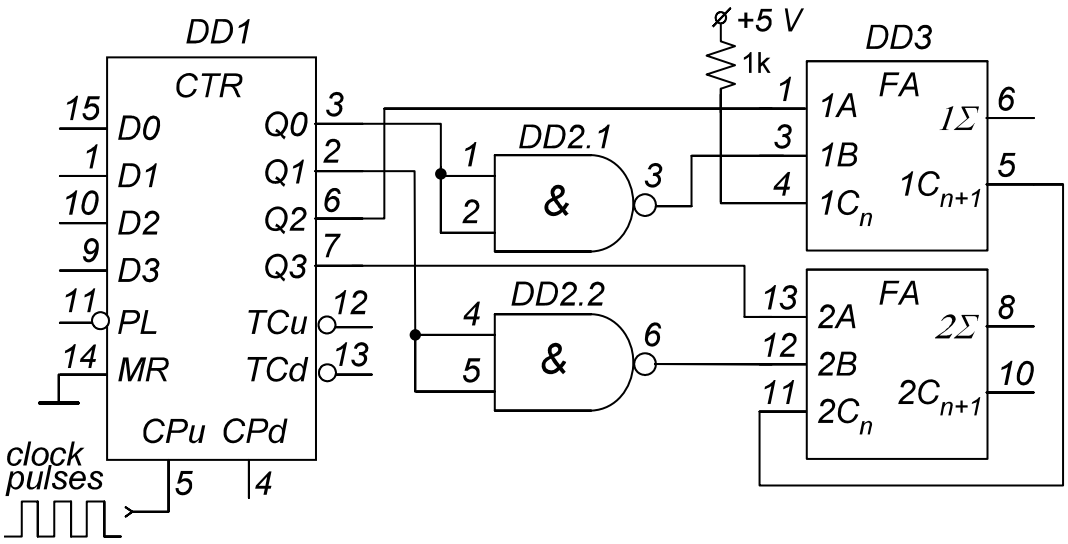


Fig. 5.6. Two-digit subtracter

Table 5.6

IC for the circuit in Fig. 5.6

IC type Circuitry symbol	KR1533IE7 DD1	KR1533LA3 DD2	K555IM5 DD3
Common	8	7	7
+5 V	16	14	14

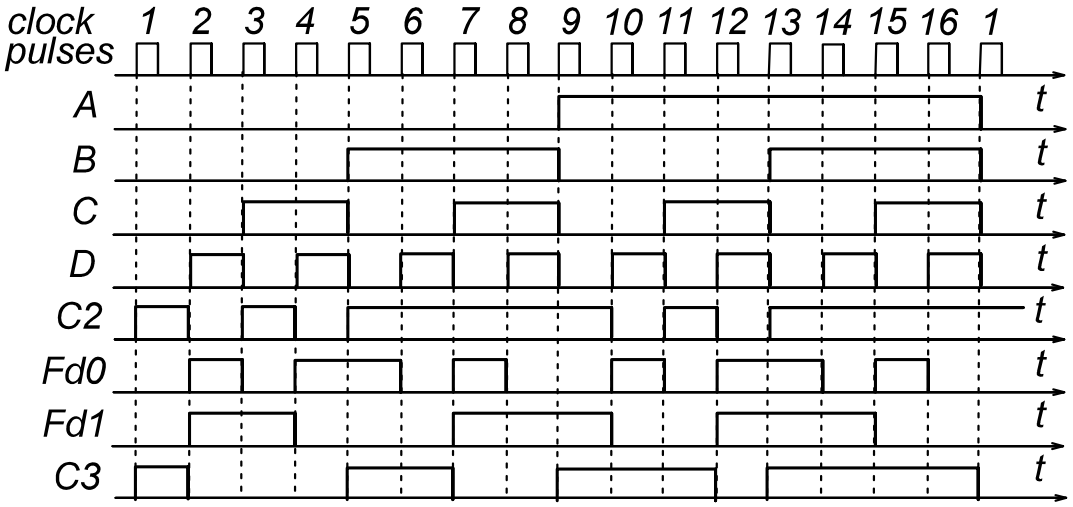


Fig. 5.7. Voltage diagrams of the circuit in Fig. 5.6

Table 5.7

Truth table for the circuit in Fig. 5.6

A	*B*	*C*	*D*	*C1*	*Fd1*	*Fd0*	*C3*
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	0
0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	1	0	1	1
0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	1	0	0
1	0	0	0	1	1	0	1
1	0	0	1	0	0	1	1
1	0	1	0	1	0	0	1
1	0	1	1	0	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	0	1

With the help of the adders multiplication can also be done. Let's have a look at the implementation of the two-digit code matrix multiplier circuit on the single-digit adder base. The circuit demonstrates the hardware multiplier construction logic, which is closely connected with the conventional multiplication execution algorithm based on summing the partial product of the multipliers. E.g., let's multiply two-digit binary codes a_1a_0 and b_1b_0:

$$\begin{array}{cccc} & a_1 & a_0 & \\ \times & b_1 & b_0 & \\ \hline & b_1a_1 & b_1a_0 & \\ + & & b_0a_1 & b_0a_0 \\ \hline M_3 & M_2 & M_1 & M_0 \end{array}$$

Table 5.8 shows all possible values of the input variables a_1, a_0, b_1, b_0 and the corresponding values of the output variables M_3, M_2, M_1, and M_0.

In contrast to the use of elementary functions succession for multiplication and division, hardware multipliers have significantly increased computing system speed.

Schematic circuit of the device is shown in Fig. 5.8 (see also table 5.9). Partial product of the multipliers bit sites are formed by means of the logic gates DD2 and DD3 (KR1533LA3). Summing up these products using the single-digit adders, the result code can be found. The given structure is called matrix multiplier unit.

Table 5.8

Truth table for the two-digit matrix multiplier

a_1	a_0	b_1	b_0	M_3	M_2	M_1	M_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

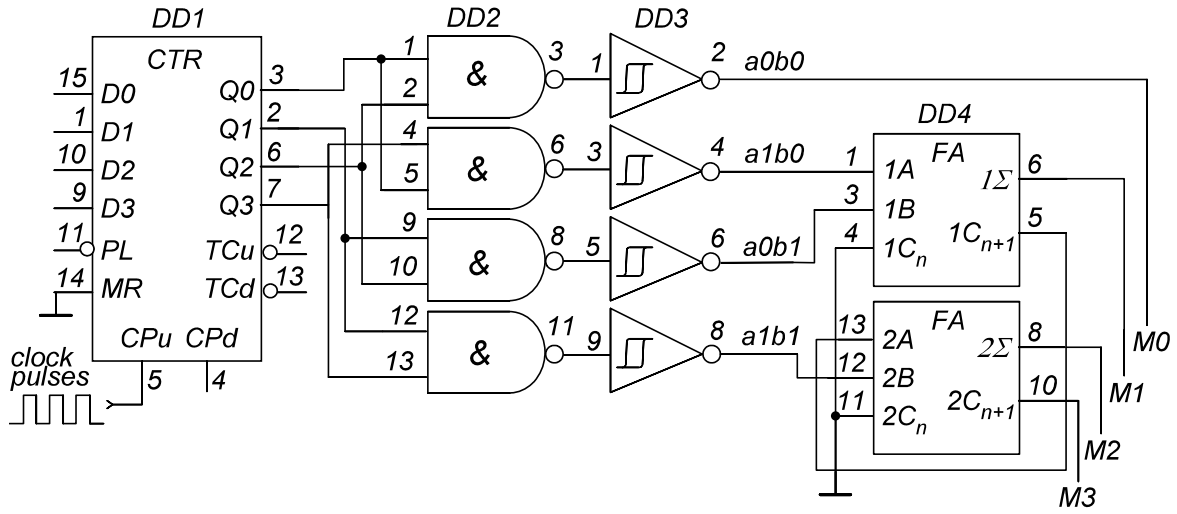


Fig. 5.8. Two-digit code matrix multiplier circuit

Table 5.9

ICs for the circuit in Fig. 5.8

IC type Circuitry symbol	KR1533IE7 DD1	KR1533LA3 DD2	KR1533LN1 DD3	K555IM5 DD4
Common	8	7	7	7
+5 V	16	14	14	14

Similarly we can design a matrix multiplier unit, operating with any digit capacity input codes. At the same time the partial product adders will keep the same capacity. The adders capacity of the couple of the partial product will be one digit higher, and the adders capacity of the quad of the partial product will be twice as more as the capacity of the partial product adders, etc.

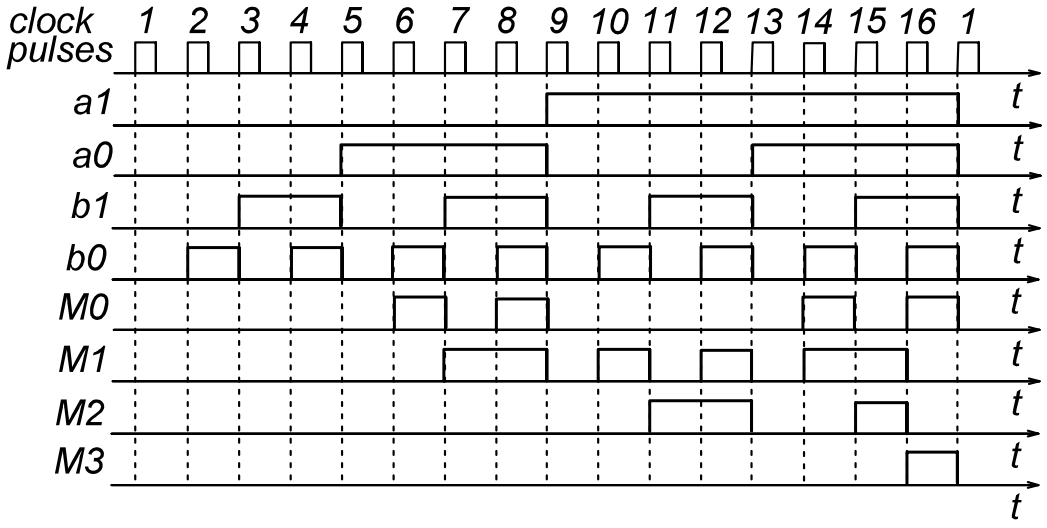


Fig. 5.9. Voltage diagrams of the circuit in Fig. 5.8

According to the pre-task, it is required to implement a binary four-digit adder on the base of the microcircuits K555IM5 connected in tandem. The device should possess 9 inputs (4 per each variable and 1 carry input from low-order bit) и 5 outputs (4 digits и 1 carry output to the high-order bit).

Let's now consider the example of adding two numbers in a binary code: 10+9:

$$\begin{array}{rcccc} & 1 & 0 & 1 & 0 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 & 1 \end{array}$$

i.e. when adding 10 and 9 we obtain number 19.

The adding by means of the hardware is implemented by supplying logic 0 and 1 to the adder inputs (table 5.10).

Table 5.10

Addition of numbers 1010_2 u 1001_2

C0	*A1*	*A2*	*A3*	*A4*	*B1*	*B2*	*B3*	*B4*	*S1*	*S2*	*S3*	*S4*	*C4*
0	0	1	0	1	1	0	0	1	1	1	0	0	1

Comparator circuit (digital comparator) is used to compare two binary numbers and can be applied, for instance, in monitoring and regulating

systems. Let's examine the IC K555SP1 as an example of a comparator circuit.

IC K555SP1 (SN74LS85N) is a comparator having two groups of inputs *A3A2A1A0* and *B3B2B1B0*. Fig. 5.10 shows the IC K555SP1 logic symbol and pin configuration. The truth table describing the IC functioning is presented in table 5.11.

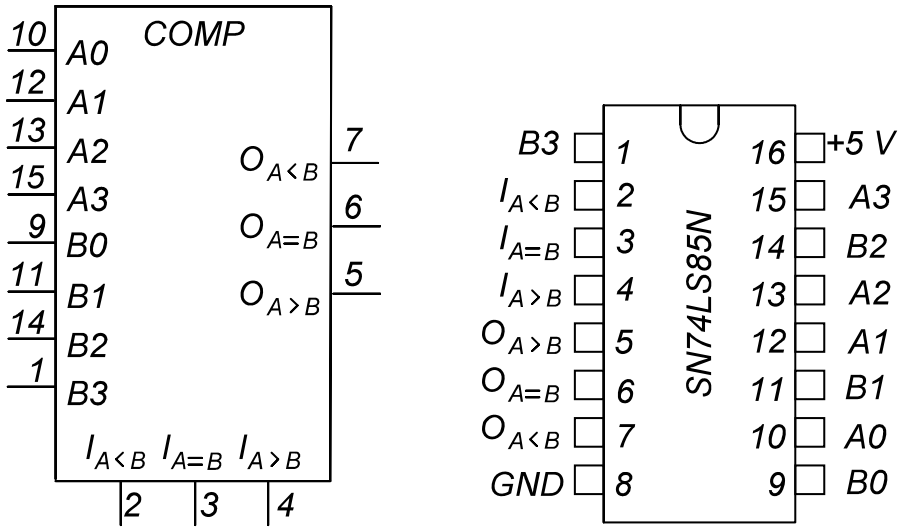


Fig. 5.10. IC K555SP1 logic symbol and pin configuration

Table 5.11

Truth table for the comparator K555SP1

A, B				I(A>B)	I(A=B)	I(A<B)	A>B	A=B	A<B
A3>B3	x	x	x	x	x	x	1	0	0
A3<B3	x	x	x	x	x	x	0	0	1
A3=B3	A2>B2	x	x	x	x	x	1	0	0
A3=B3	A2<B2	x	x	x	x	x	0	0	1
A3=B3	A2=B2	A1>B1	x	x	x	x	1	0	0
A3=B3	A2=B2	A1<B1	x	x	x	x	0	0	1
A3=B3	A2=B2	A1=B1	A0>B0	x	x	x	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	x	x	x	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	0	0	1	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	x	1	x	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	1	0	1	0	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	0	1	0	1

The digits of the first number (*A*) are provided to the first group of inputs, the digits of the second number (*B*) – to the other group. Three comparator outputs fix the result of the comparison by appearing of logic 1 at

the corresponded output. At the first output it is set when the numbers are equal ($A=B$), at the second – when $A<B$, at the third – when $A>B$. The values at the inputs $A=B$, $A<B$ and $A>B$ influence the result of the comparison only if *A3A2A1A0=B3B2B1B0*. However, if the input $A=B$ is supplied with 1, then $F_{A=B}=1$, the values at the inputs $A<B$ и $A>B$ can be any. If the input $A=B$ is supplied with 0, then $F_{A=B}=0$, and $F_{A<B}$ и $F_{A>B}$ will be governed by the input values $A<B$ and $A>B$.

Figure 5.11 shows the circuit implementing comparison of two binary numbers *A1A0* and *B1B0* in accordance with the table 5.11.

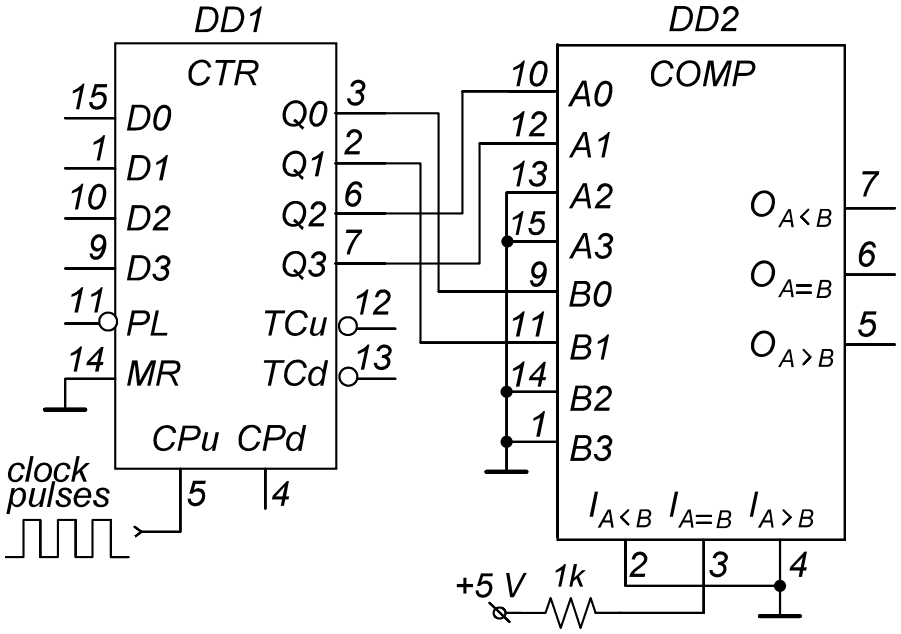


Fig. 5.11. Comparison circuit of two binary numbers

Table 5.12

IC for the circuit in Fig. 5.11

IC type Circuitry symbol	KR1533IE7 DD1	K555SP1 DD2
Common	8	8
+5 V	16	16

5.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4.

In the lab work the arithmetic unit microcircuits are applied. Two-digit adder circuits, single-digit and two-digit subtracter circuits, two-digit code

matrix multiplier circuit on the IC K555IM5 and simple logic IC KR1533LA3 basis are designed and analysed. The adding (IC K555IM5 with capacity increase) and comparison (digital comparator K555SP1) of four-digit variables is fulfilled. In order to form logic variables (as well as in the previous works) the binary counter KR1533IE7 is used.

To record the form and parameters of the output function the double-channel oscilloscope is used.

5.5 IN-LAB TASKS

1. Draw a two-digit adder scheme on the base of IC K555IM5 (Fig. 5.2), verify its functional principle in accordance with the truth table.
2. Replace high-order and low-order bits; compare the obtained diagrams with the result of the previous lab work.
3. Assemble the single-digit subtracter circuit using the IC K555IM5. Verify its functional principle in accordance with the truth table.
4. Assemble the two-digit subtracter circuit with the help of the IC K555IM5. Verify its functional principle in accordance with the truth table.
5. Assemble the two-digit code matrix multiplier at the pinboard and verify its functional principle in accordance with the truth table.
6. Increase the capacity of the adding circuit up to four using the IC K555IM5. Having added and subtracted two four-digit binary numbers (they will be given by the lecturer) verify the circuit functioning.
7. When designed the comparison circuit for two numbers, analyse the IC K555SP1 functioning in a static mode.
8. Using the digital comparator K555SP1 implement the comparison circuit for two two-digit numbers and verify its functional principle in a dynamic mode when the variables formed by the counter KR1533IE7 are connected.

5.6 QUESTIONS

1. Why are adders referred to combinational devices?
2. Will the adding result of the multidigit adder change if we replace the summands?
3. Will the subtracting result of the multidigit adder change if we replace the subtrahend, given in the complement code, and the minuend?
4. Draw the subtracting circuit for eight-bit signed numbers by means of the adder K555IM3 (74HC283N). Binary numbers will be given by the lecturer.
5. Draw the decimal correction circuit for binary numbers.

6. Draw the addition-subtraction circuit of two-digit code on the IC K555IM5 base.

7. Draw the four-digit subtracter on the IC K555IM5 base.

8. Give examples of the comparison circuit application (where it is required to compare two numbers).

9. Design the capacity increase circuit for the comparator K555SP1 using the comparator cascade connection.

10. How can you identify the number sign at the subtracter output in the circuit in Fig. 5.6?

Lab 6

STUDY OF FLIP-FLOPS AND CIRCUITS ON THEIR BASIS

6.1 OBJECTIVES

The aim of the lab work is to analyse *J-K* flip-flop functioning, its basic characteristics and peculiarities of application. You will also examine the synthesis methods of electronic units on the basis of flip-flops – counters, registers, pulse distributors.

6.2 PRE-TASKS

1. Study D-latch functioning principle.
2. Study T-flip-flop functioning principle.
3. Study J-K flip-flop functioning principle.
4. Study the IC KR1533TV9 (SN74ALS112A) operational principle.
5. Design D-latch and T-flip-flop on the JK-flip-flop (KR1533TV9) base.
6. Study the counters and registers building methods on the base of various types of flip-flops.

6.3 BASIC THEORY

Flip-flop is a logic device able to store 1 data bit. The name of the information unit *'bit'* originates from the words *'binary'* and *'digit'*. All the devices having two steady states are considered to be trigger (flip-flop).

Flip-flops can be divided into asynchronous and synchronous (or clocked). Synchronous flip-flops can run both on single-stage and two-stage circuits.

Fig. 6.1 shows single-stage SR-latch circuit implemented with NAND logic gates. The flip-flop operating modes are presented in table 6.1, and performance diagram – in Fig. 6.2.

The flip-flop switching takes place by the pulse positive edge at the clock input *C*. When the clock input *C* voltage level is high any changes at the inputs *R* or *S* lead to the flip-flop triggering, therefore, when *C*=1 it is extremely undesirable to change inputs *R* and *S* states. In this situation we say that a flip-flop is *'transparent'* relative to *S* and *R* signals.

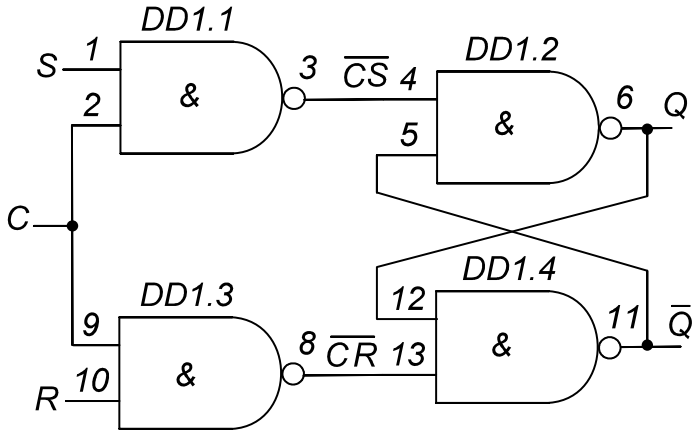


Fig. 6.1. Single-stage SR flip-flop circuit (DD1 – KR1533LA3)

Table 6.1

Synchronous SR flip-flop state table

S	*R*	*C*	*Q*	$\overline{Q}$	Operating mode
0	0	1	*Q*	$\overline{Q}$	Hold state (storage)
1	0	1	1	0	Set
0	1	1	0	1	Reset
1	1	1	1	1	Not allowed
x	x	0	*Q*	$\overline{Q}$	No action (storage)

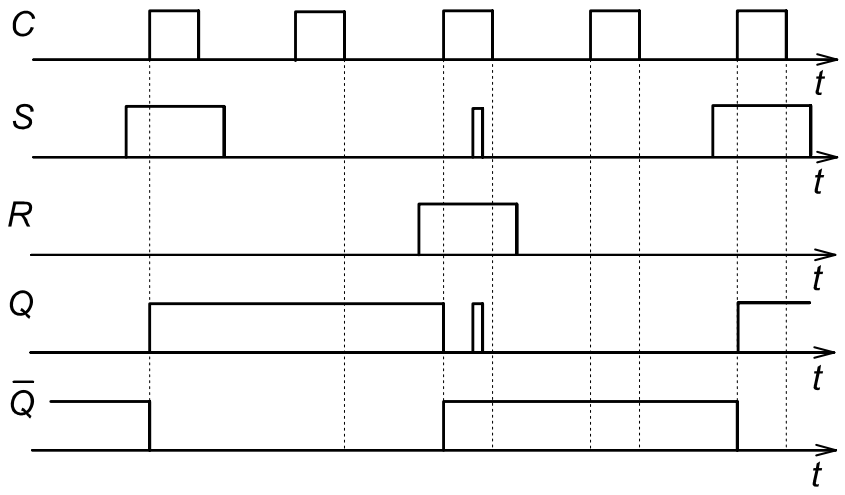


Fig. 6.2. Single-stage SR flip-flop performance diagram

Fig. 6.3 shows two-stage SR flip-flop circuit. The flip-flop consists of two single-stage SR flip-flops connected in series. The first flip-flop is called *'a master'*, and the second – *'a slave'*. As you can see, clock pulses go to the second flip-flop through the inverter. Thus, when the input *C* level is active, the state changes at the control inputs will not cause immediate change of the

output state. This is the fundamental difference between two-stage and single-stage SR flip-flops. The two-stage SR-flip-flop operating modes are given in table 6.2. The flip-flop is switched by the negative edge of the clock input *C* pulse (Fig. 6.4).

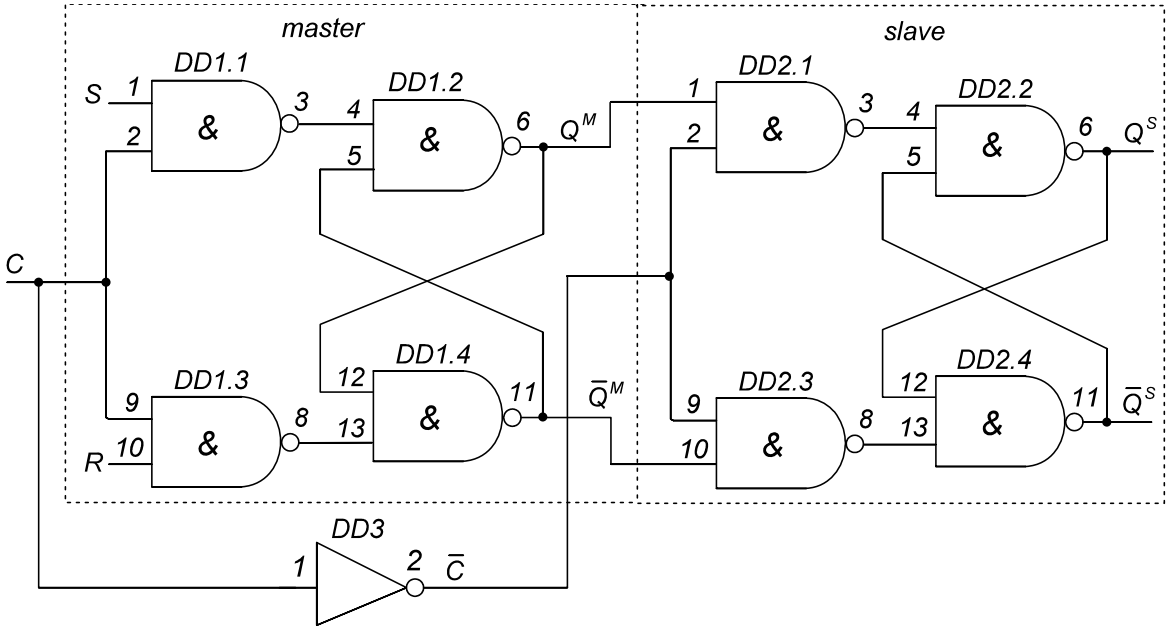


Fig. 6.3. Two-stage SR flip-flop schematic circuit (DD1, DD2 – KR1533LA3, DD3 – KR1533LN1)

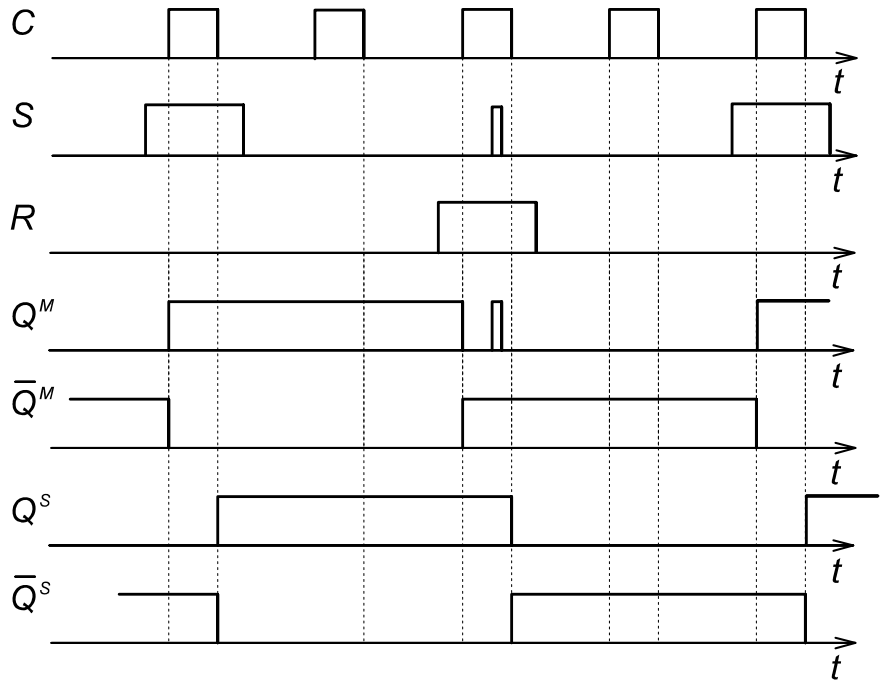


Fig. 6.4. Two-stage SR flip-flop performance diagram

Table 6.2

Two-stage synchronous SR flip-flop state table

R	S	C	Q	$\overline{Q}$	Operating mode
0	0	x	Q	$\overline{Q}$	Storage
1	0	↓	0	1	Reset
0	1	↓	1	0	Set
1	1	↓	1	1	Not allowed

Let us now consider the two-stage JK flip-flop implementation with NAND logic gates. The circuit is shown in Fig. 6.5 (see also table 6.3). JK flip-flop is a universal flip-flop, because various flip-flops can be built on its basis. In contrast to SR flip-flops, in JK flip-flops there are no unallowed combination of input signals, and no X-state can appear at the output.

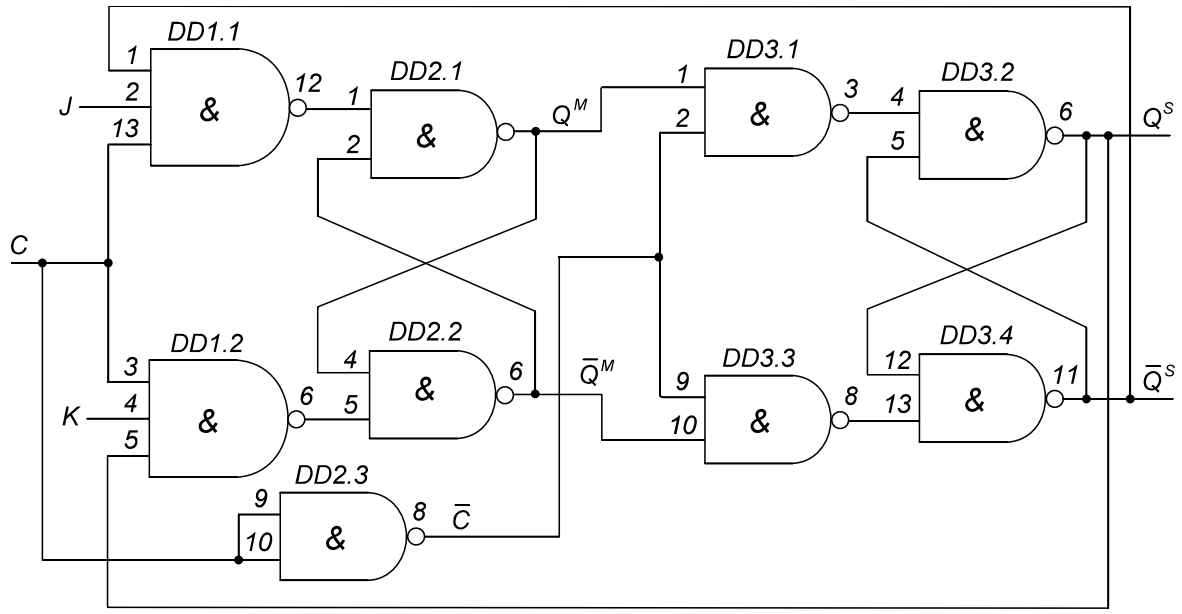


Fig. 6.5. Two-stage JK-flip-flop

Table 6.3

ICs for the circuit in Fig. 6.5

IC type Circuitry symbol	KR1533LA4 DD1	KR1533LA3 DD2, DD3
Common	7	7
+5 V	14	14

In the lab work you are suggested to examine the IC KR1533TV9 operation.

IC KR1533TV9 (SN74ALS112A) consists of two JK flip-flops with asynchronous reset and set inputs. Fig. 6.6 presents the IC KR1533TV9 logic

symbol and pin configuration. Every flip-flop has set $\overline{S}$ and reset $\overline{R}$ inputs, data inputs *J* and *K*, clock input $\overline{C}$ and two complementary outputs *Q* and $\overline{Q}$. The functioning of one KR1533TV9 section occurs with accordance to the table 6.4, and we can recognize that there are seven operating modes here.

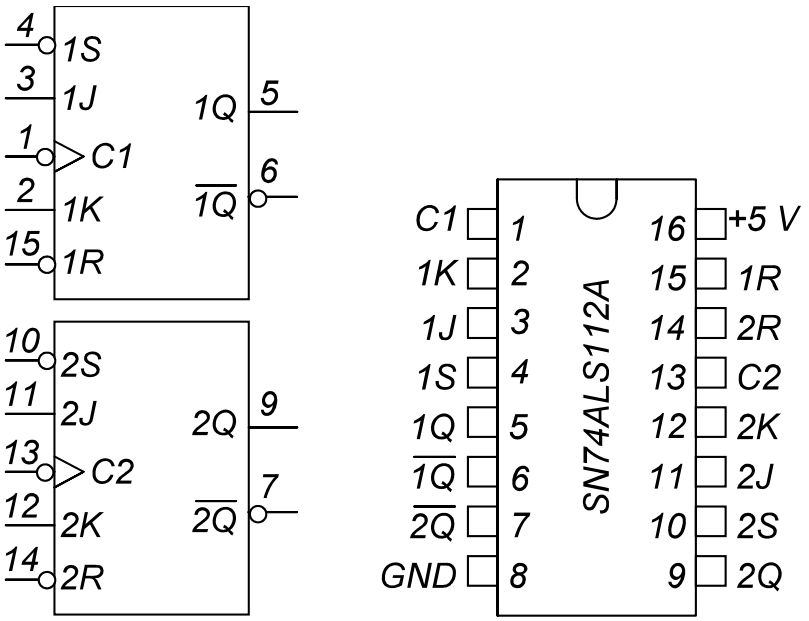


Fig. 6.6. IC KR1533TV9 logic symbol and pin configuration

Table 6.4

KR1533TV9 state table

Operating mode	Inputs					Outputs	
	$\overline{S}$	$\overline{R}$	$\overline{C}$	*J*	*K*	*Q*	$\overline{Q}$
Asynchronous set	0	1	x	x	x	1	0
Asynchronous reset	1	0	x	x	x	0	1
Not allowed	0	0	x	x	x	1	1
Toggle	1	1	↓	1	1	$\overline{q}$	*q*
Reset	1	1	↓	0	1	0	1
Set	1	1	↓	1	0	1	0
Storage (no change)	1	1	↓	0	0	*q*	$\overline{q}$

Performance diagrams of one KR1533TV9 section are shown in Fig. 6.7. The flip-flop state is changed at the negative edge of the clock pulse.

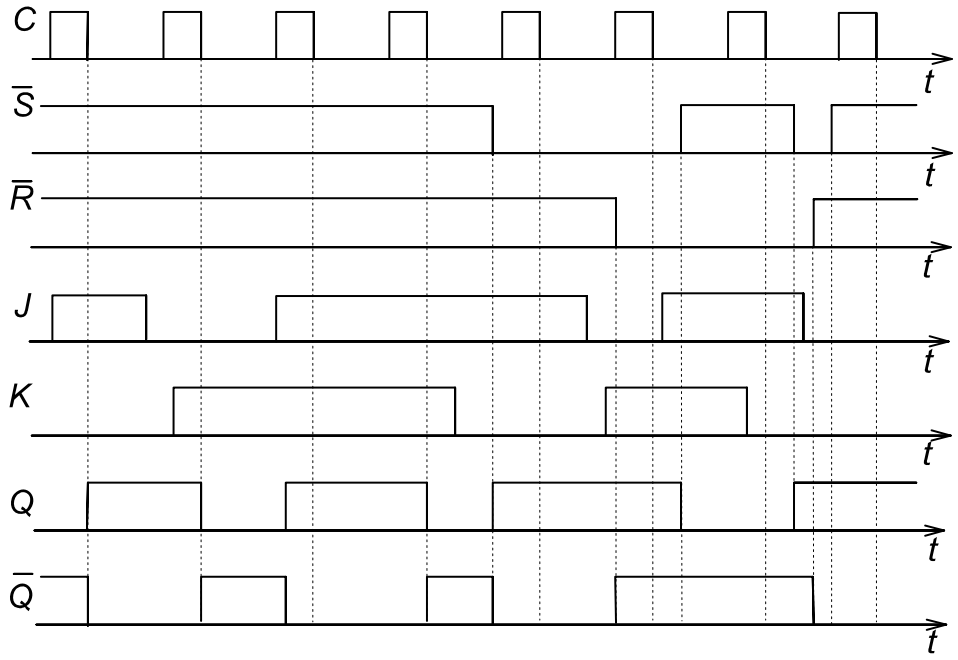


Fig. 6.7. Performance diagrams of one section of the IMC KR1533TV9

The asynchronous inputs $\overline{S}$ and $\overline{R}$ possess the first priority, i.e. reset ($\overline{S}$=1, $\overline{R}$=0) or set ($\overline{S}$=0, $\overline{R}$=1) occurs despite the input *C* state. If the inputs $\overline{S}$ and $\overline{R}$ are supplied simultaneously with the low-level signals, the flip-flop is in the so-called *'forbidden state'*, when $Q=\overline{Q}=1$.

Moreover, we should bear in mind that if there is high voltage at the clock input, the signals at the data inputs must not be switched. Otherwise, these switching can be transmitted to the output, and the flip-flop can lose its opacity at that moment. This fact should be taken into account when some flip-flops, especially single-stage flip-flops, are applied. If a flip-flop is transparent and permeable, a digital device circuit should be designed in such a way that the signals at the control inputs will change only if the clock signal is idle. When it is active, the control input signals will remain constant. In other case false operation of the device can take place. As for KR1533TV9, it is tolerant enough to the change of *J* and *K* signals when signal *C* is in progress, but sometimes it can also become transparent.

6.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4.

In order to accomplish the lab work you'll need a set of microcircuits:

KR1533LA3, KR1533LA4, KR1533TV9 and KR1533TL2 (SN74ALS14N). The last device contains six inverters with Schmitt trigger inputs (see Fig. 6.8). This logic gates are suggested to use for clock pulses generating.

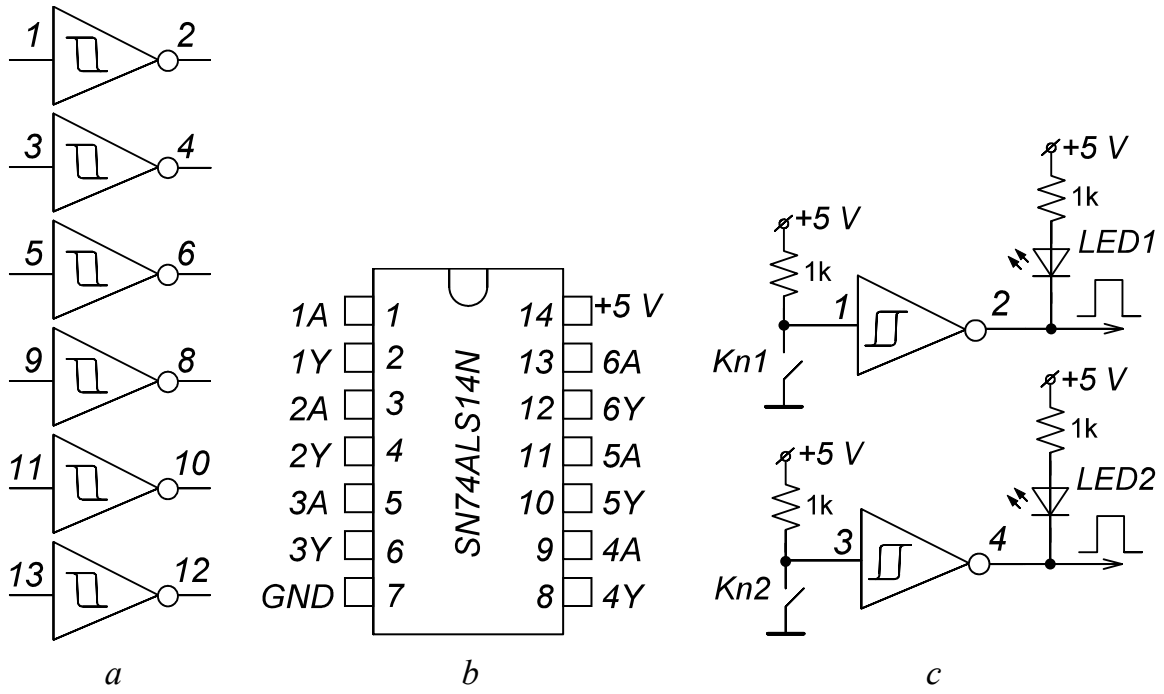


Fig. 6.8. IC KR1533TL2 logic symbol, pin configuration and clock-pulse driver on the base of this IC

It's strongly recommended to use JK flip-flop KR1533TV9 (which operates in SR flip-flop mode) as clock-pulse driver when working in a step-by-step (manual) mode. In order to form the control signals *R* and *S* the buttons *Кн1* and *Кн2* of the module UIK-1 are used. Fig. 6.9 shows the flip-flop KR1533TV9 connection circuit for clock-pulse generating in the manual mode. To set and reset the flip-flop (to form *R* and *S* signals) in the circuit in Fig. 6.9, *a* the buttons *Кн1* and *Кн2* of the module are used. The other pins of the KR1533TV9 are connected to the common bus. Thus, by setting and resetting the flip-flop the synchronization pulse with certain time parameters is formed. This circuit is the most preferable due to excluding contact bounce. Moreover, it allows forming long-duration clock pulse without the button keeping pushed.

The alternative way to form clock pulses is shown in Fig. 6.9, *b*. To perform it we need one button connected to the flip-flop input $\overline{S}$, and the input $\overline{R}$ is grounded. When the button *Кн1* is being pushed, the flip-flop is in the X-state, and $Q = \overline{Q} = 1$. When we do not push the button *Кн1*, the flip-flop is reset, i.e. Q becomes zero. This circuit as well as the circuit with

Schmitt trigger based logic gates allows to avode the contact bounce only partially.

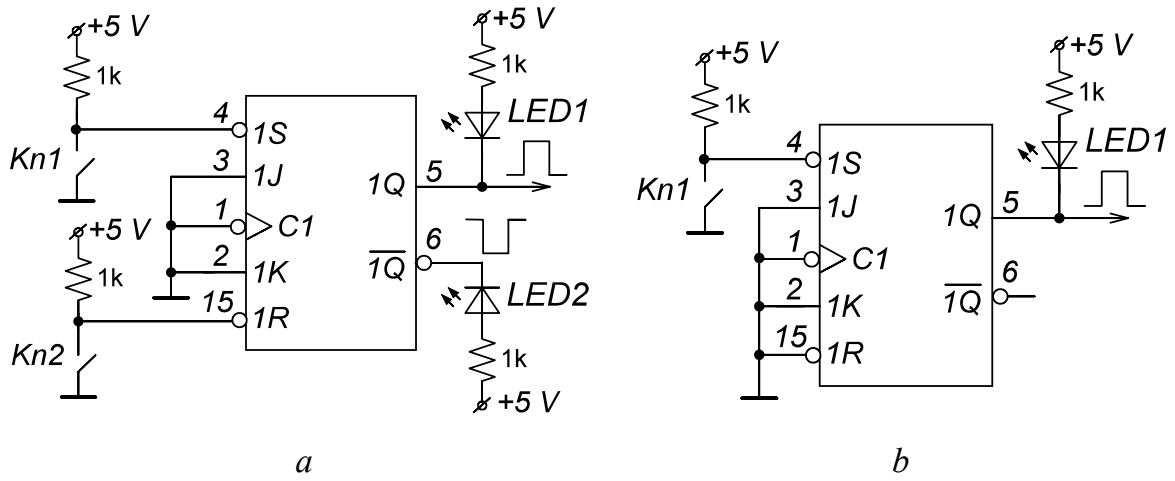


Fig. 6.9. Clock-pulse driver circuits based on KR1533TV9 flip-flop

6.5 IN-LAB TASKS

1. Assembly an asynchronous SR flip-flop on the JK flip-flop KR1533TV9 base to form clock pulses in the manual mode. In order to observe the flip-flop output states you are advised to connect the light-emitting diodes available on the pinboard. Upon this the diode will emit light when the cathode is supplied with logic 0.

2. Assembly the two-stage SR flip-flop circuit on the basis of the simple logic gates. (Fig. 6.3). Verify its operation in the step-by-step mode according to the state table (table 6.2). Take the signal diagrams at the outputs of the first and second flip-flop stages. The clock signal *C* is formed as described in paragraph 1.

3. Assembly the two-stage JK flip-flop circuit with the help of the simple logic gates. (Fig. 6.5). Verify its operation in the step-by-step mode. Take the signal diagrams at the outputs of the first and second flip-flop stages.

4. Verify the JK flip-flop KR1533TV9 state table in the step-by-step mode according to the state table (table 6.4). Take the JK flip-flop output signal diagram in accordance with the proposed series of the input signals (Fig. 6.7).

5. Assembly the three-digit up-down counter with use of two ICs KR1533TV9 and supplementary logic (Fig. 6.9, table 6.5). Analyse the up-down counter operation in the step-by-step and dynamic modes. Draw the voltage waveforms for the following cases: a) *M*=1, *N*=0; b) *M*=0, *N*=1. To observe the signals *Q1–Q3* in the step-by-step mode you are advised to use

LEDs or 7-segment display coupled with decoder 533ID18. In the dynamic mode the sequence of pulses from the built-in generator output or frequency divider (KR1533IE19) output and the oscilloscope for output signal registration are used.

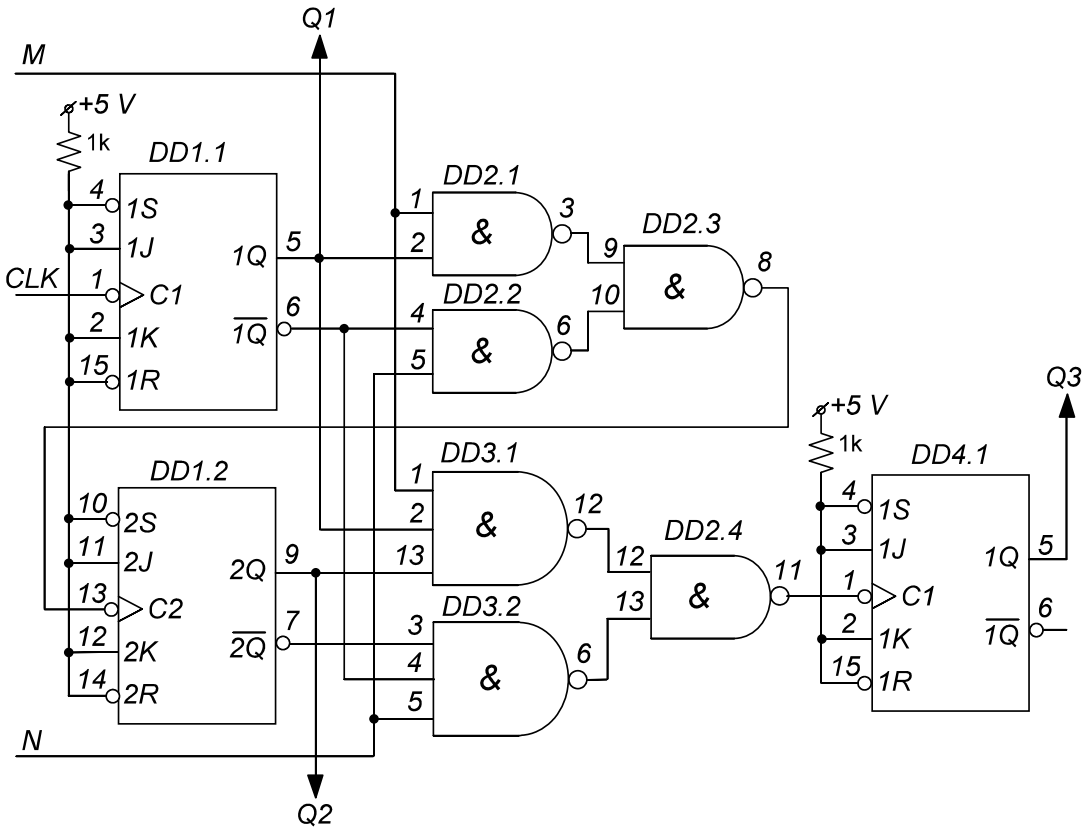


Fig. 6.9. Three-digit up-down counter on the IC KR1533TV9 base

Table 6.5

ICs for the circuit in Fig. 6.9

IC type Circuitry symbol	KR1533TV9 DD1, DD4	KR1533LA3 DD2	KR1533LA4 DD3
Common	8	7	7
+5 V	16	14	14

6. Assembly the ring register on two IC KR1533TV9 base (Fig. 6.10), examine its operation and present the functioning diagrams.

7. Assembly the cross-coupled register (Johnson ring counter) on two IC *KR1533TV9* base (Fig. 6.11). Examine its functioning, take waveforms.

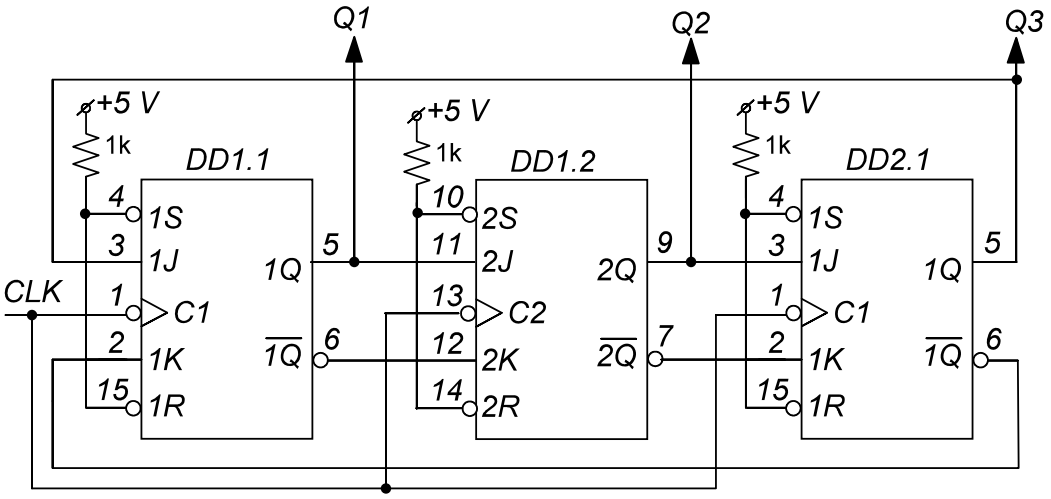


Fig. 6.10. Ring register circuit based on the IC KR1533TV9

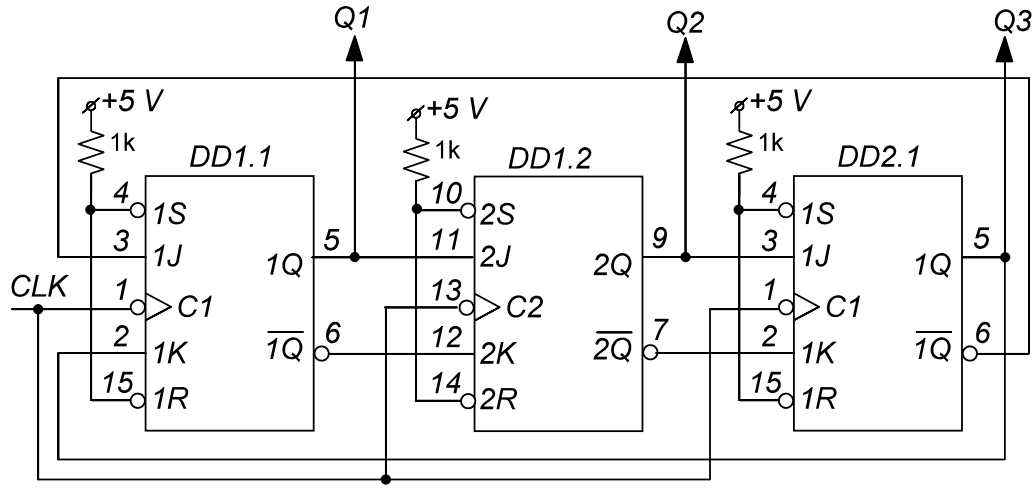


Fig. 6.11. Cross-coupled register circuit based on the IC KR1533TV9

6.6 QUESTIONS

1. What does transparent ‘latch’ mean?
2. Does the flip-flop KR1533TV9 possess the property of transparency?
3. What is ‘flip-flop permeability’? Is the flip-flop KR1533TV9 permeable?
4. How much flip-flops do you need to design a cross-coupled register when k_{count}= 9, 10, 12.
5. Can we load a codeword into a ring shift register in parallel? Why? How?
6. What signals should be supplied to the inputs *M* and *N* in the up-down counter circuit when counting up and down?
7. Draw the circuit, state table and functioning diagrams for D-, T-, SR- or JK flip-flop according to the lecturer’s task.

Lab 7

ELECTRONIC COUNTER STUDY

7.1 OBJECTIVES

The aim of the lab work is to clarify the counter operating principle and its role in digital devices. While making the experiments, the students will also get the skills of the counter performance control and frequency counter-divider design with the required counting coefficient.

7.2 PRE-TASKS

1. Study the synchronous counter circuit on the IC KR1533IE7 base, its functioning principle and operating mode.
2. Get to know the counter design concept on the counter-divider basis with a random counting coefficient ($k_{count}=N$). Design the counter using IC KR1533IE7. The k_{count} is given by the lecturer.
3. Get to know the multibit counter design methods. Build n-bit counter on the IC KR1533IE7 base according to the lecturer's task.
4. Design a stopwatch (or timer) circuit from 0 to 9 seconds on the basis of several counters KR1533IE19 or KR1533IE7 and a decoder 533ID18.

7.3 BASIC THEORY

Having connected several trigger circuits – frequency halvers – in series, we can obtain the simplest multibit binary divider. More often frequency dividers are called counters. The basic characteristics of the counters are:

1. Data capacity, which is numerically equal to the counting coefficient. For a binary counter $k_{count}=2^n$, where n – the number of flip-flops in a counter circuit.
2. Speed – maximum pulse repetition frequency at the sync input.

According to the data display all counters fall into two groups – binary and binary-decimal. In binary counters the information is presented in the straight binary code, and in the binary-decimal – straight binary-decimal code. According to the count method the counters are divided into:

- up-counter (addition counter),
- down-counter (subtracting counter),
- up-down counter – act both as down-counter and up-counter.

According to their functioning the counters can be classified as synchronous and asynchronous.

In asynchronous counters the counting pulse is supplied to the first flip-flop clock input, and the following flip-flop is clocked by the previous one. Fig. 7.1 shows the four-bit asynchronous up-counter implementation designed on the flip-flop KR1533TV9 base.

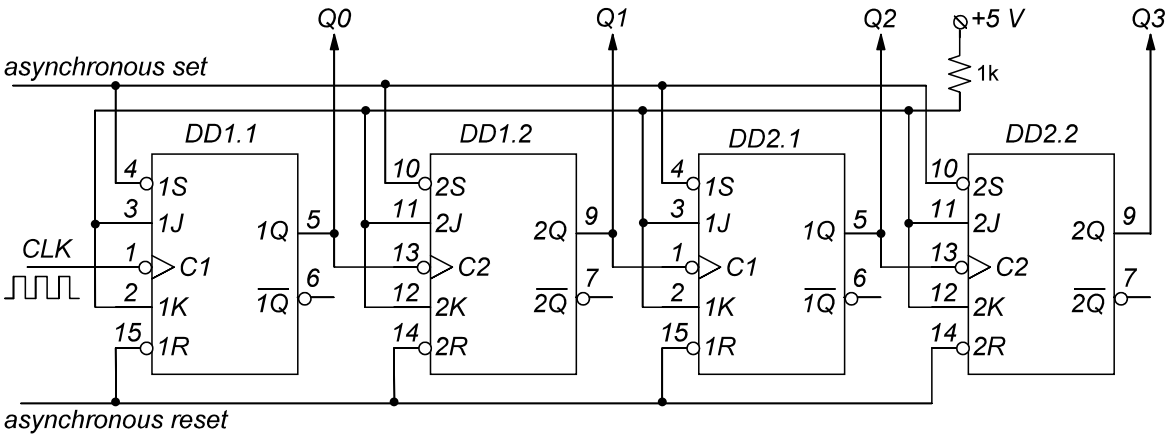


Fig. 7.1. 4-bit asynchronous up-counter
(DD1, DD2 – KR1533TV9)

In synchronous counters count pulses are supplied to all flip-flops clock inputs. The following flip-flop is toggled only when the low-order bits are filled with the ‘ones’ (in case of adding counter), and ‘zeros’ (in case of subtracting counter). Fig. 7.2 shows the four-bit synchronous adding counter implementation cdesigneded on the flip-flop KR1533TV9 base (see also table 7.1).

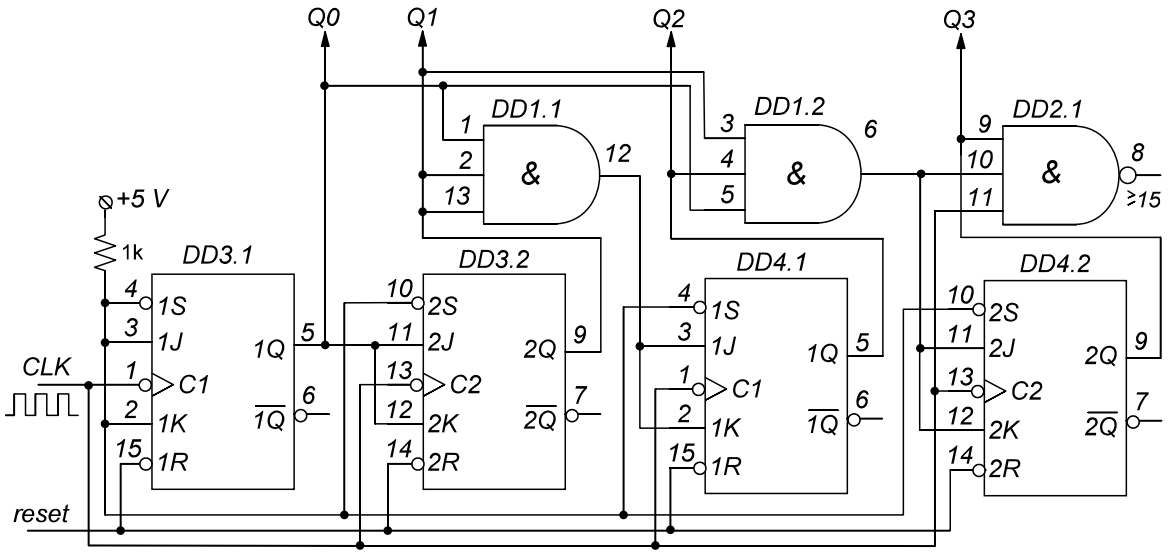


Fig. 7.2. 4-bit synchronous up-counter with AND gates

Table 7.1

ICs for the circuit in Fig. 7.2

IC type Circuitry symbol	KR1533LI3 DD1	KR1533LA4 DD2	KR1533TV9 DD3, DD4
Common	7	7	8
+5 V	14	14	16

For implementing a counter the AND gate in the circuit in Fig. 7.2 needs to be replaced with the NOR gate, and the signals to their inputs should be supplied from the foregoing flip-flop inverted outputs (Fig. 7.3, table 7.2). Thus, the function according to which the flip-flops will be toggled in series is the following: $F = Q_0 \cdot Q_1 \cdot \ldots \cdot Q_{n-1} = \overline{\overline{Q_0} + \overline{Q_1} + \ldots + \overline{Q_{n-1}}}$.

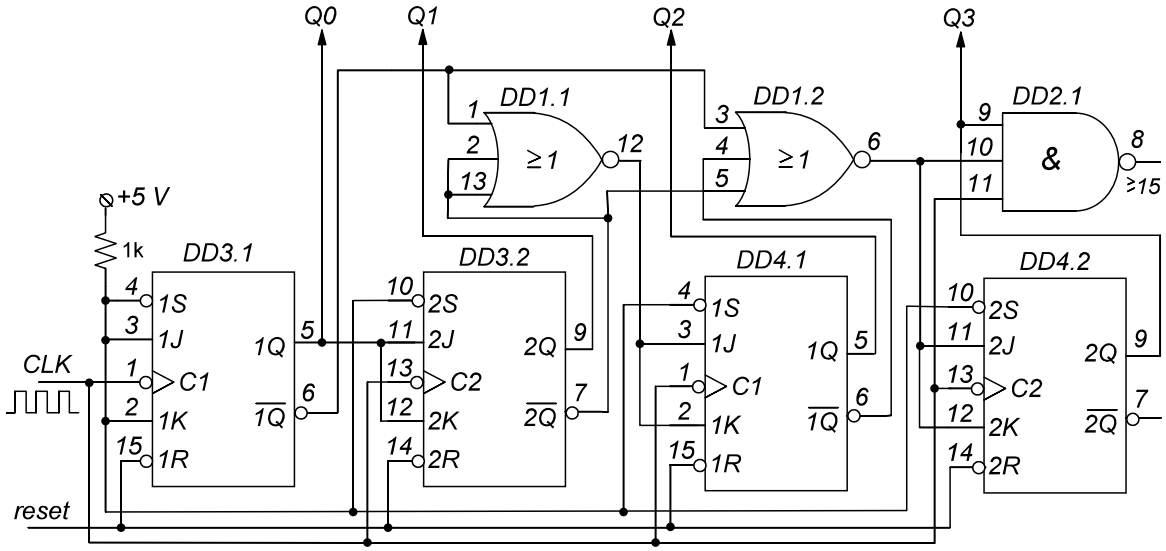


Fig. 7.3. 4-bit synchronous up-counter with NOR gates

Table 7.2

ICs for the circuit in Fig. 7.3

IC type Circuitry symbol	KR1533TV9 DD1, DD2	KR1533LE4 DD3	KR1533LA1 DD4
Common	8	7	7
+5 V	16	14	14

In the lab work it's offered to analyse a universal IC KR1533IE7. It is a four-bit synchronous binary up-down counter with the maximum counting modulo equal to 16. Fig. 7.4 depicts the IC KR1533IE7 logic symbol and pin configuration. The possible counter KR1533IE7 operating modes are given in table 7.3.

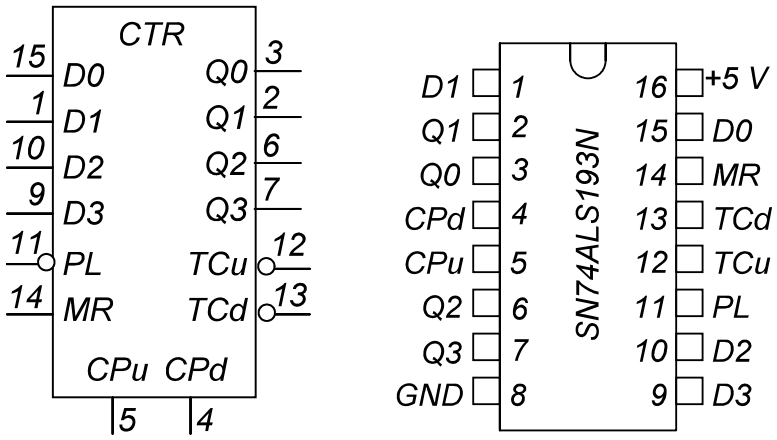


Fig. 7.4. IC KR1533IE7 logic symbol and pin configuration

Table 7.3

Counter KR1533IE7 operating modes

Mode	Inputs								Outputs					
	R	*PE*	*+1*	*–1*	*D0*	*D1*	*D2*	*D3*	*Q0*	*Q1*	*Q2*	*Q3*	*≥15*	*≤0*
Reset	1	x	x	0	x	x	x	x	0	0	0	0	1	0
	1	x	x	1	x	x	x	x	0	0	0	0	1	1
Parallel load	0	0	1	0	0	0	0	0	0	0	0	0	1	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1
	0	0	0	1	1	1	1	1	1	1	1	1	0	1
	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	0	0	x	x	D_0	D_1	D_2	D_3	D_0	D_1	D_2	D_3	1	1
Increment	0	1	↑	1	x	x	x	x	Increment				1	1
Decrement	0	1	1	↑	x	x	x	x	Decrement				1	1

Note. $D_3D_2D_1D_0 \neq 0000_2$, $D_3D_2D_1D_0 \neq 1111_2$

The input *+1* (C_u) and *–1* (C_d) – are the clock inputs for the count increase and decrease, respectively. The counter state changes according to the positive-going edge of the clock pulse, i.e. the switching from 0 to 1, at every clock input. KR1533IE7 is designed on the synchronous counter principle based on the the dynamic flip-flop.

If the input *+1* is supplied with the change of voltage from 0 to 1, the counter is incremented by one. The same change of voltage supplied to *–1* will cause the counter decrement by 1.

If the clock input *+1* is in use, *–1* should be provided with the logic-one level and vice versa. The changing of the count direction should be done when the clock signal stays at the logic-one level.

The input *R* serves for asynchronous reset of all counter bits by supplying logic 1 to the input. To enable count the input *R* is supplied with logic 0.

The outputs *D0, D1, D2* and *D3* serve for preset of any initial number $D_3D_2D_1D_0$ into the counter. The count pulses are summed up with this number (in the adding mode), or the count pulses are subtracted from the number (in the subtracting mode). The data entry takes place when a low-level signal appears at the input $\overline{PE}$. The outputs *Q0, Q1, Q2* and *Q3* are the noninverted outputs of the counter bits.

The output *≥15* (TC_U) is the carry output. The carry signal (low active level) appears when the counter is switched from the $1111_2=15_{10}$ state to the 0000_2 state. So, the carry signal is formed by the negative count pulse edge.

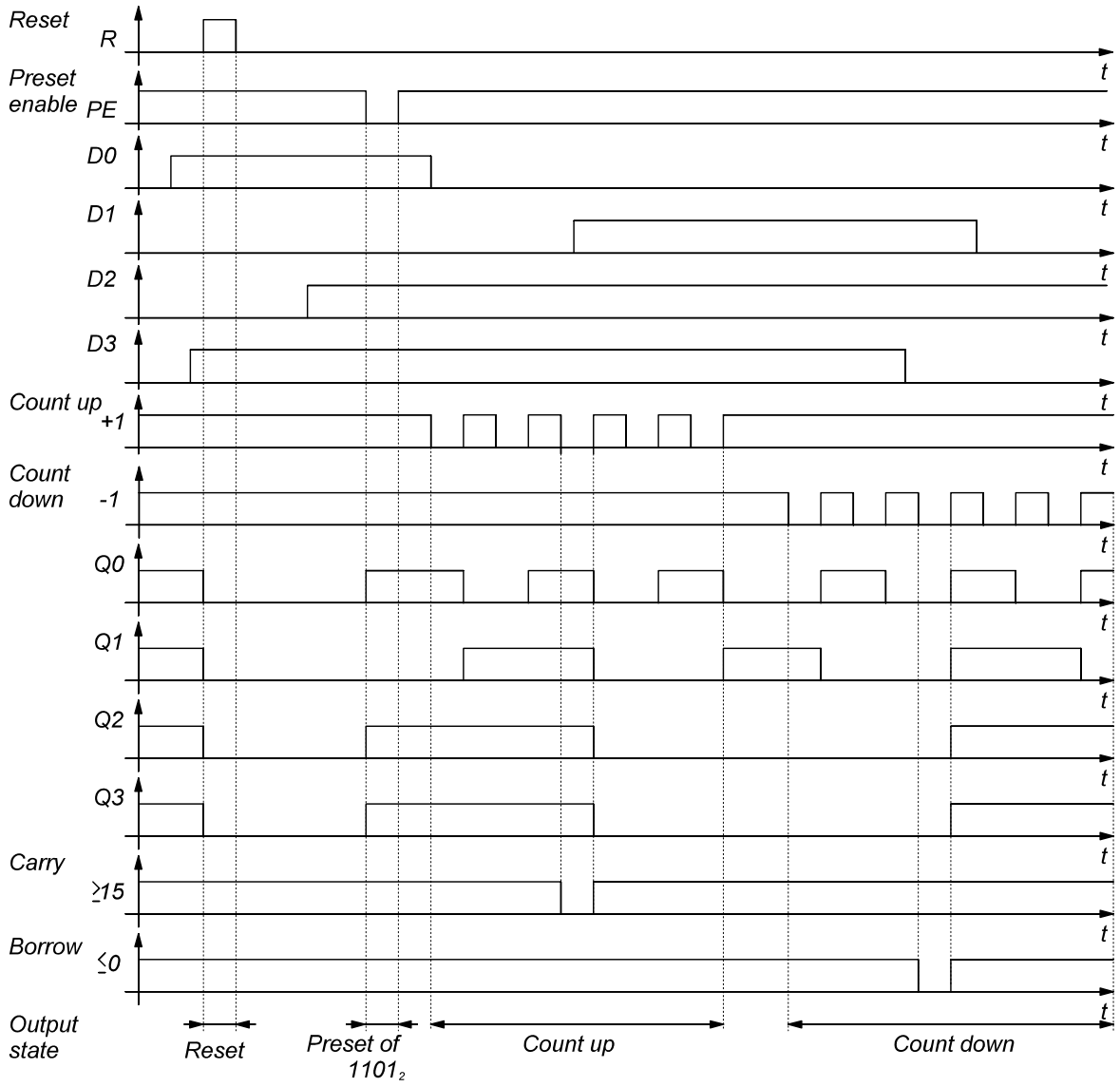


Fig. 7.5. IC KR1533IE7 performance diagram

The output ≤ 0 (TC_D) is the borrow signal output, which emerges when the state 0000_2 is toggled to 1111_2.

The carry and borrow outputs are applied in the process of microcircuits cascading and when the counter operates as a divider. In other words they can be used for periodic data recording into the counter from the inputs *D0, D1, D2* and *D3*. In order to fulfil this you should simply connect the input $\overline{PE}$ with the corresponding output ≥ 15 or ≤ 0. However, if we connect the carry output ≥ 15 with the input $\overline{PE}$, the division coefficient will be $K_{count}=15-N$, where N – is a decimal equivalent of the binary code at the inputs *D0, D1, D2* and *D3*.

Fig. 7.5 shows the counter KR1533IE7 (SN74ALS193N) functioning diagram, which are drawn for the case when the inputs *D0, D1, D2, D3* are supplied with the $1101_2=13_{10}$ code. The series of the input control pulses corresponded to the succession of the operating modes presented in table 7.3.

When the information from the inputs *D0*, *D1*, *D2* and *D3* is recorded into the counter bits and the pulses come to the input *+1*, the counter outputs change their state, beginning with 1101_2 code. If there is no data re-entry into the counter, the overflow happens after 1111_2 and the count will start with 0000_2, etc. The signal at the output ≥ 15 will emerge after the 15^{th} pulse at the input *+1*.

The countdown will be carried out when the signal is sent to the input *–1*.

Fig. 7.6 and 7.7 show the eight-bit counter circuits designed on the IC KR1533IE7 base. They are obtained by means of sequential and parallel methods of capacity increase, respectively.

With the help of the counters it is possible to design a frequency divider with large dividing coefficient. The design concept consists in gradual capacity increase. Fig. 7.8 gives an example of a counter-divider design on the IC KR1533IE19 (SN74ALS393N) base. The frequency F at the circuit output can be defined as:

$$F = \frac{f}{2^n},$$

where n – is a bit, from which a signal is taken.

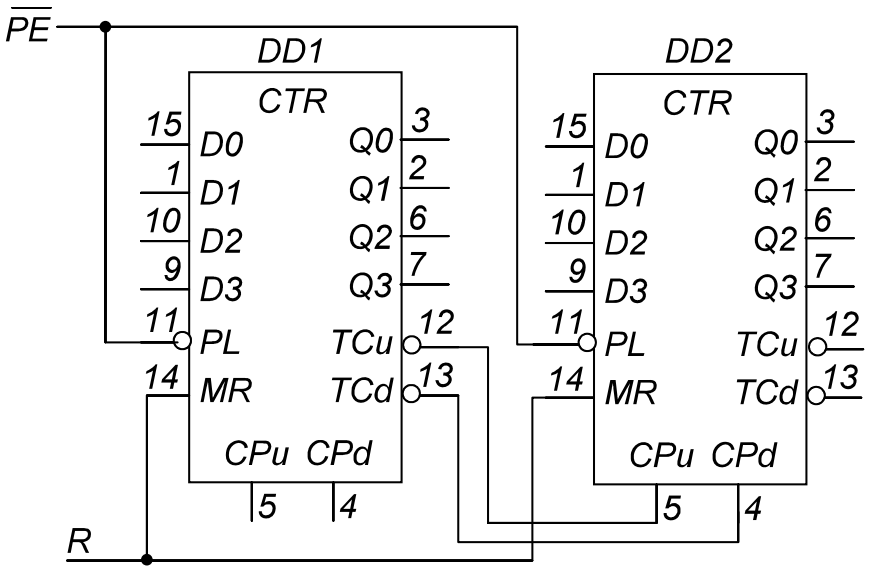


Fig. 7.6. 8-bit counter on the IC KR1533IE7 base (serial method of capacity increase)

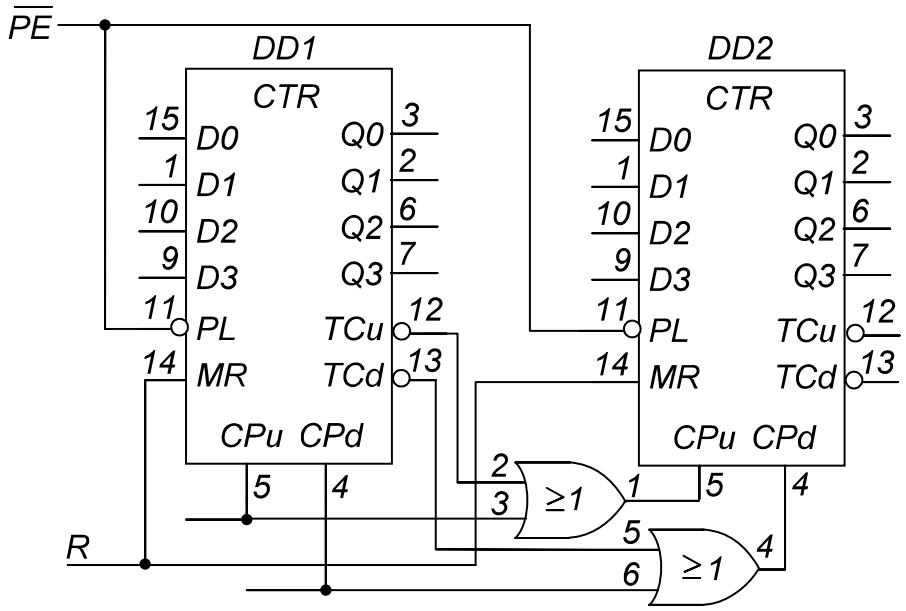


Fig. 7.7. 8-bit counter on the IC KR1533IE7 base (parallel method of capacity increase)

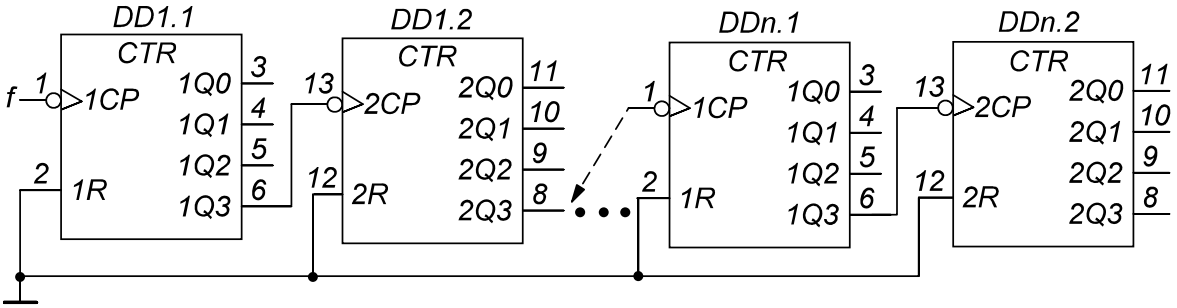


Fig. 7.8. Counter-divider based on the IC KR1533IE19

7.4. EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4.

In order to carry out the lab work you'll need a set of microcircuits: KR1533IE7, KR1533IE19, KR1533TV9, KR1533LA3, KR1533LA4, KR1533LA1, and KR1533LE4.

7.5 IN-LAB TASKS

1. Verify the counter performance for summing and subtracting functioning. For each mode provide the required logic levels at the counter inputs and take waveforms of the voltage at the clock input and all the outputs including ≥ 15 and ≤ 0. The count pulses should be supplied to the counter inputs $+1$ and -1 from the built-in pulse generator. If required, divide clock frequency using several KR1533IE7 or KR1533IE19.

2. Design and implement the counter-divider based on the identification and reset method. The k_{count} is given by the lecturer. Take the waveforms of the voltage at the counter-divider inputs and outputs.

3. Design and implement the counter-divider with the k_{count} based on presetting method according to the lecturer's task. Take the waveforms of the voltage at the counter-divider inputs and outputs, including the carry (borrow) output. The operating mode ($+1$ or -1) is given by the lecturer.

4. Increase the counter capacity up to 8 by series increase method. According to the lecturer's task implement the counter with the k_{count} more than 16 by any way you like.

5. Assemble the asynchronous counter circuit on the flip-flop KR1533TV9 base (Fig. 7.1), take the performance diagrams.

6. Construct the synchronous counter circuit on the flip-flop KR1533TV9 base (Fig. 7.3), take the performance diagrams, and compare with the results from paragraph 5.

7.6 ADDITIONAL TASKS

1. Measure the laboratory bench clock generator frequency; calculate the required division ratio for obtaining the frequency 1 Hz.

2. Design the circuit of by-turn lighting of two LEDs using several KR1533IE19 or KR1533IE7 counters.

3. Design the stopwatch (or timer) circuit from 0 to 9 seconds with the help of the available counters and a decoder 533ID18. The time should be represented on the seven-segment display.

7.7 QUESTIONS

1. Give the definition of ‘a counter’ and ‘a counter-divider’.
2. What types of flip-flops can be applied for counter design?
3. What is the difference between synchronous and asynchronous counters?
4. What are the operational characteristics of the parallel-carry counters?
5. What is the function of the counter data inputs (e.g. KR1533IE7)?
6. What flip-flops are considered to be basic in counter KR1533IE7 circuit design?
7. What is the difference between series and parallel methods of counter capacity increase?

Lab 8

SHIFT REGISTER AND PSEUDORANDOM SEQUENCE GENERATOR SYNTHESIS ON THE D FLIP-FLOP BASE

8.1 OBJECTIVES

The aim of the lab work is to master the skills of shift register synthesis on the basis of elementary memory unit cells – flip-flops and the use of registers for code converter design.

8.2 PRE-TASKS

1. Study the purpose, functions and basic circuitry of shift registers.
2. Study the principle of converting information in the serial code to the parallel, and vice versa.
3. Acquaint with the operating principle and the output functionality of the D flip-flops KR1533TM8 and KR1533TM2 microcircuits.
4. Design the four-bit shift register circuit based on the IC KR1533TM2 with the asynchronous data upload ability.
5. Implement the counting coefficient $K_{count}=4\cdot10^6$ with use of KR1533IE19 counter.

8.3 BASIC THEORY

In order to build a register we should use synchronous flip-flops. They can be switched only if there is a sync pulse at the clock input. For this purpose D flip-flops, having the data input *D* and the dynamic clock input *C* are widely utilized (Fig. 8.1). As a rule, flip-flops also have the asynchronous reset input *R*, which can be both non inverse and inverse.

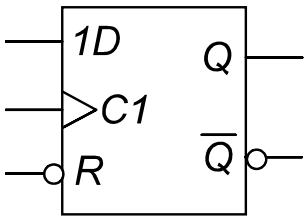


Fig. 8.1. D flip-flop

The device called *'register'* serves mainly for temporary storage of numbers in the binary code when carrying out various arithmetic and logic operations. With the help of registers we can also perform such operations as:

to transfer numbers from one device to the other, to shift to the low-order or high-order bit, to convert the serial code to the parallel one, and vice versa.

In the lab work you are supposed to implement a four-bit shift register on the IC KR1533TM8 base.

The IC KR1533TM8 (SN74ALS175N) is placed in DIP16 package (16 pins) and includes four D flip-flops (Fig. 8.2), which have a common asynchronous reset input $\overline{R}$ and a clock input C (negative dynamic input). Fig. 8.3 shows the IC KR1533TM8 logic symbol and pin configuration.

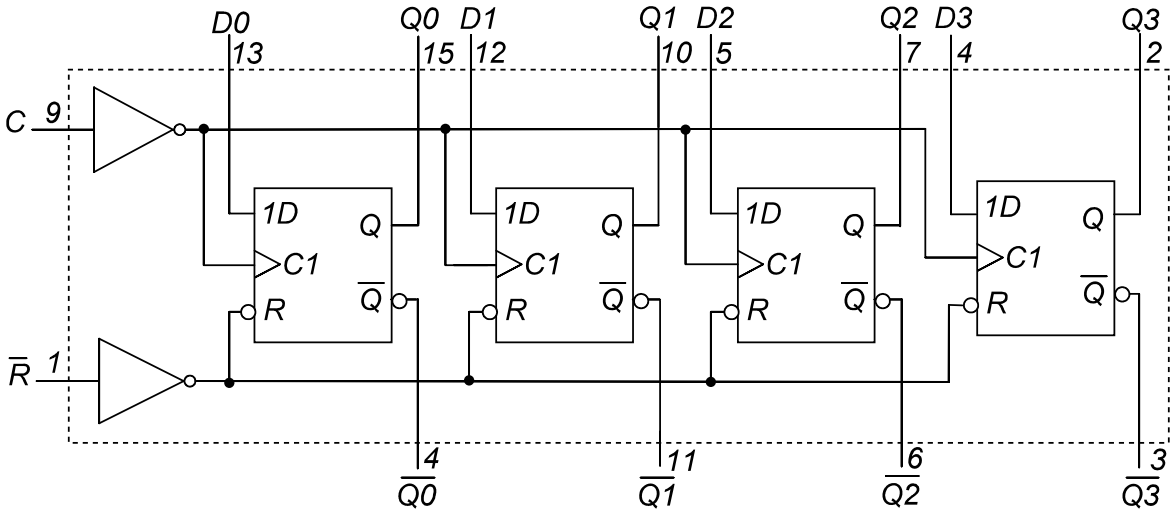


Fig. 8.2. IC KR1533TM8 schematic diagram

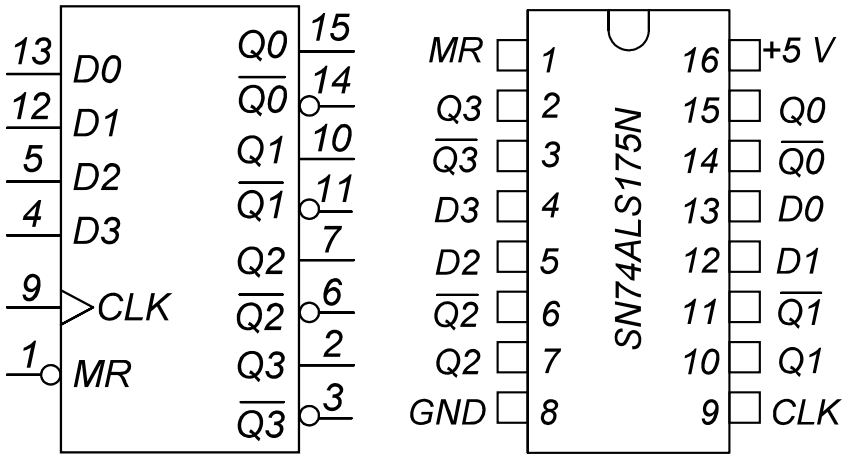


Fig. 8.3. IC KR1533TM8 logic symbol and pin configuration

The functioning of one flip-flop of the IC KR1533TM8 is presented in table 8.1. Every flip-flop has the output Q and $\overline{Q}$. The reset of all the flip-flops in the $Q_n=0$ state occurs when the asynchronous reset input $\overline{R}$ is supplied with the low voltage. When $\overline{R} = 0$, the inputs C and D_n ($D1$–$D4$) are

out of order, their state makes no difference (x). The information from the parallel data input *D1–D4* can be provided to the flip-flops circuits, if the input $\overline{R}$ is supplied with high voltage, and the clock input *C* – pulse positive-edge. The logic levels (0 or 1) preset at every input *D* will emerge at the corresponded output *Q*.

Table 8.1

KR1533TM8 state table

Operating mode	Inputs			Outputs	
	$\overline{R}$	*C*	D_n	Q_n	$\overline{Q}_n$
Reset	0	x	x	0	1
Load 1	1	↑	1	1	0
Load 0	1	↑	0	0	1

Some D flip-flops in addition to the asynchronous reset input can be equipped with the asynchronous set input, e.g. a flip-flop KR1533TM2 (Fig. 8.4). The microcircuit includes two D flip-flops, which are controlled by the positive edge of the clock pulse. A flip-flop can be set and reset asynchronously and independently of each other. Table 8.2 presents the operation of one section of the IC KR1533TM2.

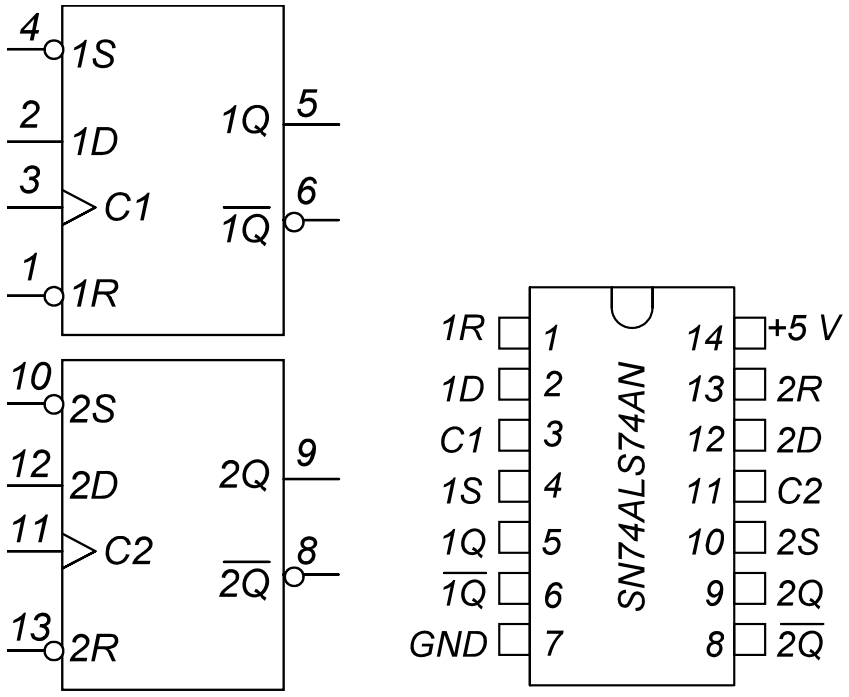


Fig. 8.4. IC KR1533TM2 logic symbol and pin configuration

Table 8.2

KR1533TM2 status table

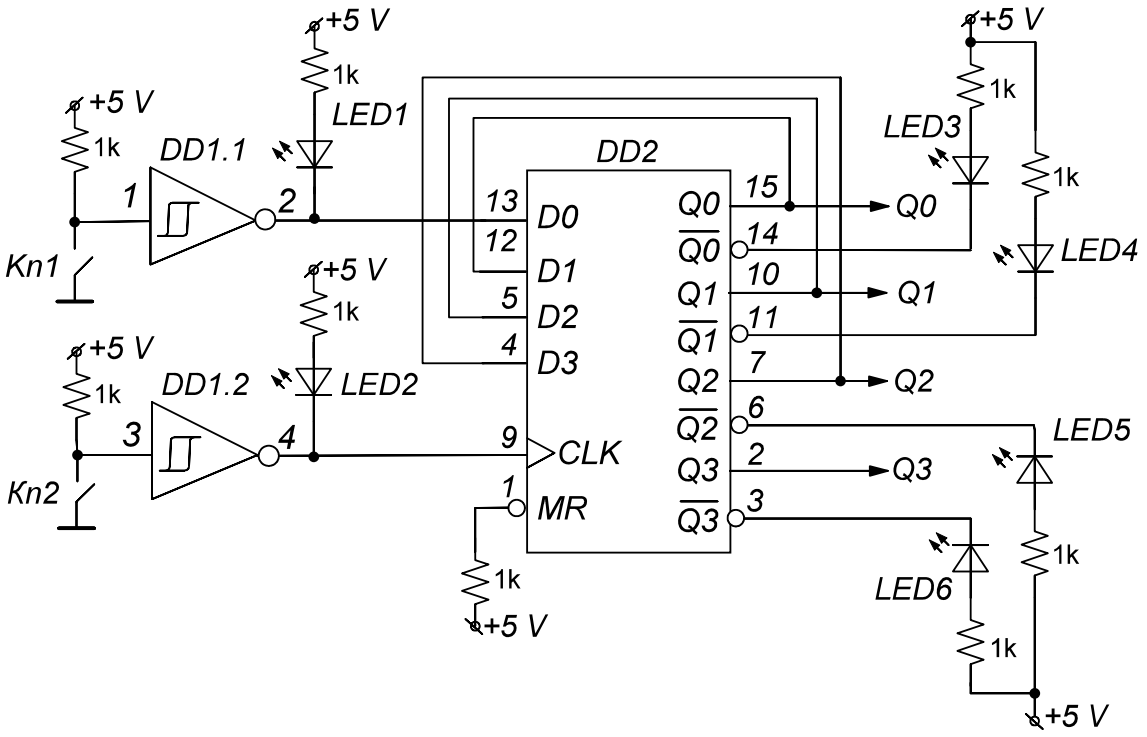
Inputs				Outputs	
$\overline{R}$	$\overline{S}$	C	D	Q	$\overline{Q}$
0	1	x	x	1	0
1	0	x	x	0	1
0	0	x	x	-	-
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	x	*Q0*	$\overline{Q0}$

Fig. 8.5 (see also table 8.3) shows the four-bit shift register circuit with the serial information data input which is based on the IC KR1533TM8. It also depicts the implementation of the shift register operation using available laboratory equipment. Fig. 8.6 shows the voltage diagrams at the flip-flop outputs. They will help you to understand the shift register operating principle.

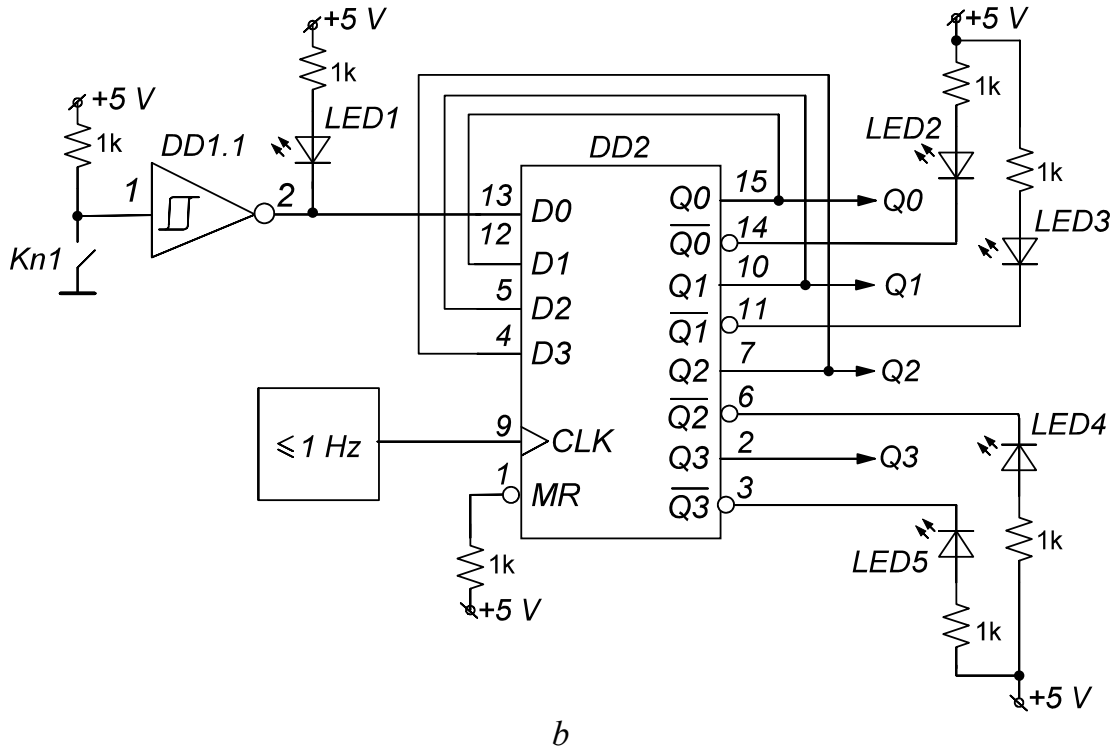
In order to write the data in the low-order bit *Q0*, at first it is needed to form them at the input *D0*, and then to generate a sync pulse at the input *C*. On its positive-going edge the data from the input *D0* will be latched and transmitted to the *Q0* output. The voltage, corresponding to the logic 1 and logic 0 at the input *D0* and *C*, is formed with the help of the KR1533TV9 flip-flop. For writing 1 you should push the button 1 (*Kn1*) at first, and keeping on pushing it, click the button *Kn2*. For writing 0 you can simply click the button *Kn2*. The voltage levels at the inputs *D0*, *C*, *Q0–Q3* are advised to be controlled with the help of the LEDs (VD1–VD6). For the circuit shown in Fig. 8.5, *a*, the glowing LED at the KR1533TV9 output corresponds to $DI = 0$ or $C = 0$, and at the KR1533TM8 output $-Q_n = 1$, $\overline{Q}_n = 0$ because they are connected to the inverted outputs.

You should bear in your mind that if the sync signal is formed in such manner, the circuit operation is influenced by the possible contact bounce of the button *Kn2*, and the data shift can occur several times even if pressed only once. Another variant of shift implementation is more preferable (Fig. 8.5, *b*). The input *D0* should be connected to the driver with one or two buttons. The driver is based on the KR1533TV9 (Fig. 6.9) or KR1533TL2 (Fig. 6.8). The input *C* should be supplied with the pulses of frequency ≤ 1 Hz. The clock frequency is chosen in such a way that there is enough time for forming the input *D0* value by hand, visual fixing of the LED lighting and recording the results of the experiment. The frequency divider is

designed by series connection of the same type counters (see the circuit in Fig. 7.8).



a



b

Fig. 8.5. 4-bit shift register with the serial data input based on the IC KR1533TM8

Table 8.3

ICs in Fig. 8.5

IC type Circuitry symbol	KR1533TL2 DD1	KR1533TM8 DD2
Common	8	8
+5 V	16	16

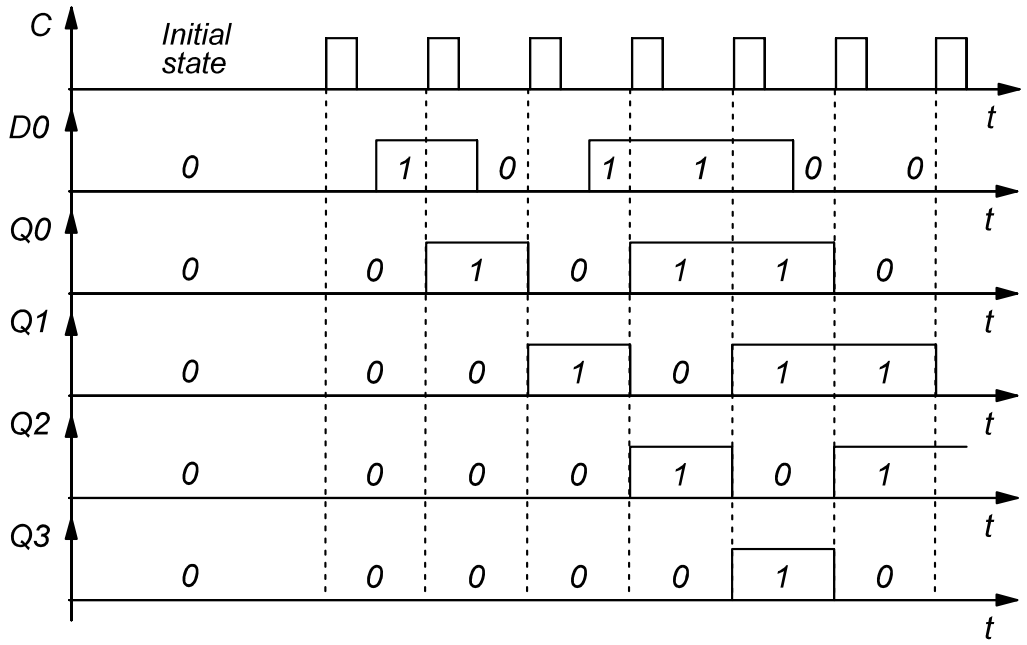


Fig. 8.6. Voltage diagrams for the circuit in Fig. 8.5

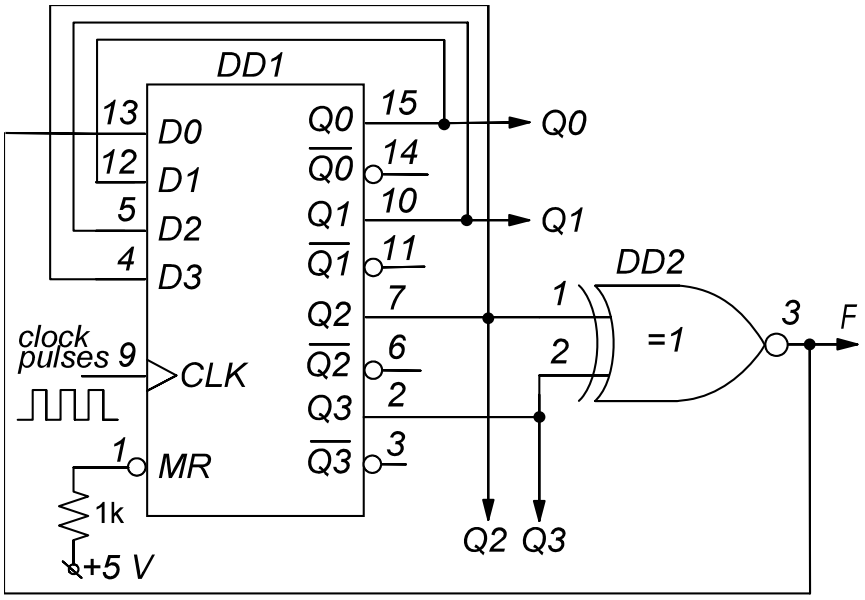


Fig. 8.7. Pseudorandom pulse sequence generator circuit

Table 8.4

ICs in Fig. 8.7

IC type Circuitry symbol	KR1533TM8 DD2	KR1533LP5 DD2
Common	8	7
+5 V	16	14

In the lab work you will also have to implement a pseudorandom pulse sequence generator on the IC KR1533TM8 base and supplementary logic 'XOR'. Fig. 8.7 shows the generator circuit (see also table 8.4).

Fig. 8.8 shows an 8-bit ring counter. It is formed by supplying reset signal *R* to all D flip-flops from one of the shift register outputs. The counting coefficient in the circuit in Fig. 8.8 is equal to 6.

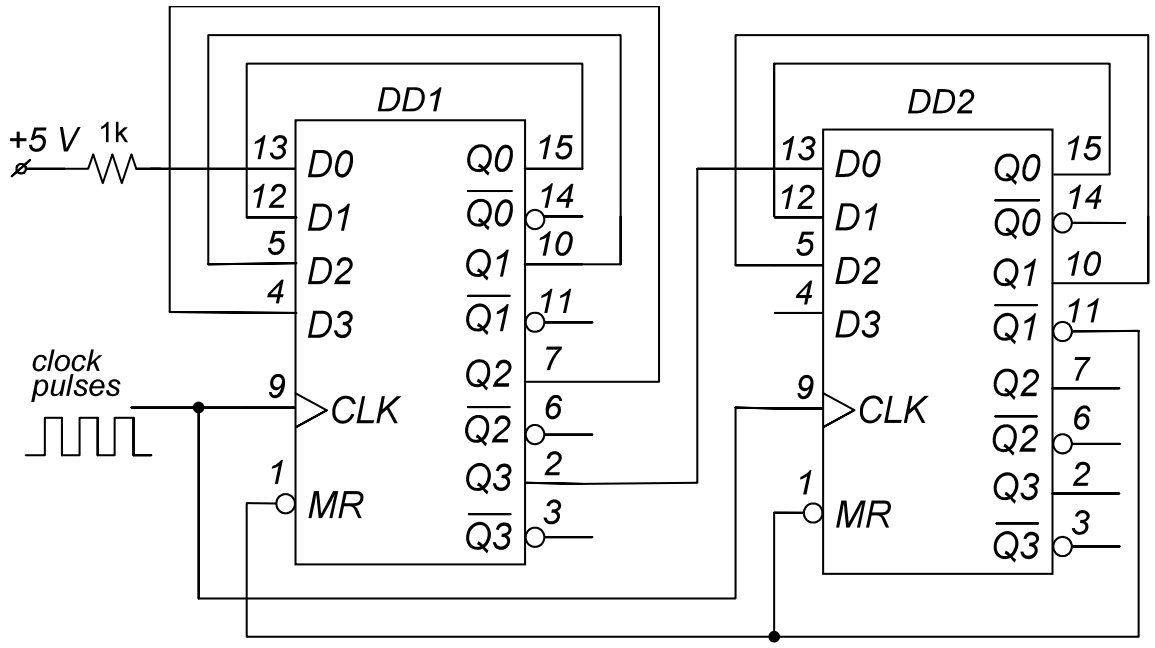


Fig. 8.8. Ring counter (DD1, DD2 – KR1533TM8)

With the help of registers the conversion of the parallel binary code to the serial binary code can be achieved. Fig. 8.9 (see also table 8.5) depicts the circuit representing this process by means of D flip-flops – IC KR1533TM8.

The signal *V* chooses the register operating mode and form appropriate inter-digit connections. When *V*=0 (writing), across the sync pulse *C* positive edge at the flip-flop outputs *Q3*, *Q2*, *Q1* and *Q0* the signal levels emerge. These signals were at that time at the inputs *D3*, *D2*, *D1* and *D0*, respectively, i.e. the parallel data upload takes place. The circuit operates as a memory register now.

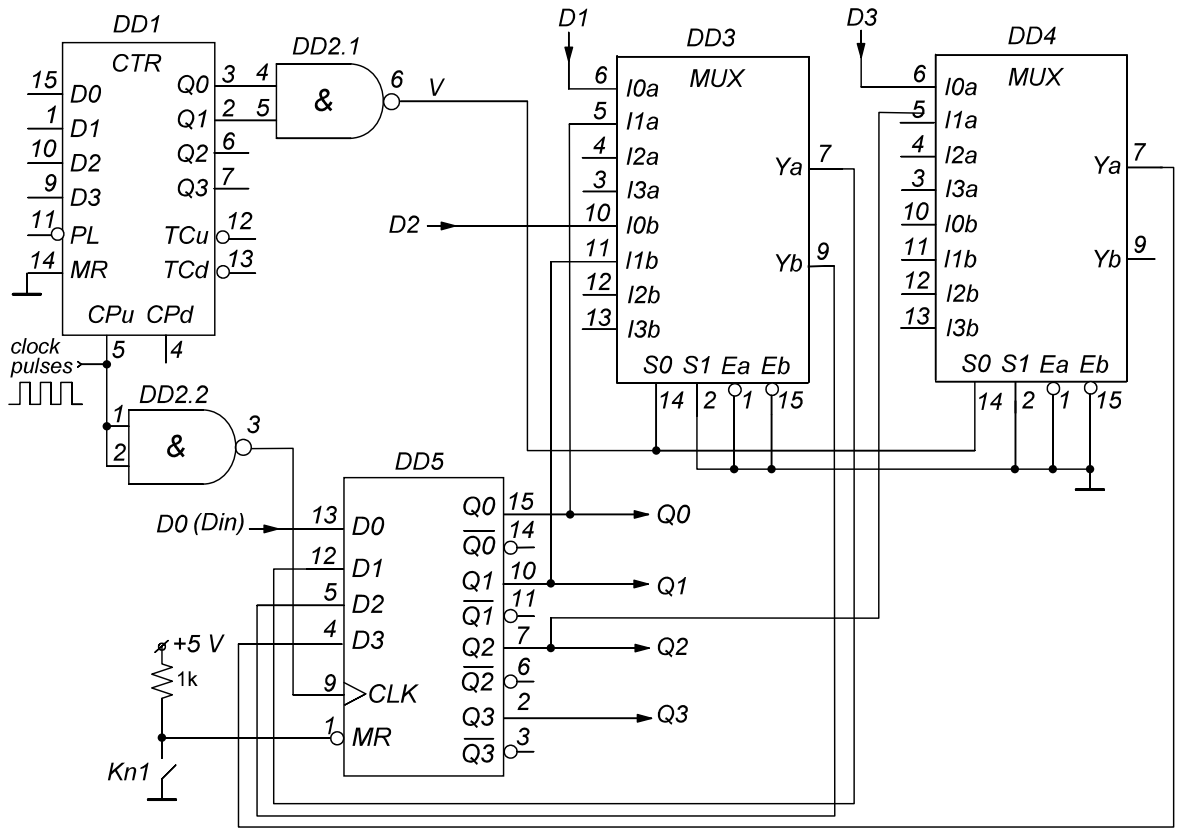


Fig. 8.9. The circuit of parallel binary code conversion to the serial one

Table 8.4

ICs in Fig. 8.9

IC type Circuitry symbol	KR1533IE7 DD1	KR1533LA3 DD2	KR1533KP2 DD3, DD4	KR1533TM8 DD5
Common	8	7	8	8
+5 V	16	14	16	16

If V=1, the successive shift of the data stored in D flip-flops occurs. The output data do not depend on *D3*, *D2* and *D1*. If the input of the first flip-flop D_{input} is supplied with the serial data, the circuit converts the serial code to the parallel one in the same way as discussed earlier. The data storage, writing and reading in the series form can also be done by the circuit.

To make the circuit convert the parallel binary code to the serial one, at first you should write the input parallel data *D3D2D1D0* to the corresponding flip-flops, and then execute the successive shift. The information from *Q3*, *Q2*, *Q1* and *Q0*, i.e. *D3*, *D2*, *D1* and *D0* will occur gradually at the output D_{out} by the every positive edge of clock pulse. The first signal to appear is *D3*, and the last – *D0*. Fig. 8.10 shows the performance diagrams of the circuit, presented in Fig. 8.9.

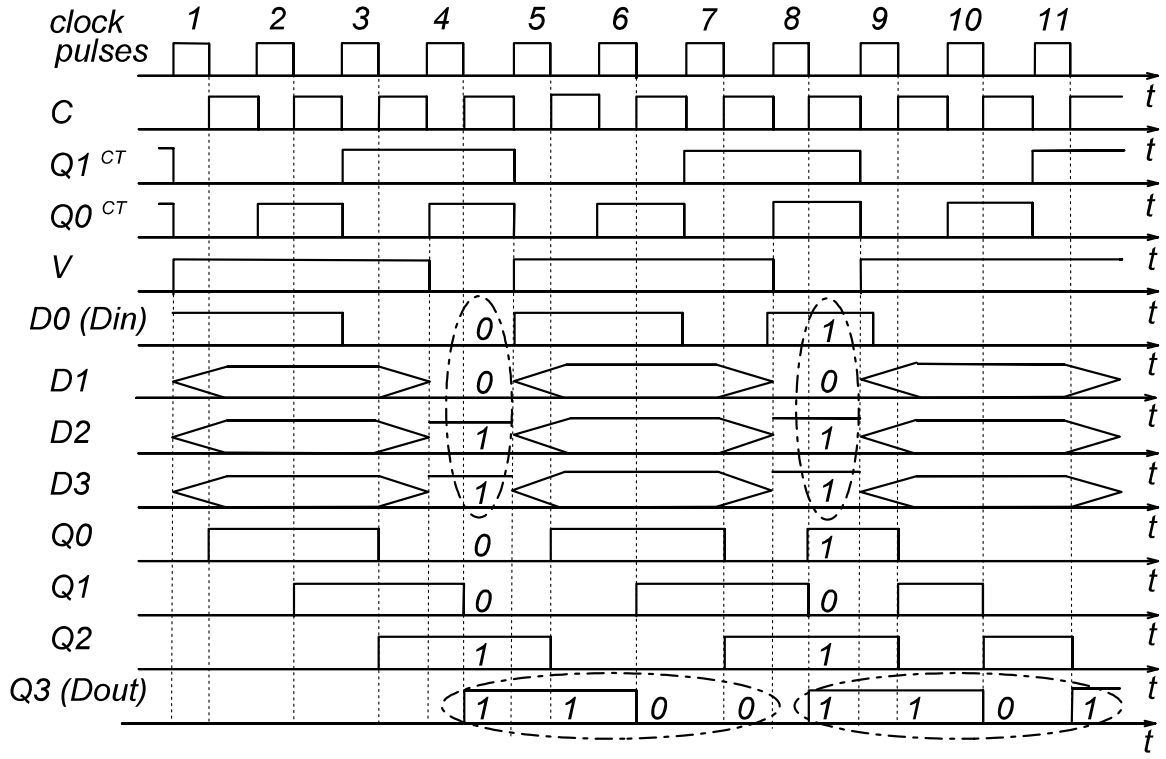


Fig. 8.10. Performance diagrams of the circuit in Fig. 8.9.

8.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4. In the lab work the emphasis is upon the IC KR1533TM8 analysis, which is the base for the implementation of shift register circuit and pseudorandom pulse sequence generator.

In order to carry out the laboratory work you'll need a set of microcircuits: KR1533TM8, KR1533TM2, KR1533KP2, KR1533LA3, KR1533LP5 and KR1533IE7. To register the signals waveforms the double-channel oscilloscope is used.

8.5 IN-LAB TASKS

1. Implement the 4-bit shift register on the IC KR1533TM8 base (Fig. 8.5); verify its operation accuracy by uploading the required information and shifting it to the right.

2. By means of the 4-bit shift register based on the IC KR1533TM8 and supplementary logic 'XOR' implement pseudorandom pulse sequence generator (Fig. 8.7). Remember that the code 0000_2 blocks the generator functioning.

3. Verify the generator functioning in the continuous mode using the built-in pulse pulser and frequency divider as clock signals. Determine the generator operation cycle and characterise the time sequence in the form of a diagram or a truth table.

4. Assemble the ring counter on the IC KR1533TM8 base (Fig. 8.8). The counting coefficient is given by the lecturer.

5. Assemble and study the circuit of parallel binary code conversion to the serial one on the IC KR1533TM8 base (Fig. 8.9).

8.6. QUESTIONS

1. What are the functions of shift registers? Memory registers?
2. How many flip-flops are required for designing a 12-bit shift register?
3. How can the memory register capacity be increased?
4. How can the shift register capacity be increased?
5. Which data processing method is faster: parallel or series one?
6. What is the purpose of the element DD2.2 in the circuit in Fig. 8.9.

Lab 9

SHIFT REGISTER INTEGRATED CIRCUITS STUDY AND CIRCUIT DESIGN ON THEIR BASIS

9.1 OBJECTIVES

The aim of the lab work is to study the functional power of shift register ICs and to develop problem-solving skills of digital technique applying nonconventional methods.

9.2 PRE-TASKS

1. Study the functioning principles of register ICs KR555IR8 (74LS164), KR1533IR10 (SN74ALS166N) and K531IR11 (SN74S194).
2. Get to know the register capacity increase principle.
3. Study the principle of the reverse shift register design.
4. Study the method of the circuit design for serial data transfer with the help of the shift register.

9.3 BASIC THEORY

Shift registers, as already noted in the Lab work 8, represent a group of flip-flops connected in series. Their basic operating mode is the shift of code bits, which are stored in the flip-flops, i.e. with the clock signal the content of every preceding flip-flop is written in the next flip-flop. The code, stored in the register is shifted by one bit to the low-order or high-order bit region with every clock period. That is why they are called so.

There is often some mix-up of the name of shift direction in shift registers. Shift can be of two types: to the right (the basic mode which all shift registers possess) and to the left (the mode which only some reverse shift registers possess). The names reflect the shift register internal structure and signal rewriting sequentially through the flip-flops chain. It is natural for the flip-flops to be numbered from left to right, e.g., from 0 to 7 (or from 1 to 8) for 8-bit registers. Eventually, the data shift by the register to the right is the shift to the region with high-order bits. The data shift by the register to the left is the shift to the region with low-order bits. The shift direction is marked by the arrow at the register logic symbol.

However, it is well known that in any binary number the high-order bits are situated to the left, and low-order bits – to the right. Therefore, in computer and microprocessor technique the shift of a binary number to the

right is considered to be the shift to the low-order bit region, and the shift to the left – the shift to the high-order bit region.

With the help of registers we can multiply and divide numbers by two. The multiplication of the stored number by two is performed by shifting to the high-order bit region and recording 0 to the low-order bit. The integer division of the stored number by two is performed by shifting to the low-order bit region and recording 0 to the high-order bit.

The standard family of digital ICs includes several types of shift registers which differ in operating modes, writing, reading and shifting modes. The majority of shift registers have eight bits. In the lab work you are suggested to examine the IC KR555IR8 functioning. The IC logic symbol and pin configuration are shown in Fig. 9.1. The register KR555IR8 state table is represented in table 9.1.

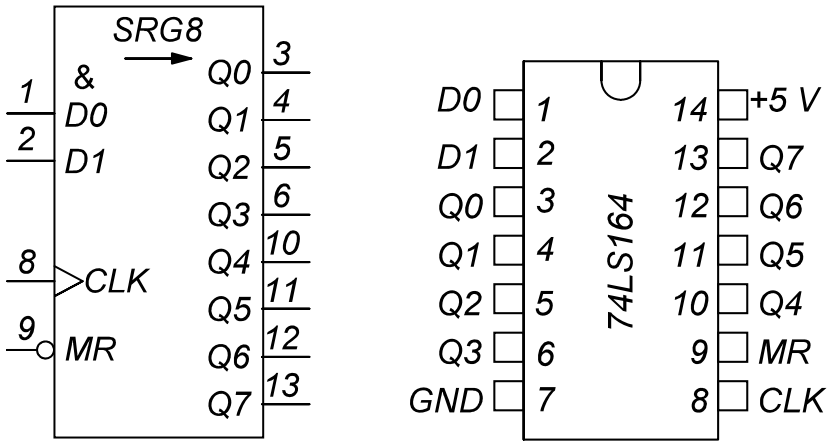


Fig. 9.1. IC KR555IR8 logic symbol and pin configuration

Table 9.1

Register KR555IR8 state table

Inputs				Outputs			
R	*C*	*&*	*D*	*Q0*	*Q1*	...	*Q7*
0	x	x	x	0	0	...	0
1	0	x	x	No change			
1	1	x	x	No change			
1	0→1	1	1	1	*Q0*	...	*Q6*
1	0→1	0	x	0	*Q0*	...	*Q6*
1	0→1	x	0	0	*Q0*	...	*Q6*

The register KR555IR8 is one of the simplest shift registers. This is an 8-bit shift register. It has the clock input *C* for supplying shift pulses, the reset input *R*, two equal data inputs *D1* and *D2* for providing the data

shifting, joined by AND circuit, and eight outputs. The register flip-flops are reset when the logic 0 is supplied to the input *R*. The data reception from the internal inputs $D_{n+1}=Q_n$ and its shift to the right (to the region with high-order bits) occurs across the pulse positive edge at the input *C*.

Fig. 9.2 shows the register KR555IR8 capacity increase circuit. By means of series connection of two 8-bit ICs we obtain the 16-bit shift register. Moreover, the capacity increase doesn't lead to the increase of shift delay, because the clock inputs of the utilized registers are connected in series. The serial input code is converted to the 16-bit output parallel code. In the same way the greater number of microcircuits can be combined.

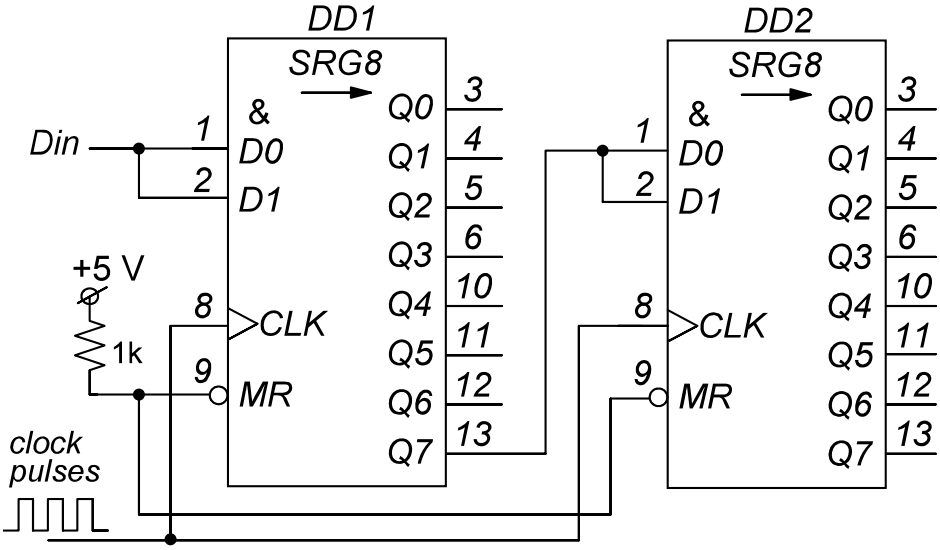


Fig. 9.2. Registers connection for achieving double capacity increase (DD1, DD2 – KR555IR8)

In the lab work it is offered to design a Johnson ring counter on the closed-loop shift register base. Johnson counters are widely applied in automatic equipment as pulse distributors. Their counting coefficient is twice more than the number of flip-flops comprising it. Fig. 9.3 presents a counter constructed on the KR555IR8 base (see also table 9.2). It consists of eight flip-flops, i.e. they will have 16 stable states. Table 9.4 shows the circuit functioning (*N* – count pulses).

Table 9.2

ICs in the circuit in Fig. 9.3

IC type Circuitry symbol	KR555IR8 DD1	KR1533LA3 DD2
Common	7	7
+5 V	14	14

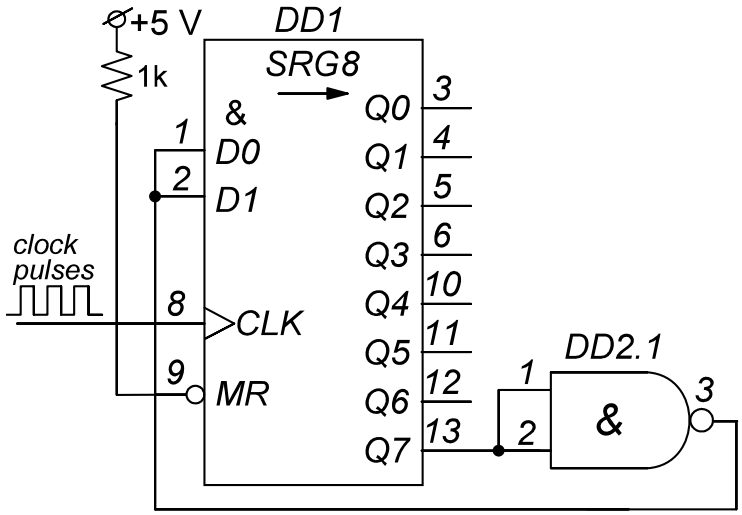


Fig. 9.3. Johnson counter on the IC KR555IR8 base

Table 9.3

The state table of the circuit in Fig. 9.3

N	*Q0*	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0
6	1	1	1	1	1	1	0	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1
11	0	0	0	1	1	1	1	1
12	0	0	0	0	1	1	1	1
13	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	1	1
15	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0

Shift registers are used mainly for converting the parallel code into the series one, and vice versa. Such converting is essential, e.g., when transferring information over long distances (in information networks), when recording the data onto magnetic carrier, when dealing with TV-monitors and camcoders, and also for many other purposes.

In order to implement a serial data transfer circuit by means of shift registers the registers KR555IR8 and KR1533IR10 are used in the lab work.

The microcircuit KR1533IR10 logic symbol and pin configuration are shown in Fig. 9.4. The register KR1533IR10 operating modes are given in table 9.4.

The register KR1533IR10 fulfils the function opposite to that of the register KR555IR8. KR1533IR10 converts the input parallel binary code to the output series one. However, the essense of shifting does not change. In KR1533IR10 all inner flip-flops have led out parallel inputs, and only the last flip-flop has a led output. The data writing and shift is carried out on the pulse positive edge at one of the clock inputs *C1* or *C2*, which are joined by the 2OR function. Thus, the input signal $\overline{WR}$ makes the choice: $\overline{WR}=0$ – writing, $\overline{WR}=1$ – shift. There is also an extension input *DR*. The signal from this input in the shift mode is written to the low-order bit of the shift register.

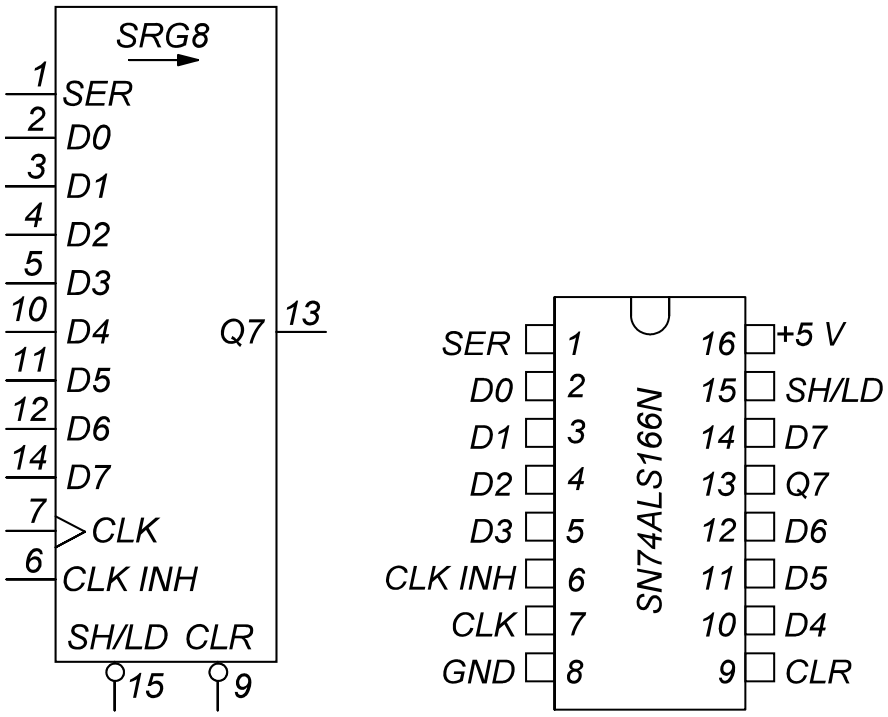


Fig. 9.4. IC KR1533IR10 logic symbol and pin configuration

Table 9.4

KR1533IR10 operating modes

Inputs			Function
R	*WR*	$C1 \vee C2$	
1	0	↑	Parallel upload
1	x	0	Storage
1	x	1	
1	1	↑	Shift
0	x	x	Asynchronous reset

Fig. 9.5 (see also table 9.5) introduces two circuits of digital data transceiving in a serial code in two lines: data line and clock-line. In comparison with the parallel data transfer, this type of transfer let us reduce the number of connecting wires, and simplify the protection of transmitted data from external electromagnetic interference, but the transfer speed will be lower in this case. The difference between the circuits in Fig. 9.5, *a* и 9.5, *b* is the following: in the first case the data from the register DD1 input *DI* are uploaded only once when the button *Kn1* is pressed, in the second case – the parallel data are loaded automatically at every eight clock pulse.

At the transmitting end (at left in Fig. 9.5) with the help of the shift register KR1533IR10 the 8-bit parallel input code is converted to the data bit sequence having the clock frequency. At the receiving end (at right in Fig. 9.5) with the help of the shift register KR555IR8 the data sequence is converted again to the parallel code. Both registers are clocked by the same clock signal, which is transferred in the connection line in parallel with the data sequence. In order to increase the reliability of the transfer the clock signal for the receiver part needs to be delayed with the help of two (or more) inverters.

The first bit of the input parallel code of the register KR1533IR10 input *DI7* starts to be transfered when the data is being uploaded to the register ($\overline{WR} = 0$). Next bits are transferred with every succeeding positive edge of the clock signal at the inputs *C1* and *C2*. The last signal is transferred from the input *DI0*. While pushing the button *Kn1* for a long time the serial code at the output DD1 represent a number of identical values equal to *DI7*.

The serial code bits are uploaded to the register KR555IR8 in the order they were in the register KR1533IR10 after the shifting started (signal withdrawal $\overline{WR} = 0$). When the transfer is over, the first transferred data bit occurs in the digit position *Q7* of the register KR555IR8, and the last transferred data bit – in the position *Q0*.

One more IC which is suggested to be studied in the lab work is the IC of the shift register K531IR11. The IC K531IR11 logic symbol and pin configuration is shown in Fig. 9.6. Table 9.6 presents the role of microcircuit outputs.

Table 9.5

ICs in the circuit in Fig. 9.5

IC type Symbol	KR1533IR10 DD1	KR555IR8 DD2	KR1533LN1 DD3	KR1533IE19 DD4	KR1533LA4 DD5
Common	8	7	7	7	7
+5 V	16	14	14	14	14

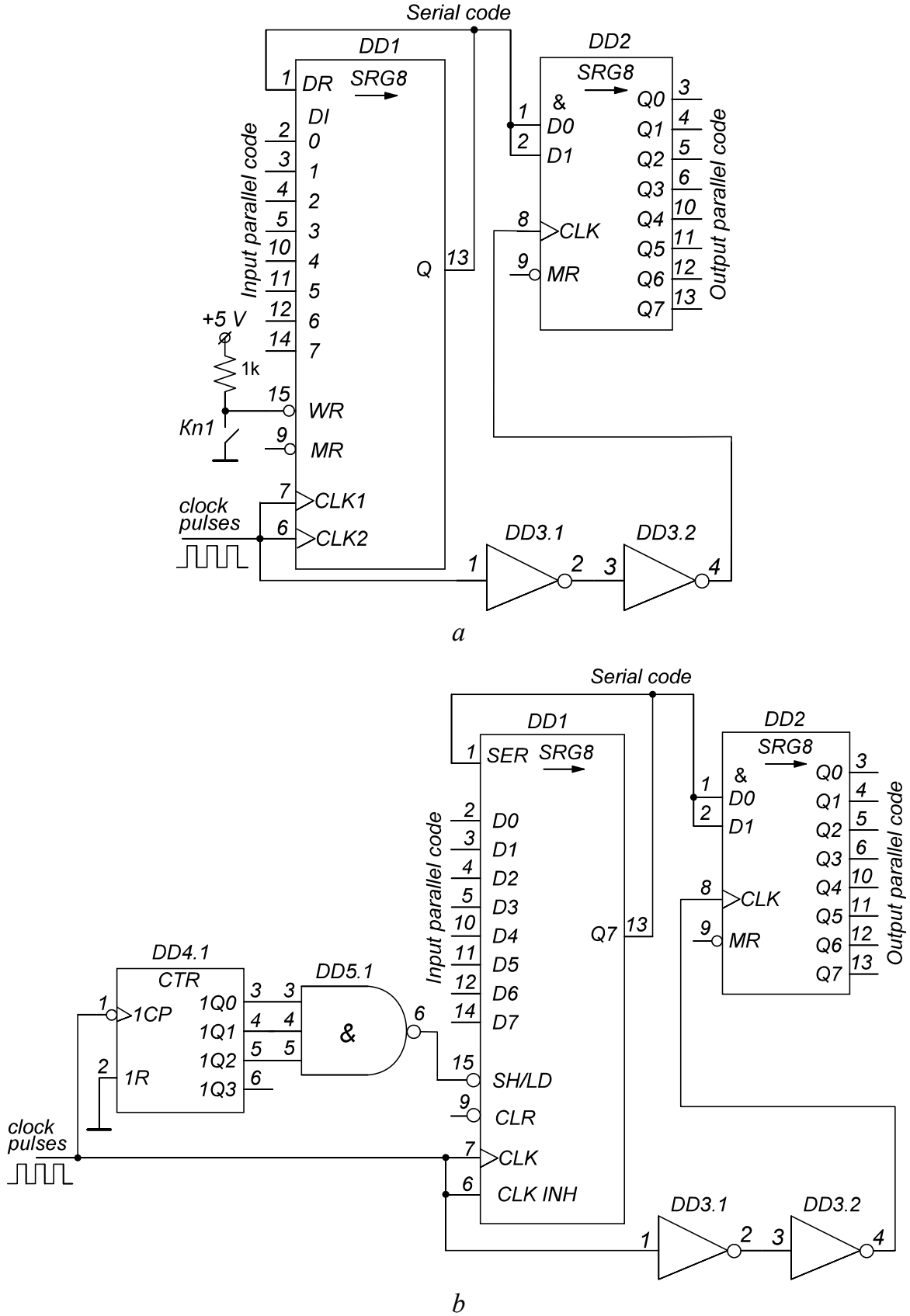


Fig. 9.5. Serial transfer of the parallel data by means of the shift registers

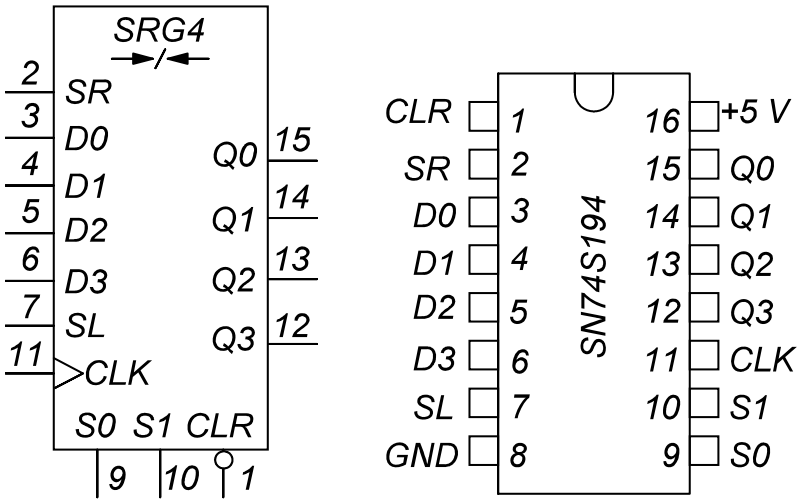


Fig. 9.6. IC K531IR11 logic symbol and pin configuration

Table 9.6

The role of the IC K531IR11 outputs

Output symbol	Role
Q0 – Q3	Data outputs
D0 – D3	Data inputs for parallel uploading
C	Shift/record clocking
DL	Serial data input (shift to the left)
DR	Serial datan input (shift to the right)
S0, S1	Mode select input
R	Asynchronous reset

Table 9.7

IC K531IR11operating modes

S0	*S1*	Mode
0	0	Storage
0	1	Shift to the left
1	0	Shift to the right
1	1	Parallel upload

The IC K531IR11 is a 4-bit shift register which makes it possible to execute serial and parallel data writing, serial and parallel reading, and shift. The IC operating modes are given in table 9.7. The input *C* serves for supplying clock pulses which shift or upload information. The shift or upload happens at the pulse positive edge. When supplied to the control inputs *S0*=0 and *S1*=0 the clock pulses input is being blocked, and the register stores the information which was uploaded earlier. When *S0*=1 and *S1*=1 the register operates as a memory register, i.e. the information from the data inputs *D0*–

D3 is written at every positive edge of the input *C*. The inputs *DL* and *DR* serve for the serial data upload to the register. In case *S0*=1 and *S1*=0 at the clock pulse positive edge, all bits at the outputs *Q0–Q2* shift to the high-order bit region, and the data bit from the input *DR* is uploaded to the low-order bit *Q0*. If *S0*=0 and *S1*=1 at the clock pulse positive edge, all bits at the outputs *Q1–Q3* shift to the low-order bit region, and the data bit from the input *DL* is uploaded to the high-order bit *Q3*.

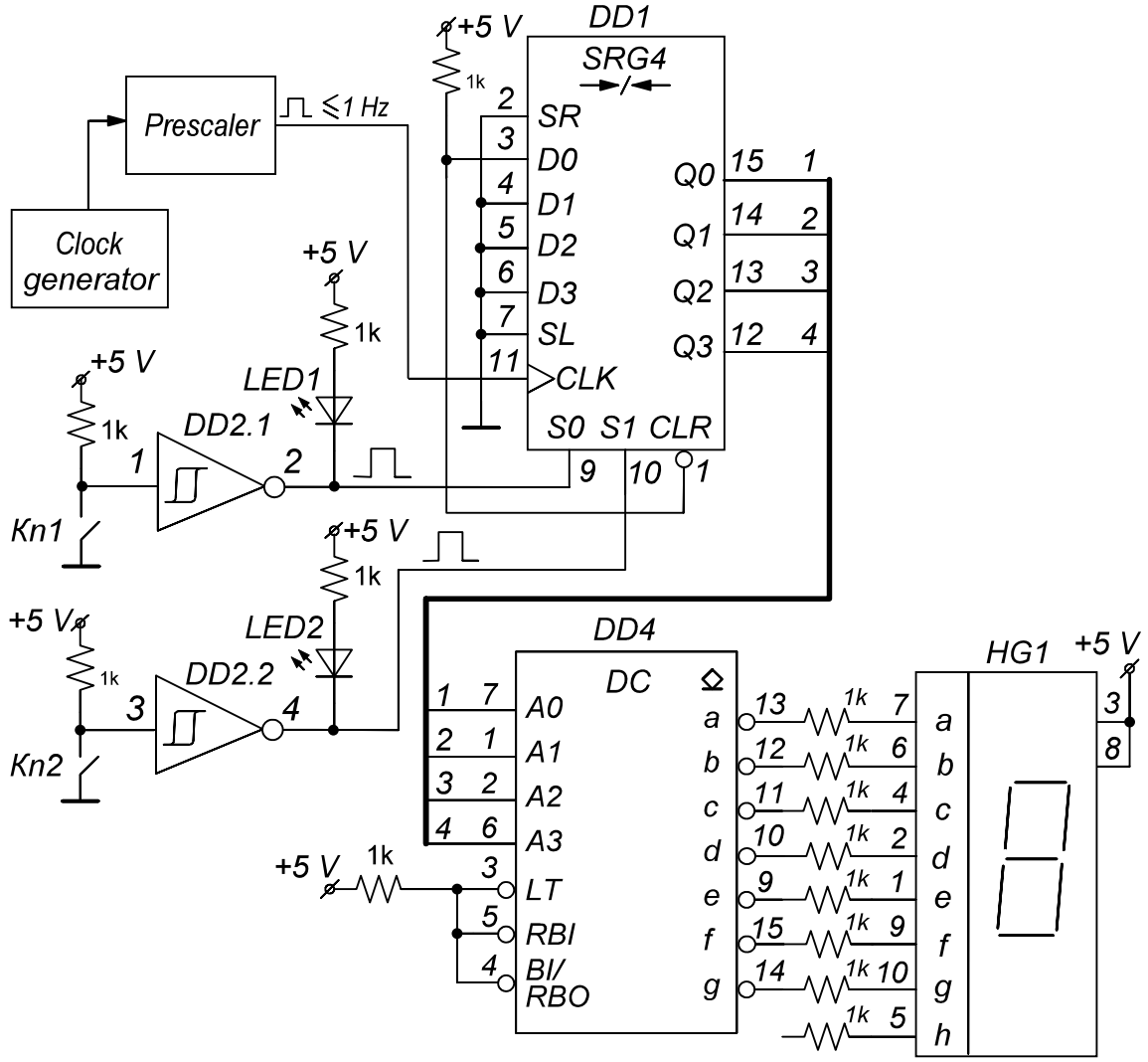


Fig. 9.7. Reverse shift register

Table 9.8

ICs in the circuit in Fig. 9.7

IC type Circuitry symbol	KR1533TV9 DD1, DD2	K531IR11 DD3	K533ID18 DD1
Common	8	7	8
+5 V	16	14	16

Let us now examine the structure of the reverse shift register on the IC K531IR11 base. Fig. 9.7 (see also table 9.8) shows the possible circuit implementation. This circuit provides the shift of data stored in the register to the left or to the right. Such a way multiplication or division by 2 is realized. The data which are going to be shifted are given in the parallel code at the inputs *D0–D3* (in this case – 0001_2).

The microcircuit operating mode is defined by the drivers on the flip-flops DD1 and DD2 in accordance with table 9.7. At first you are offered to upload the number 0001_2 to the register by pushing the buttons *Kn1* and *Kn2* simultaneously. By doing so you'll see that digit '1' starts glowing at the 7-segment indicator (HG1). In order to perform the shifting to the left or to the right and to obtain digit '2'at the indicator (the multiplication of 0001_2 by 2 will give us 0010_2) we should keep on pushing the button *Kn1* during the whole pulse-repetition period of the input *C* signal. If we continue pushing the button, the next shift will happen, i.e. it will be multiplied by 2 again, etc. The division by 2 is performed in similar way (by keeping the button *Kn2* pushed).

Using the register K555IR8 and KR1533IR10 it is possible to implement a serial adder circuit. Such a circuit is shown in Fig. 9.8 (see also table 9.9). The circuit fulfils the serial data addition of the bits of the same position in the registers DD1 and DD2 with account of the carry bit. The last is stored in the flip-flop DD6. The input and output data in the circuit are set in the parallel code. As the registers make shifting to the right only through the high-order bits, the low bits of the summands must be supplied to the inputs *DI7* of the microcircuits DD2 and DD3, respectively, and the high bits – to the *DI0*. The correct result of the addition will be at the outputs starting from *Q0* (high bit) to *Q7* (low bit) of the DD5 after eight clock pulses (including the preset). Before uploading the numbers to the registers DD2 and DD3 the register DD5 and the flip-flop DD6 should be reset. The reset signal is formed by connecting the pin *R* to the zero potential at first, and then to the voltage supply.

9.4 EQUIPMENT

In the lab work the module UIK-1 with a kit of ICs for 'Digital devices' discipline is used. The procedure of the work is the same as described in section 1.4.

In order to carry out the lab work you'll need a set of microcircuits: K555IR8, KR1533IR10, KR531IR11, K555IM5, KR1533TM8, KR1533TV9, KR1533LA3, KR1533LA4 and KR1533IE19.

To record the signals the double-channel oscilloscope and the available LEDs of the module UIK-1 are used.

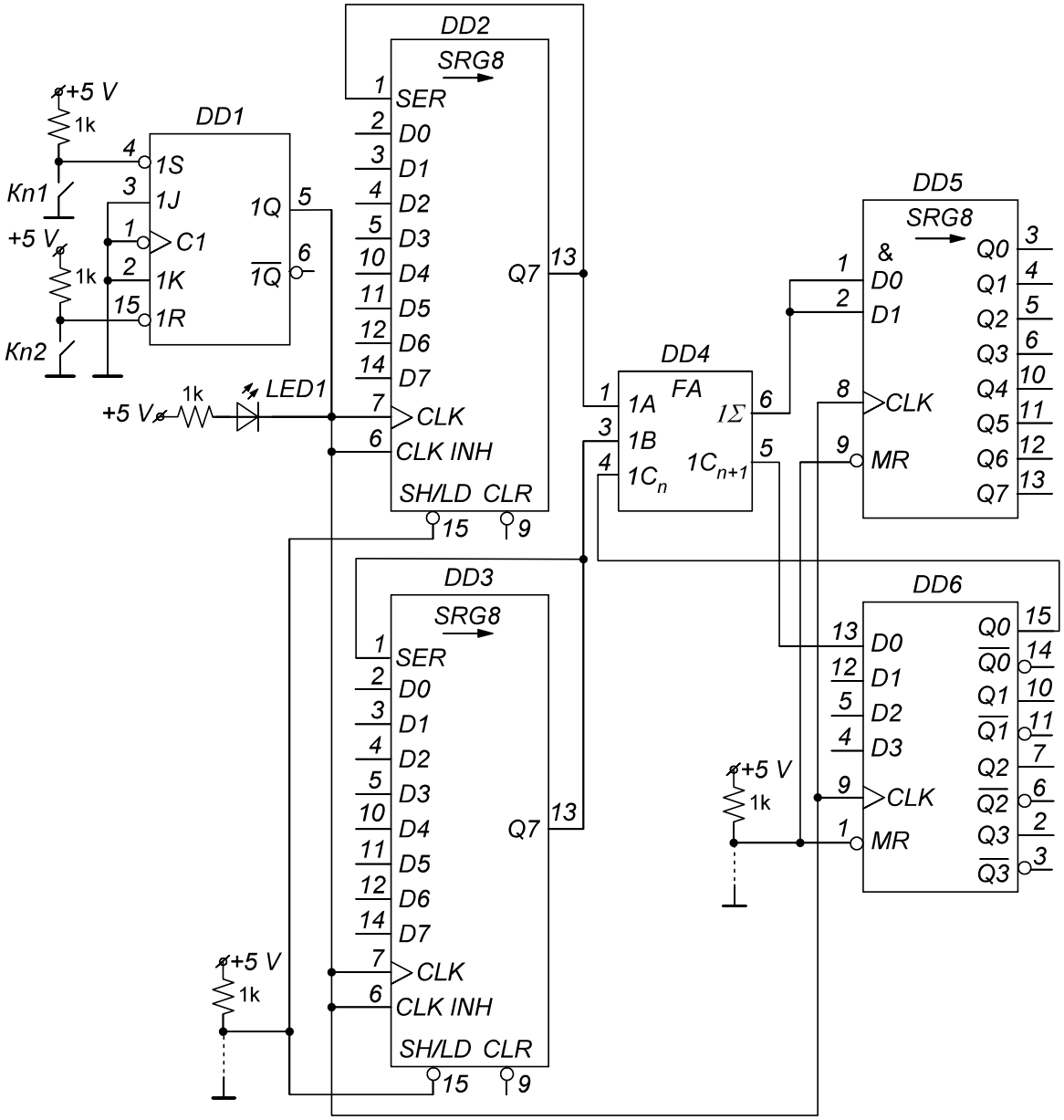


Fig. 9.8. Serial adder

Table 9.9

ICs in the circuit in Fig. 9.8

IC type Symbol	KR1533TV9 DD1	KR1533IR10 DD2, DD3	K555IM5 DD4	KR555IR8 DD5	KR1533TM8 DD6
Common	8	7	7	7	8
+5 V	16	14	14	14	16

9.5 IN-LAB TASKS

1. Implement the register K555IR8 series connection for capacity increase (Fig. 9.2). Upload the binary code given by the lecturer to the register.
2. Assemble Johnson counter with the given counting coefficient.
3. Implement the serial data transfer circuit by means of shift registers (Fig. 9.5). Verify its operating in terms of two binary numbers.
4. Assemble the circuit on the reverse shift register KR1533IR11 basis (Fig. 9.7). Analyse the register functioning (shift to the right and to the left) by setting several random numbers at the inputs *D0–D3*.
5. Implement the series adder circuit (Fig. 9.8). The summand values are given by the lecturer.

9.6 QUESTIONS

1. What are the advantages and disadvantages of the data transfer in the series code? For what purposes are parallel and series data transfer modes applied?
2. How does the sync pulse provide the data shift exactly for one bit in the shift register?
3. Why can't the level synchronized flip-flops be used for shift register design?
4. Enumerate the IC K531IR11 capabilities and suppose where the microcircuit can be utilised.
5. Why is the microcircuit DD1 output *Q* connected with the input *DR* of the same circuit in Fig. 9.5?
6. What is the purpose of the elements DD4.1 and DD5.1 in the circuit in Fig. 9.5, *b*.
7. Explain the series adder operating principle.
8. Where should a delay line in the circuit in Fig. 9.8 be embodied in order to provide reliable operating and how can it value be estimated?
9. Suggest a means of functionality verification of the serial adder in Fig. 9.8, if the clock pulses are generated by external pulser.

REQUIREMENTS TO THE LAB-WORK REPORT

The report should comprise all electrical circuits designed, implemented and assembled during the lab work according to the in-lab tasks. The waveforms of electrical signals at the inputs and outputs of the circuits including intermediate points also need to be presented in the report.

The digital devices operation in static mode should be illustrated by the proper state tables. If during the work it has been needed to make the additional exercises, their results are also included in the report. If the pre-tasks implies calculations (simplifying or another transformations), this calculations are also presented in the report.

The conclusion of the report should give the resume on every sort of study, answers the control questions (min 2 on your choice). Any comments about the work as far as your own impressions are welcome.

The defense of the lab work is carried out when the report has been prepared.

CONCLUSION

The teaching aid presents the series of laboratory works devoted to study of operation principle of digital devices integrated circuits and their application for electronic circuits design.

The authors are grateful to the scientific group supervised by Prof. Soldatov A.I. (Tomsk Polytechnic University) for the engineered commutation module UIK-1, and to the scientific group supervised by Prof. Evtushenko N.V. (Tomsk State University) for the designed PC software. All this makes possible to organize laboratory practice in digital devices on the modern, higher technical level.

REFERENCES

1. Commutation environment: Pat. RU 2280891 C2. Application: 001133237/09, 06.12.01. Published 27.07.06. Bull. 21. – 8 p.
2. Soldatov A.I., Kim O.H. Technic and algorithmic commutation problems of modern electronics // Russian Physics Journal. – 2010 – V. 53. – N 9/3. – P. 308–311. (in Russian)
3. Soldatov A.I., Chertov A.C. Methodical instructions to the laboratory works on discipline 'Digital devices'. – Tomsk: TPU Publishing House, 2004. – 62 p. (in Russian)
4. Shilo V.L. Popular digital devices: handbook. – 2nd ed., Moscow: Radio i svyaz', 1989. – 350 p. (in Russian)
5. Avansian G.R., Levshin V.P. Integrated microcircuits TTL, S-TTL: handbook. – Moscow: Mashinostroenie, 1993. – 252 p. (in Russian)
6. Shilo V.L. Popular digital devices TTL: Series: KR1533, KR1531, K531, K555, K155. – Moscow: Argus, 1993. – 64 p. (in Russian)
7. Avansian G.R., Bespalov A.A. Unipolar integrated circuits: reference manual – Moscow: Goryachaya liniya - Telekom, 2003. – 220 p. (in Russian)
8. Electronic component's datasheet search site, http://www.alldatasheet.net
9. Analog and digital electronics: a first course / Beards H. Peter. – 2nd ed., London: Prentice Hall, 1996. – 646 p.
10. Electronic Devices and Circuits / S. Salivahanan, N. Suresh Kumar, A. Vallavaraj. – New Delhi: Tata McGraw-Hill Publishing Co. Ltd., 1998. – 705 p.
11. Farhat H.A. Digital Design and Computer Organization / H.A. Farhat. – New York: CRC Press, 2004. – 487 p.
12. Online basic electronics tutorials and revision web site, http://www.electronics-tutorials.ws
13. All About Circuits: online textbooks covering electricity and electronics, http://www.allaboutcircuits.com
14. Wikipedia, the free encyclopedia, http://en.wikipedia.org

Appendix

THE USED UNTEGRATED CIRCUITS AND THEIR ANALOGS

Name	Discription	Russian name	Foreign analog
Logic gates			
KR1533LA1	Dual 4-input positive-NAND gates	КР1533ЛА1	SN74ALS20AN
KR1533LA3	Quadruple 2-input positive-NAND gates	КР1533ЛА3	SN74ALS00AN
KR1533LA4	Triple 3-input positive-NAND gates	КР1533ЛА4	SN74ALS10AN
KR1533LE1	Quadruple 2-input positive-NOR gates	КР1533ЛЕ1	SN74ALS02N
KR1533LE4	Triple 3-input positive-NOR gates	КР1533ЛЕ4	SN74ALS27AN
KR1533LP5	Quadruple 2-input EXCLUSIVE-OR gates	КР1533ЛП5	SN74ALS86N
KR1533TL2	Hex inverters with Schmitt trigger inputs	КР1533ТЛ2	SN74ALS14N
KR1533LN1	Hex inverters	КР1533ЛН1	SN74ALS04AN
Combinative devices			
KR1533KP2	Dual 4-input multiplexer	КР1533КП2	SN74ALS153N
KR1533KP7	8-input multiplexer	КР1533КП7	SN74ALS151N
KR1554ID14	Dual 1-of-4 decoder/demultiplexer	КР1554ИД14	74AC139N
533ID18 / 555ID18	BCD-to-seven-segment decoder/driver	533ИД18 / 555ИД18	SN54LS247 / 74LS247N
K555IM5	Dual carry save full adder	К555ИМ5	SN74LS183N
K555SP1	4-bit magnitude comparator	КР555СП1	SN74LS85N
Sequential devices			
KR1533TV9	Dual J-K negative edge-triggered flip-flop	КР1533ТВ9	SN74ALS112A
KR1533TM8	Quad D flip-flop with common reset	КР1533ТМ8	SN74ALS175N

THE USED UNTEGRATED CIRCUITS AND THEIR ANALOGS
(continue)

Name	Discription	Russian name	Foreign analog
Sequential devices			
KR1533TM2	Dual D-type flip-flop with set and reset	КР1533ТМ2	SN74ALS74AN
KR1533IE7	Synchronous 4-bit up-down counter	КР1533ИЕ7	SN74ALS193N
KR1533IE19	Dual 4-bit decade counter	КР1533ИЕ19	SN74ALS393N
KR555IR8	8-bit serial-input/parallel-output shift register	КР555ИР8	74LS164
KR1533IR10	Parallel-load 8-bit shift register	КР1533ИР10	SN74ALS166N
K531IR11	4-bit bidirectional universal shift register	КР531ИР11	SN74S194

CONTENTS

For notes

For notes

Educational Edition

Национальный исследовательский
Томский политехнический университет

ГУБАРЕВ Фёдор Александрович
НАКОНЕЧНАЯ Марина Евгеньевна

ЦИФРОВЫЕ УСТРОЙСТВА

Практический курс

Учебное пособие

Издательство Томского политехнического университета, 2013

На английском языке

Published in author's version

Science Editor
ScD, Professor A.I. Soldatov

Typesetting *F.A. Gubarev*
Cover design *A.I. Sidorenko*

Printed in the TPU Publishing House in full accordance with the quality of the given make up page

Signed for the press 17.07.2013. Format 60x84/16. Paper "Snegurochka".
Print XEROX. Arbitrary printer's sheet 6.51. Publisher's signature 5.90.
Order 783-13. Size of print run 100.

TPU PUBLISHING HOUSE. 30, Lenina Ave, Tomsk, 634050, Russia
Tel/fax: +7 (3822) 56-35-35, www.tpu.ru