МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт студентами з англійською мовою навчання при вивченні дисципліни «Основи мікропроцесорної техніки» для підготовки бакалаврів за спеціальністю 141 «Електроенергетика, електромеханіка та електромеханіка» з подальшим навчанням за освітною програмою «Електричні машини і апарати»

2016

Методичні вказівки до виконання лабораторних робіт студентами з англійською мовою навчання при вивченні дисципліни «Основи мікропроцесорної техніки» для підготовки бакалаврів за спеціальністю 141 «Електроенергетика, електромеханіка та електромеханіка» з подальшим навчанням за освітньою програмою «Електричні машини і апарати » / Укл.: Л.Б. Жорняк, Г. А. Рябенко - Запоріжжя: ЗНТУ, 2016. – 66 с.

Укладачі: Л.Б. Жорняк, доцент, к.т.н.
Г. А. Рябенко, викладач

Рецензент: О.В. Близняков, доцент, к.т.н.

Відповідальний
за випуск: П.Д. Андрієнко, професор, д.т.н.

Затверджено
на засіданні НМК ЕТФ
протокол №1
від 20.09.2016

Затверджено
на засіданні кафедри
"ЕЕА", протокол №1
від 16.09.2016

CONTENT

1. LABORATORY WORK №1
EDUCATIONAL MICROPROCESSOR SET

The purpose of work is to study the device of an educational microprocessor set УМК-1 and bodies of its control, input, indication.

Subject of a research: an educational microprocessor set УМК-1 (УМК), its opportunities and control devices.

1.1. Assignment and characteristics of УМК

УМК represents the completed microcomputer and is intended for preparation of the experts in the field of microprocessor engineering by studying of structure and bases of programming of the microprocessor КР580ВМ80А. УМК can be used as the controlling computer at creation and research of operation of control systems of electrical devices and processes. It is an easily Make ed and convenient tool for debugging small (up to 2 Kilobytes) programs of the user. The means of indication on a front panel allow to observe processes of transformation and information transfer during operation of УМК.

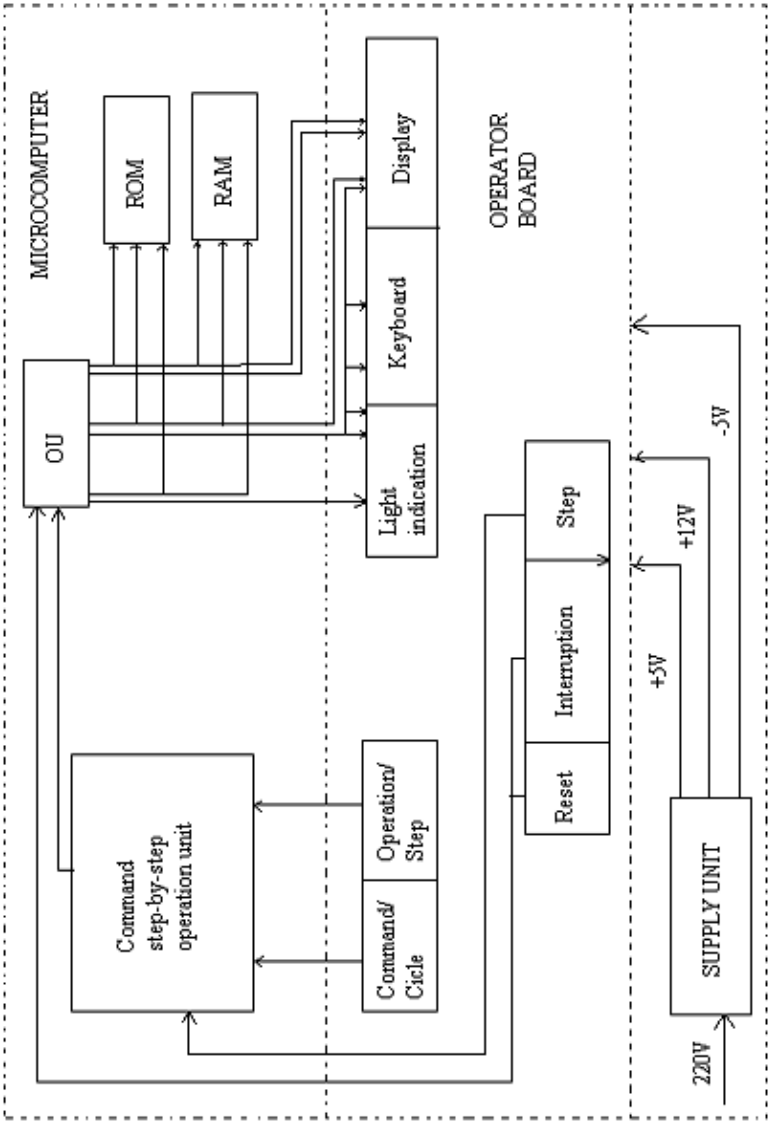
Technical characteristics of УМК-1.

Type of the used microprocessor ---------- КР580ВМ80А
Volume of the random-access memory ---------- 2 Kilobytes
Volume of the read-only memory- ----------2 Kilobytes
Opportunity of interruption ---------- 1 vector
The software ---------- System program "Monitor"
Voltage of power supply ---------- 220 V + /- 22 V
frequency 50 Hz + /- 1 Hz
levels of input and output signals are compatible to levels a TTL IC

1.2. Arrangement of УМК

УМК consists of the following components (pic. 1.1):

- Microcomputer;
- Operator board;
- Supply unit.



Picture 1.1

Microcomputer is one of the main parts and it controls the work of the hole УМК, making calls to memory, input, output and indication of information. The base of the microcomputer is operation unit (OU), which

consists of the microprocessor (MP) and generator of clock frequency (GCF). Microprocessor KP580BM80A is used for executing of a definite set of commands and is made on one BIC, which includes accumulator and a few registers of common application (RCA). Under register we should under stand a minimum storing element, consisting of 8 digits (bits). Information, stored in such register is called a word. Register a RAM are usually called cells, to which a definite number-address is assigned. Those registers are used for data and user commands storing and are made as separate BIC. Microprocessor can address them with the help of special commands. Accumulator (A) – is the most important register of MP. The operation of УМК consists in the data exchange between registers and is described by the system of commands, every of which defines the definite type of exchange and transform of data.

1.2.1. Architecture of УМК memory

Let's consider structure of the registers, accessible to the user, УМК on basis of MP KP580BM80A (fig. 1.2). Address of all registers inside УМК is carried out in a binary notation, while it is convenient to the user for reduction of record to use a fallback hexadecimal system. Further for a label of addresses and Content of the registers we shall use a fallback hexadecimal system. The size of a MP KP580 word makes 8 bits (1 byte), therefore digit capacity of all reduced registers, except for the index SP (Stack Pointer) and counter of commands PC (Program Counter) also makes 1 byte. MP KP580 has one accumulator A and six RCA: B, C, D, E, H, L. RCA can be combined in the two-byte registers BC, DE, HL. In this case in the registers B, D, H - higher, and in the registers C, E, L - lower byte of a sixteen - digit word. MP contains also register of flags of conditions f, in which 5 digits from 8 are used.

The execution of the programs requires, as a rule, to address to a base memory and buffer registers of an input unit (IU). Buffer registers are used for coordination of MP and peripherals (printer, display, transmitters, measuring instruments etc.) performance. Calling to all registers is because of a lot of data is cannot be located in RCA and the program should have communication devices with "an exterior world" for deriving and getting the information. It is necessary therefore to imagine an accessible amount of memory and registers of IU.

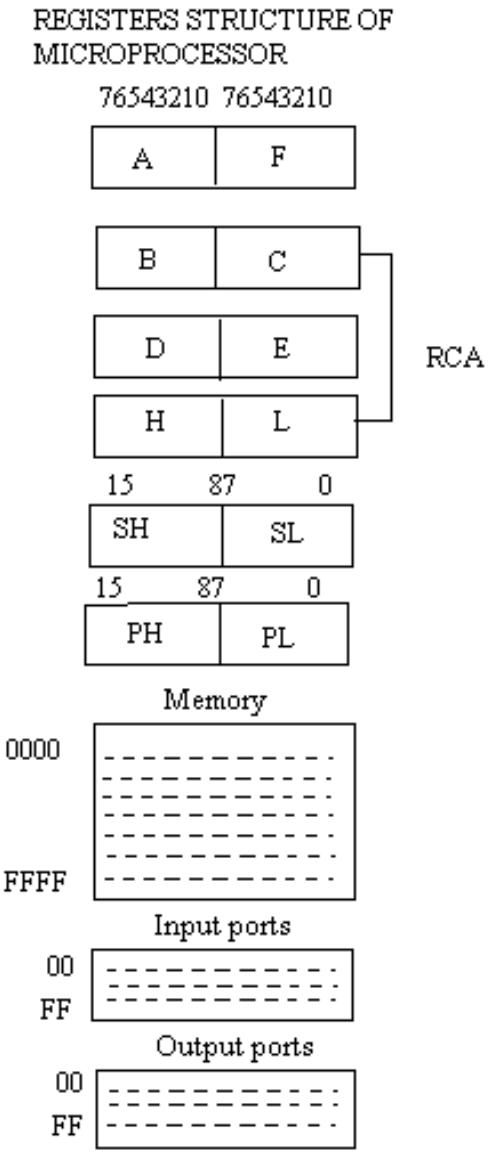

REGISTERS STRUCTURE OF MICROPROCESSOR
76543210 76543210
A
F
B
C
D
E
H
L
RCA
15 87 0
SH
SL
15 87 0
PH
PL
Memory
0000
FFFF
Input ports
00
FF
Output ports
00
FF

Figure 1.2

The address space (field) of memory is a set of cells of memory, which can be addressed by MP. The buffer register of the address of MP KP580 - 16-digit, therefore field of memory contains 2 = 65536 cells (64 Kilobytes, since 1 Kilobyte = 1024 bytes) capacity of everyone is 1 byte with the appropriate address from 0000 up to FFFF. The real microcomputer can use not all address field, but only its part. The set of cells of memory as a matter of fact available in the given microcomputer forms working (physical) space (field) of memory. In УМК the working space of memory makes 4 Kilobytes, from which 2 Kilobytes are occupied with a ROM and 2 Kilobytes - RAM. From a ROM the information can only be read out, and the RAM allows both to read out Content from its cells, and to record in them new data. The address space IU allows the microcomputer to contain up to 2- 256 devices (ports) of input and output addressed separately. The addresses of ports can be in limits 00 - FF.

In УМК 1 Kilobyte of the ROM (address 0000 - 03FF) occupies the program "Monitor" and 1 Kilobyte of the ROM (address 0400 - 07FF) is given at disposal of the user. The RAM is used for a battery-drive storage of the varying programs and data of the user.

Last 54 cells a RAM (the addresses FFCA - FFFF) are occupied by the program "Monitor" for a battery-drive storage of operative data and should not be used by an operator. The address 0800 is initial, with which all programs of the user begin.

1.2.2. The operating unit

The operating unit makes all operations on processing the information. Its initial condition is reading the information to the zero address of a ROM every time after pressing the controlling button RESET on the board. Thus the program "Monitor" is called which ensures output of the information from the pad and putting it on the indicator. The information on a condition of on-line storage (OLS) is put to the register of a condition in the beginning of each machine cycle. In the table 1 the possible condition OLS are shown.

To a condition 0 in the table there corresponds a low level of potential, and condition 1 - high. Depending on a condition of this register the signals controlling operation of all microcomputer are shaped. In the table 2 the definition of each bit of the register of a condition is given. The

feature above a pilot signal WO in the tables 1.1 and 1.2 specifies the active condition of signal, that is logic 0.

Table 1.1 - Possible conditions of the OU

Interrogate	Storage stage of a register							
	D0 INTA	D1 WO	D2 STACK	D3 HLTA	D4 OUT	D5 M1	D6 INP	D7 MEMR
Computer instruction choice	0	1	0	0	0	1	0	1
Storage reading	0	1	0	0	0	0	0	1
Storage Writing	0	0	0	0	0	0	0	0
Stack Reading	0	1	1	0	0	0	0	1
Writing of stack	0	0	1	0	0	0	0	0
Input	0	1	0	0	0	0	1	0
Output	0	0	0	0	1	0	0	0
Interruption	1	1	0	0	0	1	0	0
Halt	0	1	0	1	0	0	0	1
Interruption of a Halt	1	1	0	1	0	1	0	0

Table 1.2 - Bits of the register of a condition

Name	Control digit	Comment
INTA	D0	Shows the signal of the Interruption
WO	D1	Shows that in the current machine cycle there is writing to memory or output operation
STACK	D2	The address bus has the data of stack
HLTA	D3	Shows the signal of the command “NO”
OUT	D4	Shows that in the current machine cycle there is output command working
M1	D5	Shows that in the current machine cycle for reading of the 1 command byte
INP	D6	Shows that in the current machine cycle the input command is working
MEMR	D7	Shows that in the current machine cycle there will be storage reading

1.3 УМК input and indication control devices

On the face side of УМК there are:

- On/off button;
- Reset and interruption button;
- Step-by-step mode control button;
- Functional keyboard;
- Data input keyboard;
- Data bus, address bus condition, MP control signals light indicators.

On/off button is placed in the lower left part of the face panel. Pressed button corresponds to ON position. Over the button there are 3 light indicators: +5V; -5V; +12V. During overloading the safety works and the correspondent indicator lights on. In this case the УМК should be turned off. Button RESET (СБ) is used for the initialization of the system program "Monitor". After pressing it the program "Monitor" start is made

and in the left position of the display symbol "-" appears. It means that УМК is ready for receiving programs. Button INTERRUPTION (ПР) is placed under the СБ. After pressing this button the signal INTERRUPTION CALL OF 7-TH LEVEL is made and if the interruption is allowed (command EI (Enabled Interrupt) was made) the current program execution will stop, after that the control will be transferred to the address 38H. If ПР is pressed during the program "Monitor" execution the sign "?" will appear on the screen. In the opposite case the address of interruption point will be on the screen. Buttons OPERATION/STEP (РБ/ШГ), COMMAND/CICLE (КМ/ЦК), STEP (ШГ), control the execution of step-by-step mode. Those buttons set one of the two modes of step-by-step operation. First – command. For setting this mode button РБ/ШГ should be pressed. Each pressing of the button ШГ causes the execution of the current command. During this on the light indicators of data bus, address bus condition, MP control signals address, code of executed command, control signals will be lit in the binary code.

Second mode – operation by command cycles. For putting this mode buttons РБ/ШГ and КМ/ЦК should be pressed. In this the way of command execution can be traced. After pressing ШГ the next machine cycle will be executed. Light indicators there will be information corresponding to each machine cycle. Keyboard of УМК is divided into two parts. In the left there are functional buttons. Definite function of "Monitor" program is assigned to that buttons:

П – read and edit memory cell;
РГ – read and edit register content;
СТ – program start;
КС – calculation of control sum;
ЗК – filling of memory array by a constant;
ПМ – moving memory array;
__ -- divider;
ВП – execute.

The right part is used for entering parameters in hexadecimal system.

PH – higher byte of command counter;
PL – lower byte of command counter;
SH – higher byte of stack pointer;
SL – lower byte of stack pointer;
A, B, C, D, E, F, H, L – registers.

Six-segment display is used for data displaying in the hexadecimal system. Four left segments show address and register identification, two right – data.

1.4 Getting ready for work

It's necessary to execute following:

– put On/Off button to no pressed position;
– connect УМК to the circuit with 220V;
– press buttons РБ/ШГ and КМ/ЦК;
– turn on the УМК. Light indicators +5V, -5V, +12V should not be lit;
– press button СБ and sign "__" should appear on the display. УМК is ready.

The next turning on of the УМК should be no less then in 20 sec after the turning off. In the other case the supply unit safety will work and the indicators will be on. In this case turn off the УМК and wait until the indicators will go down.

1.5 Task

1. Study the structure, basic technical characteristics and operating controls of УМК.
2. Study and practically to Make the working procedure with УМК.
3. Make the basic operations with keyboard, setting of different modes and using of the indication.

1.6 Methodical guide

Define the location of basic controls on the УМК, and their meaning.

Define the location and the type of indication.

Study characteristics and structure of basic УМК parts.

Make practically the order of turning on and off.

Make the way of calling and reading of content of registers, memory cells.

Make the work with keyboard.
Study the way of setting of different modes, role of lone indicators.

1.7 Content of the report

1. Topic, purpose, review of the done work.
2. Practical aspects of considered questions.
3. Conclusions.

1.8 Questions for self-verification

1. Basic characteristics of УМК, its functional possibilities.
2. Structure of УМК, basic parts, their characteristics.
3. Characteristics and memory organization.
4. Register structure of microprocessor.
5. Characteristic of the possible conditions of operating unit.
6. Characteristic of the controls, indication.
7. Basic requirements for the proper operation of the УМК.

2. LABORATORY WORK № 2 ARCHITECTURE OF THE MICROPROCESSOR KP580BM80A. THE REGISTERS OF THE MICROPROCESSOR. COMMANDS OF REGISTERS LOAD. COMMANDS OF TRANSFER

The purpose of work is to study the structure of MP KP580BM80A and commands of operation with programmable accessible registers of this microprocessor.

2.1. Architecture of the microprocessor KP580BM80A

The block diagram of investigated MP is in fig. 3.1. The following devices go into structure of MP KP580BM80A:

- Six 8-digit registers of common assignment (RCA) -B, C, D, E, H, L;
- 8-digit accumulator (A);
- Four 8-digit registers for temporal storage of the information BP1, BP2, W and Z;

- 8-digit register of attributes (flags) RA (F);
- 8-digit arithmetic-logic device of parallel action (arithmetic and logic unit);
- Circuit (scheme) of a decimal correction (CDC);
- 8-digit register of commands (RC);
- 16-digit counter of commands (CC);
- 16-digit index of stack (SP);
- 16-digit register of the address (RA);
- Decoder of commands and circuit (scheme) of control of a machine cycle (DC and CCMC);
- Multiplexer (M);
- Circuit (scheme) of the register choice (CCR);
- Control unit (CU);
- Buffer of data (BD);
- Buffer of the address (BA).

For the programmer in MP KP580BM80A 10 registers are accessible, they are: six RCA, accumulator, counter of commands, index of stack, and register of attributes. RCA are used for storage of numbers participating in operation, for the indicating of the address of a cell of memory or 16-digit data. In the latter case 8-digit registers B, C, D, E, H, L are combined in register pairs BC, DE, HL. In the first registers of pairs - B, D, H are situated the higher bytes, and in the second – C,E,L – lower bytes. The calling is carried out by a name of the first register. Hence, to the listed pairs it is possible to call under the letters B, D, H accordingly. Except for a name every RCA, and also register A have a three-digit binary code:

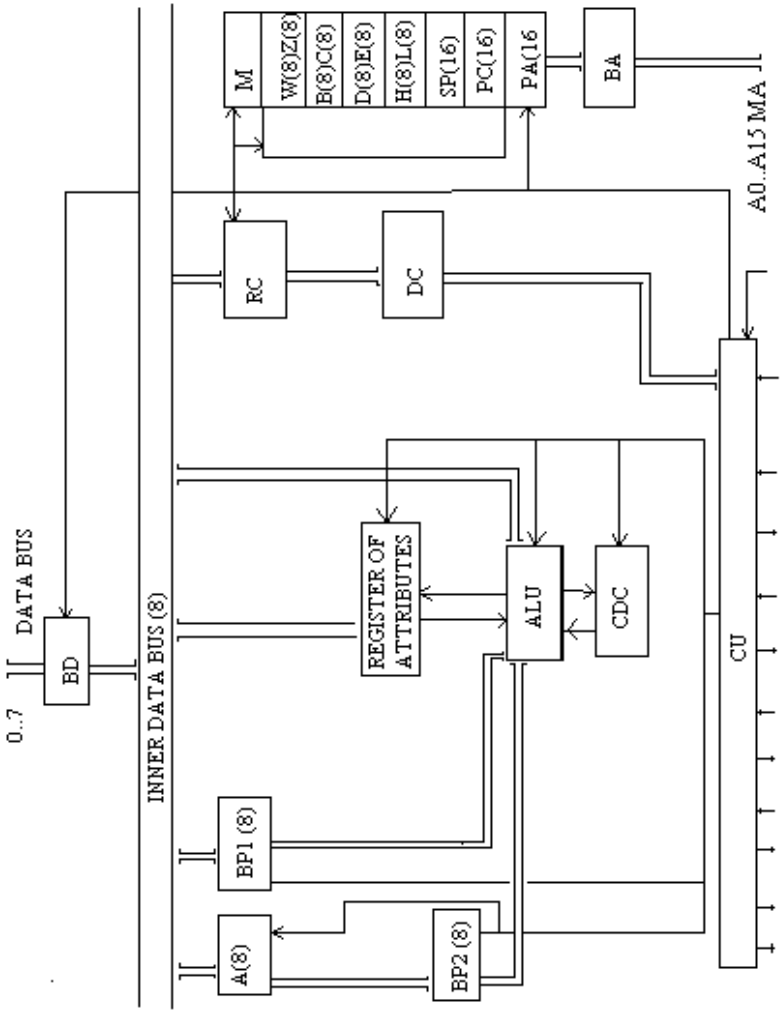
Name of the register.... …..	B	C	D	E	H	L	M	A
Binary code..……...	000	001	010	011	100	101	110	111

M - is the memory cell, which address contains in register pair HL.

The arithmetic and logic unit is intended for execution of arithmetical and logic operations on 8-digit data and operands.

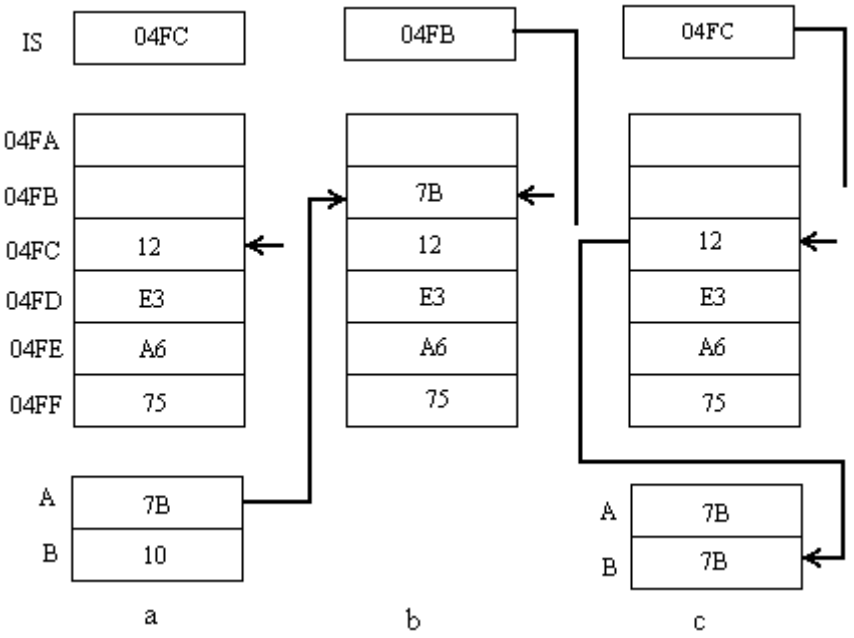
The accumulator - main body of the arithmetical device MP. All arithmetical and logic operations are executed in an arithmetic and logic unit with use of the accumulator. For any of such operations putting one operand in the accumulator, and another - in memory or in RCA is supposed. The result of operation is placed in the accumulator.

The counter of commands is used for making calls to cells of memory, in which the program is stored. It always stores the address of a command, which should be executed. After execution of the next command its content is augmented on 1, on 2, or 3 depending on length of a command in bytes.



Picture 2.1 – Structural scheme of microprocessor KP580BM80

The index of stack SP (IS). Stack - is an area of memory chosen by the programmer, in which the data records and from which reads out in the strictly defined order according to algorithm LIFO (Last-In, First - Out) last has come - first has left. With the help a 16-digit SP the programmer determines what area of memory is selected under the stack. The SP contains the address of a cell of stack, which is accessible for reading of the information. After execution of the reading content of the IS is automatically augmented on 1. Before recording to stack, the Content of IS diminishes by 1, then the record is made. The operations of record and reading from stack are shown in a fig. 2.2.



a) initial condition of stack;

b) stack after execution of the command "write to stack code from the accumulator";

c) stack after executing command "read code from stack to register B".

Figure 2.2 – The operations of record and reading from stack

The register of attributes is intended for storage of 5 attributes, worked out during execution. These attributes (bits of conditions) are kept in the appropriate digits of RA (fig. 2.3).

THE REGISTERS of ATTRIBUTES

7	6	5	4	3	2	1	0
S	Z	O	AC	0	P	1	C

Figure 2.3 – The register of attributes

S (Sign) - attribute of the sign of result kept in the accumulator. At the positive sign S=0, at negative S=1;

Z (Zero) - if Content of the accumulator =0, an attribute Z=1, differently Z=0.

C (Carry) - attribute of carry. C=1, if at execution of commands there is unit of carry from the higher discharge (overflowing).

AC (Auxiliary Carry) - additional attribute of carry. Is established in unit, if at execution of commands there is unit from the third discharge of number (from a lower tetrad in the grown-up)

P (Parity) - attribute of parity. P=1, if the amount of units in discharges of the accumulator will be even. The zero result also falls to even. In an opposite case P=0.

The digits 1, 3 and 5 in the register of attributes are not used as attributes. The complete exposition of each MP command should contain the information on what attributes in the register F this command influences. Content of register pair A and F named as a word of a condition of the program and mean PSW.

2. 2. Structure of commands

The binary numbers chosen from memory and indicating execution of defined operation, refers to as a command. The binary number subjected to processing refers to as an operand. The system of commands executes by MP contains 78 base commands (common number of commands - 244).

Usually command of MP KP580BM80A consists of two fields: fields of an operation code (КОП) and field of the address of an operand (address part). In MP KP580 some ways of setting the address of operands

- ways address are used: a direct address, indirect register address, direct address and immediate address.

Different address ways need different number of digits in a format of a command. Therefore commands of MP have different length - one, two or three bytes.

At direct register address an operand (one or two) is in RCA, which address is showed in an address part of a command (in a binary system of the address have a code 000-111, and in 16 - 0-7). Such commands have length of 1 byte. In a fig. 2.4,a there is the binary representation of a command of transfer of Content of the accumulator to the register B. On the right hex code of a command is specified.

At direct addressing the address of a memory cell is showed in a field of the address of a command. Such commands have length in three bytes: 1st byte – is operation code, 2nd byte – is lower digits of the address, 3-rd byte – is higher digits of the address. In a fig. 2.4, b the command of a load of the accumulator by Content of a cell 7AC3 is shown.

At immediate addressing the operand is a part of the command and depending on its digit occupies the second or second and third bytes of a command. In fig. 2.4 c there is a command of transfer of a code 4C to the accumulator, and in fig. 2.4, d — is loading of register pair BC by a code 1F5A.

Any command has mnemonic (symbolical) exposition facilitating spelling of the program for the programmer. For example, ADD (to combine), MOV (to send), XCHG (exchange of Content of register pairs D and H) etc. Field of a mnemonics should be separated from operands even by one blank. The operands are numbers in hexadecimal system. If the number begins with the letter, before the letter the insignificant zero is put, differently this number will be recognized as a name. For example, OFAH, OB7H, ODEH etc.

The operands disjoint by a coma, if there is more than one. The blanks between operands inadmissible. It is necessary to remember, that:

a) Maximum 8-digit number 0FFH, and 16-digit - 0FFFFH.

b) It is necessary whenever possible to use shorter (smaller number of bytes) command.

c) If the command influences on definite bits of the register of attributes, other bits of this register do not change the condition.

Types of addressing

CODE	Register receiver	Register source	B1	0	1	0	0	0	1	1	1	47

a)

CODE	B1	0	0	1	1	1	0	1	0	3A
Lower address byte	B2	1	1	0	0	0	0	1	1	C3
Higher address byte	B3	0	1	1	1	1	0	1	0	7A

b)

CODE	B1	0	0	1	1	1	1	1	0	3E
OPERANDS	B2	0	1	0	0	1	1	0	0	4C

c)

CODE	0	0	0	0	0	0	0	1	01
OPERAND	0	1	0	1	1	0	1	0	5A
OPERAND	0	0	0	1	1	1	1	1	1F

d)

a) direct register addressing; b) direct addressing; c, d) immediate addressing.

Figure 2.4

All commands can be divided into 7 groups:

1) Command of transfer of the information.
2) Command of transmission of control or transition.
3) Command of arithmetical and logic operations.
4) Command of organizing of subprograms.
5) Command of input and output.
6) Command of operation with stack.
7) Command of management of a condition of MP. Further we shall use the following conditional labels:

R, Rl, R2...- One of 8-digit RCA or accumulator. RP - register pair B, D or H, and also index of stack.

data - 8-digit operand or its name.

data l6 – 16-digit operand or its name.

addr - address of a cell of memory.

port - 8-digit address УВВ or its name.

2. 3 Commands of RGP loading

The command MVI (Move Immediate) serves to load of an RGP by an 8-digit binary operand. Its length is 2 bytes, executed in 7 steps.

Format of a command MVI R, data.

For example: MVI A, 0FFH - load the accumulator by number FFH; MVI C, FFH - to load the register C by the name FFH.. The load command of each register has hex code. In first byte КОП is put and the receiver register, in second - operand. If as the register R in a command the conditional register M is specified, second byte of a command (operand) loads into a cell of memory, which address previously was written down in a register pair HL.

Table 2.1- Codes of first byte of a command MVI

The register the receiver	A	B	C	D	E	H	L	M
Code of first byte of a command MVI	3E	06	OE	16	1E	26	2E	36

The task.

1. Write down to memory, from address 800H the following sequence of commands.

It is necessary to remember, that the command MVI has 2-bytes, therefore for each command it is assigned two cells of memory. The machine code of a command is entered at one byte. So, in a cell 800 the code 3E, and in a cell 801 - 00 etc.

The address	Command	Machine code	The comment
800	MVI A, 0	3E 00	To load the register A by a code 00H
802	MVI B, 1	06 01	To load the register B by a code 01H
804	MVI C, 2	0E 02	To load the register C by a code 02H
806	MVI D, 3	16 03	To load the register D by a code 03H
808	MVI E, 4	1E 04	To load the register E by a code 04H
80A	MVI H, 5	26 05	To load the register H by a code 05H
800	MVI L, 6	2E 06	To load the register L by a code 06H

2. Execute this sequence with the help of a command <СТ 800 _80E ВП>. On the display the stopping address 80E will appear.

3. Check up Content of the registers, using a command SCANNING AND MODIFICATION of CONTENT of the REGISTERS.

The command LXI (Load register pair Immediate) serves for loading of register pair, specified in a command, by 16- digit data. Length of a command is 3 bytes; it is executed in 10 steps.

In first byte there is operation code and register, in second and third - 16-digit operand is put. And third byte of a command is loaded into the first register of pair, and second byte - in the second register of pair.

Format of a command LXI RP, data 16.

As RP the register pairs B, D, H, SP are used.

For example: LXI D, FF00 H - in the register D higher byte - FF, in the register E lower -00 records.

The codes of a command LXI are shown in the table 3.2.

Table 2. 2 - Codes of a command LXI

Addressed register pair	B	D	H	SP
Code of first byte of a command LXI	01	11	21	31

The task.

1. Write down in memory, starting from addresses 800H, codes of the following sequence of commands:

The address	Command	Machine code	The comment
800	LXI B, 3132H	01 32 31	To load RP BC by a code 3132H
803	LXI D, 3334H	11 34 33	To load RP DE by a code 3334H
806	LXI H, 3536	21 36 35	To load RP HL by a code 3536

2. Execute this sequence of commands with the help of a command START of the PROGRAM.

3. Check up Content of all registers, which should coincide with set values.

Command of loading of the register of the index of stack. The command of an immediate load of the register SP looks like LXI SP, data 16.

The command of an indirect load of the register SP looks like SPHL.

Command has 1-byte. A code of this command is F9. On a command SPHL in the index of stack Content of pair HL are loaded. Therefore register pair HL previously was loaded by the necessary number.

The task.

1.Write down to memory to the address 800H codes of a command:

The address	Command	Machine code	The comment
800	LXI SP, 0B10H	31 10 0B	To load the index of stack SP by a code 0B10H

2. Execute this command.

3. Check up Content of the register SP.

4.Write down in memory to the address 800H codes of the following commands:

The address	Command	Machine code	The comment
800	LXI H 0B30H	21 30 0B	To load RP HL by a code 0B30H
803	SPHL	F9	To load the register SP. by a code from RP HL

5. Execute this command: <СТ 800_ 804 ВП>.

6. Check up Content of the register SP byte-by-byte.

All commands of RGP loading do not form the attributes, i.e. condition of the register F doesn't change.

2. 4 Commands of transfer

The given command is needed for an information transfer from the register to the register, from the register to memory and from memory to the register. A general view of a command:

MOV Rl, R2

R1- Register - receiver,

R2- Register - source.

The command occupies 1 byte and is executed in 5 steps. Doesn't form attributes.

For example, MOV A, B - to send Content of the register B to the accumulator, MOV M, A - to send Content of the accumulator to a cell of memory, which address is stored in pair HL. At execution of all these commands Content of the register - source R2 is saved.

The task

1. Write down to memory, starting from address 800H, codes of the following sequence of commands:

The address	Command	Machine code	The comment
800	MVI A, OOH	3E 00	Zero the register A
802	MOV B, A	47	To send Content A to B
803	MOV C, B	48	To send Content B to C
804	MOV D, C	51	To send Content C to D
805	MOV E, D	5A	To send Content D to E
806	MOV H, L	63	To send Content E to H
807	MOV L, H	6C	To send Content H to L

2. Execute this sequence of commands.

3. Check up Content of the registers.

4. Compare an amount of memory occupied by this program, and duration of its execution to volume and duration of the similar program zeroing of the same registers executed with the help of commands MVI for each register (see the task to section 2.3.1).

2. 5 A command of loading of the counter of commands

On this command in the counter of commands Content of register pair HL record. The command looks like: PCHL. Code of this command is E9. For example, to load to the counter of commands the address 900 it is necessary previously to load it to RP HL, and then to execute a command PCHL. This command has no operands and executes unconditional transition with indirect addressing. Length of a command is 1 byte, duration 4 steps. The command PCHL does not form any attributes.

The task

1. Write down to memory to the address 800H codes of the following commands:

The address	Command	Machine code	The comment
800	LXI H, 0900H	21 00 09	To load RP HL by a code 0900H
803	PCHL	E9	To load PC by a code RP HL. To proceed to the address 0900H

2. Execute these commands <СТ 800_ 804 ВП>. On the display the address 0900 will appear. It will mean, that the address 0900H is loaded in the counter of commands and the transition to this address was made.

The task.

1. Write and execute the program of RGP load with codes corresponding to the your own height in cm.
2. Write and execute the program of loading of register pairs with codes corresponding to your height in cm.
3. Write and execute the program of a load of the register SP:
4. Write and execute the program of transfer for all RGP, previously having loaded one of RGP your own height in cm.

2.6 The Content of the report

1. Theme, purpose, basic information on this work.
2. Results of the executed tasks.
3. Conclusions of the work.

2.7 Question for self-verification

1. Different types of buses.
2. Programmable accessible parts of microprocessor, their short characteristic.

3. Command system, command group, command description.
4. Peculiarities of addressing.
5. Commands of registers and register pairs loading.
6. Transition commands.

3.LABORATORY WORK № 3 ADDRESSING TO MEMORY. METHODS AND COMMANDS OF WORKING WITH MEMORY

The purpose of work is to study addressing methods and commands of operation with memory.

Subject of a research: address space of memory, methods of the addressing and operation with УМК memory.

3.1 Brief theoretical information

Memory is the device that stores commands and data. There is fixed (long-time) and operative memory. Accordingly the devices realizing each type of memory in abbreviated form called ROM and RAM. Accessible to the user practically in all basic modes of УМК operations is the RAM, and RAM operation further should be understood under the term memory.

The device of memory consists of blocks of the identical size - cells of memory, size of everyone is one byte, used for storing of one word of the information in a binary code. In its turn each cell consists of elements of memory, the condition of which corresponds to one binary digit - bit of the information. Cells of memory are numbered by numbers from 0 up to 65535, which called address. Frequently for convenience hexadecimal values of addresses are used, then the address range of УМК is 0000H-FFFFH.

To write down to memory a word, it is necessary to give on the bus of the address of memory signals appropriate to the address of a cell, in which it is necessary to place a recorded word, and to give necessary signals appropriate to a value of a word, on the bus of record.

The memory is arranged in such a manner that the given word will be transferred in a cell with the specified address and can be stored there as long as necessary. In the RAM it is stored as long as the supply voltage is on. At any moment, having addressed the memory, is possible to receive Content of a stored word (a copy). While reading, Content of a cell of

memory do not vary. The time of access does not depend on addresses of a cell of memory. In УМК accessible to the user are the cells with addresses from 0800 up to FFC9 are. The addresses input/output ports can be in limits 00H-FFH, the program monitor occupies the space FFCAH - FFFFH.

The registers and register pairs consisting of a set of storing elements, numbered with digits are applied for intermediate operative storage of words.

In a microprocessor system two methods of addressing to memory: immediate and indirect. The following types of addressing are distinguished: register, indirectly - register, immediate and direct.

At immediate addressing, a cell of memory is put in the second and third bytes of a command.

Format of a command in a general view: КОП AHAL,

where КОП – is operation code;

AHAL – is address of a cell of memory (AH – is address higher byte, AL – is address lower byte).

Whereas the specified command is 3-byte, it will be placed in three cells of memory. After byte of an operation code put down lower byte of the address, and after – higher.

n КОП

n + 1 AL

n + 2 AH

There are two commands of immediate record to memory.

STA AHAL - record to memory by the immediate address AHAL of Content of the register A.

SHLD AHAL - record to memory of Content of register pair HL.

On the given command to the address AHAL Content of the register L, and to the address AHAL + 1 - Content of the register H will be recorded.

EXAMPLE: Write down codes of commands from the table 3.1:

Table 3.1

Address	Command	Code	Comment
800	STA 880H	32 80 08	Record to memory from reg. A to the address 880H
803	SHLD 890H	22 90 08	Record to memory: From reg. L to the address 890H; From reg. H to the address 891H.

Initial values of the registers: A = C7H, H = 55H, L = AAH

Execute by a command: <СТ 800 _ 806 ВП>.

Checkup result on values of Content of cells 880H, 890H, 891H.

Indirect addressing assumes, that the address of a cell of memory located in register pairs HL, BC, DE. For each concrete command of operation with memory register pair is fixed. Therefore before to set a command it is necessary to specify the address in appropriate register pair.

FOR EXAMPLE:

LXI H, 800H; record of the address in the register HL.

MOV M, A; record to memory from the register A to the address HL.

LXI D, 900H; record of the address in the register DE.

STAX D; record to memory from the register A to the address DE.

The commands of reading of memory are distinguished by a type of addressing on immediate and indirect.

Commands of immediate reading of memory:

LDA AHAL - reading of memory by immediate address AHAL in the register A.

LHLD AHAL - reading of memory on immediate address AHAL in register pair HL. Thus in the register L will be recorded content of a cell AHAL, and in the register H – will be Content of a cell with address AHAL + 1.

EXAMPLE - Write down codes of commands according to the table 3.2.

Table 3.2

The address	Command	Code	The comment
800	LDA 880H	3A 80 08	Reading in the register A from a cell 880H.
803	LHLD 890H	2A 90 08	Reading in register L from a cell 890H, in register H of Content cell 891H

To execute the specified commands: <СТ 800 _ 806 ВП>. To checkup Content of the registers: A = C7H, H = 55H, L = AAH.

The commands of read - record of memory at address through a register pair HL looks like:

MOV H, R - record in memory of Content of the register;

MOV R, H - load of the register from memory;

where R - register of common use A, B, C, D, E, H, L.

EXAMPLE

1. Write down the memory codes of the program, shown in the table 3.3. Execute the specified sequence with a command: <СТ 800 _ 826 ВП>. Checkup Content of cells of memory: 900 H = AAH, 901 = CCH, 902 = BBH, 903 = EEH, 904 H = DDH, 905 H = 09H, 906 H = 06H.

2. Write down codes of the program shown in the table 3.4.

Execute the specified sequence of commands: <СТ 800 _ 81С ВП>. Checkup Content of the registers: A = DDH, B = EEH, C = BBH, D = CCH, E = AAH, H = 09H, L = 06H.

The commands of reading - record of addressing with use of register pairs BC and DE look like:

STAX B - record Content of the register A to memory to the address specified in register pair BC;

STAX D - record from the register A to memory to the address from register pair DE;

LDAX D - reading of Content of memory in the register A to the address specified in register pair DE.

LDAX B - reading from memory in the register A to the address from register pair BC.

EXAMPLE Write down to memory codes of the program from the table 3.5.

Table 3.3

The address	Command	Code	The comment
1	2	3	4
800	MVI A, AAH	3E AA	Load the register A with AA
802	MVI B, BBH	06 BB	Load the register B
804	MVI C, CCH	OE CC	Load the register C
806	MVI D, DDH	16 DD	Load the register D
808	MVI E, EEH	1E EE	Load the register E
80A	LXI H, 900H	21 00 09	Load HL with address 900H
80D	MOV M, A	77	Record A to the address HL
80E	LXI H, 901H	21 01 09	---------- ‘ ’ ----------
811	MOV M, C	71	---------- ‘ ’ ----------
812	LXI H, 902H	21 02 09	
815	MOV M, B	70	
816	LXI H, 903H	21 03 09	
819	MOV M, E	73	
81A	LXI H, 904H	21 04 09	
81D	MOV M, D	72	
81E	LXI H, 905 H	21 05 09	
821	MOV M, H	74	
822	LXI H, 906 H	21 06 09	
825	MOV M, L	75	

Table 3.4

The address	Command	Code	The comment
800	LXI H, 900 H	21 00 09	Load HL = 900, address M
803	MOV E, M	5E	Reading E=M, to the address HL
804	LXI H, 901H	21 01 09	Load HL = 901, address M
807	MOV D, M	56	Reading D=M, to the address HL
808	LXI H, 902H	21 02 09	Load HL = 902, address M
80B	MOV C, M	4E	Reading C=M, to the address HL
80C	LXI H, 903H	21 03 09	Load HL = 903, address M
80F	MOV B, M	46	Reading B=M, to the address HL
810	LXI H, 904H	21 04 09	Load HL = 904, address M
813	MOV A, M	7E	Reading A=M, to the address HL
814	LXI H, 905H	21 05 09	Load HL = 905, address M
817	MOV H, M	66	Reading H=M, to the address HL
818	LXI H, 906 H	21 06 09	Load HL = 906, address M
81B	MOV L, M	6E	Reading L=M, to the address HL

EXAMPLE

1. Write down to memory codes of the program from the table 3.5:

Execute the specified sequence with command:<СТ 800 _ 80C ВП>. Checkup Content of cells 900H and 910H: 900H = 0FH, 910H = F0H.

Table 3.5

The address	Command	Code	The comment
800	LXI B, 900H	01 00 09	Load BC = 900H
803	MVI A, OFH	3E OF	Load A =OFH
805	STAX B	02	Record M =A to the address BC
806	LXI D, 910H	11 10 09	Load DE = 910H
809	MVI A, OFOH	3E FO	Load A =FOH
80B	STAX D	12	Record M =A to the address DE

2. Write down to memory codes of the program from the table 3.6.

Table 3.6

Address	Command	Code	Comment
800	LXI D, 900H	11 00 09	Load DE = 900H
803	LDAX D	1A	Reading A = M to the address DE
804	MOV L, A	6F	Transfer L < - A
805	LXI B, 910H	01 10 09	Load BC = 910H
808	LDAX B	OA	Reading A = H to the address BC
809	MOV H, A	67	Transfer H < - A

To execute the specified sequence of commands: <СТ 800_ 80А ВП>. Checkup Content of the registers H, L: H = F0H, L = 0FH.

Register address is used in commands of an aspect: MOV R1, R2; ADD R; DCHI, etc.

The task

1. To study the device of УМК memory, basic ways of addressing and command of call.

2. To write down and to execute the programs of data record to memory from the register, register pair.

3. To write and to execute the programs of a loading the registers from memory, using commands of reading of memory to the register, register pair to the immediate address and command of transfer.

4. To write and to execute the programs of record to memory and reading of its Content at addressing through a register pair and two register pairs.

5. To accept the decisions and to execute any variants of the tasks, in those cases when they are shown. To make and to solve examples with register addressing.

3.2 Methodical guide

1. Study structure of memory, its functional features, ways of recording and writing of Content, basic commands, their formats, ways of exposition. To execute examples.

2. To write and to execute the programs of data recording to memory with your own height.

3.3 Content of the report

1. Specification and requirements of operation, brief review on the investigated questions.

2. Programs written on the requirements of language of the Assembler.

3. Results of execution of the programs, conclusions.

3.4 Question for self-verification

1. Physical fundamentals of memory, its technical realization. Types of memory.

2. Ways and types of addressing.

3. Feature and types of register addressing.

4. Feature of call to cells of memory.

5. Distinction between direct and immediate address.

6. Difference of register addressing from indirect – register addressing.

4.LABORATORY WORK № 4
BINARY ARITHMETICS OF THE MICROPROCESSOR. OPERATIONS AND COMMANDS

Purpose of work is to study arithmetical operations and commands of the microprocessor KP 580.

Subject of a research is way of realization and technique of arithmetical operations with binary numbers realized by the microprocessor.

4.1 Brief theoretical information

Independently of the way of introducing numbers in the initial programs, MP operates only with binary numbers, which has its own specificity. Binary arithmetic's reflect specifics of arithmetical operations with binary numbers. Basic arithmetical operations of MP are addition and subtraction of binary numbers, that implies from principles of architecture of computing processes.

Addition of binary numbers, as well as decimal, is made on digit from lower to the higher, but it is much easier in realization, as the numbers have only two values. The greatest conformity to natural representation of addition gives tabulated form, in which members form lines and columns, and the result of addition of numbers of the same name digits is on intersection of the appropriate line and column. With reference to addition of binary digits the specified table looks like the table 4.1.

Table 4.1 - Addition of binary numbers

	Number		
	1	0	1
Number	0	1	1
	1	1	0 *

* The carry of unit to the higher digit takes place.

The binary addition of pair of numbers is evaluated by the following rules:

- the addition of two units, gives zero, in lower digit of result and carry of unit to the higher digit (1 + 1 = 0 *);
- the adding of unit to zero of lower digit will give in result unit without carry to the higher digit (1 + 0 = 1);
- the result of adding zeros is zero (0 + 0 = 0).

At all simplicity of binary addition as procedures, big size of record of the large numbers in the binary type causes a set of carries from one digit to another, that creates definite inconveniences.

A binary subtraction also in many respects similar to decimal. The tabulated form of record is given in table. 4.2.

Table 4.2 - A subtraction of binary numbers

		Subtraction	
	1	0	1
Reduced	0	1*	1
	1	1	0

* Means a loan of unit from the next higher digit

Fundamentals of a subtraction are:

- a subtraction from unit of unit and from zero of zero gives zero (1-1=0,0-0=0);
- the subtraction of zero from unit gives unit (1 - 0 = 1);
- the subtraction of unit from zero requires a loan of unit from the higher digit also gives result unit (0-1=1).

Arithmetical operations also are shift of a binary number on one digit to the left or to the right, comparison of two binary numbers, and reduction or increasing of number by 1.

Complicated arithmetical operations are such, which can be presented as a combination of several simple. So, the multiplication can be

presented as a combination of addition and shift, and division as a sequence of a subtraction and shift.

In a microprocessor system MP580, which variant is УМК the realization of the following commands of binary arithmetic is possible.

1. Addition of eight-digit numbers.
2. Addition 16 - digit numbers.
3. Subtraction of eight-digit numbers.
4. Increment.
5. Decrement.

All arithmetical operations with eight-digit operands in a system KP580 are grounded on the notion has one of operands situated in the accumulator (register A), and another - in the register (RGP) or in memory (M). The address of a cell of memory (M) should be specified in register pair HL. The second operand can be also number given immediately to the command. The result of arithmetical operation is recorded to the accumulator. The operation of subtraction has that feature that it is always made from content of the accumulator, i.e. reduced number should be an operand placed in the accumulator.

By the result of arithmetical operation of addition and the subtraction the following types of attributes are established:

C - carry, Z - zero, S - sign, P - parity, AC - auxiliary carry

Addition of 16-digit binary numbers in the specified system is referred to as double addition. It's supposed that one of operands is in register pair HL, and second - either in DE, or in BC. The result records to HL. By result of operation bit of carry C is established or reset.

The operation of increment consists in increase of content of the registers, register pairs, cell of memory at the address HL by unit. Increment of register and memory changes bits of attributes: Z, S, P, AC. Increment of register pair does not effect on bits of attributes.

The operation of decrement consists in reduction of Content of the registers, of register pairs, cell of memory at the address in HL on unit. The changed bits of attributes are similar to a command increment.

Commands of addition of eight-digit numbers:

ADD R - addition with the register A, B, C, D, E, H, L;

ADD M - addition with a cell of memory (address in HL);

ADI B - addition with immediate number, B - byte;

ADC R – is addition with the registers: A, B, C, D, E, H, L plus bit of carry C;

ADC M – is addition with cell of memory (address in HL) plus bit of carry C;

ACJ B – is addition with immediate number, B - byte, plus bit of carry C.

EXAMPLES

1. Execute the program from table 4.3 made the operation of A = A+B+M+1

Table 4.3

The address	Command	Code	The comment
800	ADD B	80	A=A+B
801	LXI H, 900H	21 00 09	Load address H
804	ADD M	86	HL=900 H A=A+M
805	ADI 01	C6 01	A=A+1

Execute the program, previously taking the initial data of your own height

2. Input to the memory the program of addition of 16-digit numbers with the basis of commands for 8-digit addition from the table 4.4 for the equation: HL=DE+BC.

Execute the program, previously taking the initial data of your own height

The commands of a subtraction 8-digit numbers look like:

SUB R - subtraction of the register: A, B, C, D, E, H, L;

SUB M-subtraction of a cell of memory (address HL);

SUI B - subtraction of immediate number, B - byte;

SBB R - subtraction of the register: A. B. C. D. E. H. L minus bit of carry C;

SBB M - subtraction of a cell of memory (address HL), minus bit of carry C;

SBI B - subtraction of immediate number. A minus bit of carry.

Table 4.4

Address	Command	Code	Comment
800	MOV A, C	79	Add lower byte of carry bit
801	ADD E	83	at overflowing
802	MOV L, A	6F	Lower byte into L
803	MOV A, B	78	
804	ADCD	8A	Add higher byte, taking carry into account
805	MOV H, A	67	

EXAMPLES

1. Execute the program from the table 4.5 made the operation of A = A – B – M - 1

Table 4.5

Address	Command	Code	The comment
800	SUB B	90	A = A - B
801	LXI H, 900H	21 00 09	Load HL=900H, address M
804	SUB M	96	A=A – M
805	SBI 1	DE 01	A=A – 1

Execute the program previously taking the initial data of your own height. Checkup obtained results.

2. Input to the memory the program from the table 4.6 of a subtraction of 16-digit numbers HL=DE-BC

Table 4.6

Address	Command	Code	Comment
800	MOV A, E	7B	Load A from the register E
801	SUB C	91	Subtraction of data bytes E - C
802	MOV L, A	6F	Load L from the register A
803	MOV A, D	7A	Load A from the register D
804	SBB B	98	Subtraction of higher byte counting carry D, B - bit of carry
805	MOV H, A	67	Load H from the register A

Execute the program previously taking the initial data of your own height. Checkup obtained results.

The commands of double addition look like:

DAD H - addition HL=HL+HL.

DAD B - addition HL=HL+BC .

DAD D - addition HL=HL+DE.

EXAMPLES

1. Execute the program from table 4.3 made the operation from the table 4.7 made the operation HL=BC+DE.

Table 4.7

Address	Command	Code	Comment
800	MOV H, B	60	Transfer H < -B
801	MOV L, C	69	Transfer L < - C
802	DAD D	19	HL=HL + DE

Execute the program previously taking the initial data of your own height. Checkup obtained results.

2. Input to the memory the program of subtraction of the address content of an array cell by its serial number and reading of an array cell in the register A, number of an element (0 - 256) - in the register C, base address of an array - in HL, shown in the table 4.8.

Table 4.8

Address	Command	Code	Comment
800	LXI H, 900H	21 00 09	Load base addresses of an array
803	MVI B, 0	06 00	Load B=0 - higher byte of mot.
805	DAD B	09	HL=HL + BC, address. element array
806	MOV A, M	7E	Reading of an array in register A

Execute the program previously taking the initial data of your own height. Checkup obtained results.

The commands of increment look like:

INR R - increase by unit content of the register:

A, B, C, D, E, H, L.

INR M - increase by unit content of a cell of memory. The address M is in HL.

INX RP - increase at unit of Content of pair registers: BC, DE, SP. In the given command the identifier of the higher register (INX B) should be given.

EXAMPLES

1. Write down to memory to the address 800H a command:

INR E: E = E + 1

Execute the program previously taking the initial data of your own height. Checkup obtained results.

2. To write down in memory codes of commands

800 21 00 09 LXI H. 900H: load. HL = 900H, address M

803 34 INR M : M = M + 1

Execute the program previously taking the initial data of your own height. Checkup obtained results.

3. Write down to memory a code of a command

800 13 INX D: DE = DE + 1

Execute the program previously taking the initial data of your own height as Content of register pair DE. Checkup obtained results.

The commands of decrement look like:

DCR R - reduction by unit of Content of the register A. B. C. D. E. H. L:

DCR M - reduction by unit of Content of a cell of memory - address M in HL

DCX RP - reduction by unit of Content of pair registers: BC, DE, HL. SP.

In a command the identifier of the higher register is specified. For example DCX B.

EXAMPLES

1. Write down to memory the following command: DCR C: C=C - 1.

Execute the program previously taking the initial data of your own height for the register C.

2. Write down to memory codes of commands:

800 21 00 09 LXI H, 900H: Load HL = 900H, address M;

803 35 DCR M: M = M - 1;

Execute the program previously taking the initial data of your own height for initial values of memory cell Content M. Checkup obtained results.

3. Write down in memory a code of a command: DCX H and execute formula HL=HL- 1.

Execute the program previously taking the initial data of your own height as Content of register pair HL. Checkup obtained results.

THE NOTE: the command decrement of register pair does not effect on Content of bits of attributes.

4.2 The task

1. Write and execute the programs of addition and subtraction of Content of the registers, register pairs, cells of memory agrees of the methodical guide.

2. Write and execute the program of filling of an array on the given index of an array cell, duplication of Content of arrays, using commands of double addition.

3. Write and to execute the program of filling of an array of memory by data using commands increment of pair of registers and register.

4. Write and to execute the program of filling of an array of memory by data, using commands of decrement of pair registers and register.

4.3 Questions for self-verification

1. Characteristic of elementary operations of binary arithmetics.
2. Feature and type of introducing of operations of addition and subtraction.
3. Essences and assignment of operations of double addition, subtraction.
4. Operations of shift, their practical value.
5. Principles of filling and carry of arrays.

5.LABORATORY WORK № 5 LOGIC OPERATIONS. ARCHITECTURE AND COMMANDS

Purpose of work is to study principles of architecture and commands of logic operations. Subject of the research: logic operations and commands of data processing.

5.1 Brief theoretical information

Application of microprocessor devices assumes logic processing of information handling. Therefore they are able to replace a set of logic circuits, to execute complicated logic functions on the basis using elementary logic operations.

In a system of commands of МП KP580 for executing of logic operations the following commands are used:

– logic addition;
– logic multiplication;
– excluding or;
– inversion.

The commands of logic addition realize logic operation OR. Result is equal to 1, if even one of the appropriate bytes equal to 1, and is equal to 0, if both are equal to zero. For example:

```
        10101001
OR
        00110010
        10111011
```

where OR — a label of operation OR.

The commands of logic multiplication realize logic operation AND. The result is equal 1, if both appropriate bits are equal to 1, and is equal 0, if even one of them is equal to 0. For example:

```
        1010100
AND
        00110010
        00100000,
```

where AND — a label of operation AND.

The commands of excluding OR realize logic operation ADDITION BY MODULE TWO. The result is equal to 1, if the appropriate bits are opposite (1 and 0), and is equal to 0, if they are identical (1 and 1, 0 and 0). For example:

```
        10101001
XOR
        00110010
        10011011
```

where XOR - label of operation ADDITION BY MODULE TWO.

The command of inversion realizes operation DENYING of Content of the accumulator. For example:

```
        10101001
NOT     ________
        01010110
```

where NOT — a label of operation DENYING.

All logic commands are executed bit-by-bit with eight digit operands. Thus one of operands should be in the accumulator, and second — or in one of the registers of common assignment, or in a cell of memory - or is set in second byte of a command. The result of execution of a command is recorded to the accumulator. Thus bit of carry is established to zero, and other bits are established according to result of execution of a command.

5.2.1 Commands of logic addition:

ORA R - with the register: A, B, C, D, E, H, L;

ORA M - with a cell of memory, which address is in HL;

ORI B - with an immediate operand, B — byte.

EXAMPLES

1. Write down in memory codes of the program, according to the table 5.1. executing the formula : A = (A OR C OR M OR 80H)

Table 5.1

Address	Command	Code	Comment
800	ORA C	B1	A = A OR C
801	LXI H, 900H	21 00 09	Load HL = 900 H, address M
804	ORA M	B6	A = A OR M
805	ORI 80H	F6 80	A = A OR 80H

Execute the program with the command <СТ> 800 __ 807 <ВП> previously taking the initial data of your own height. Checkup obtained results.

2. Write down in memory codes of the program from table 5.2 realizing formula HL = (BC OR DE)

Table 5.2

Address	Command	Code	Comment
800	MOV A, C	79	Transfer A < - C
801	ORA E	B3	A = A OR E
802	MOV L, A B	6F	Transfer L < - A lower byte of result
803	MOV A, B	78	Transfer A < - B
804	ORA D	B2	A = A OR D
805	MOV H, A	67	Transfer H < - A, highest byte of result

Execute the program with the command: <СТ> 800 __ 806 <ВП> previously taking the initial data of your own height. Checkup obtained results.

5.2.2 Commands of logic multiplication

ANA R - with the register A, B, C, D, E, H, L;

ANA M - with a cell of memory, which address is in HL;

ANA B - with an immediate operand. B — byte.

EXAMPLES

1. Write down to memory from table 5.3 codes of the program realizing formula : A = (A AND D AND M AND 7FH)

Table 5.3

Address	Command	Code	Comment
800	ANA D	A2	A = A AND D
801	LXI H, 900H	21 00 09	Load HL = 900 H, address M
804	ANA M	A6	A = A ANA M
805	ANI 7FH	F6 7F	A = A AND 7FH

Execute the program with a command <СТ> 800 __ 807 <ВП> previously taking the initial data of your own height. Checkup obtained results.

2. Write down to memory codes of the program, according to the table 5.4 realizing formula HL = (B OR C) AND DE AND F0FFH

Table 5.4

Address	Command	Code	Comment
800	MOV A, C	79	Transfer A < - C
801	ORA B	B0	A = A ORB
802	ANA E	A3	A = AND E
803	ANI FFH	E6 FF	A = A AND FFH
805	MOV L, A	6F	Transfer L < - A, lower byte of result
806	MOV A, D	7A	Transfer A < - D
807	ANI FOH	E6 FO	A = A AND FOH
809	MOV H, A	67	Transfer H < - A, higher byte of result

Execute the program with a command: <СТ> 800 __ 80A <ВП> previously taking the initial data of your own height. Checkup obtained results.

5.2.3 Commands of addition by module two:

XRAR – with a register: A, B, C, D, E, H, L;

XRAM - with a cell of memory, which address is in HL;

XRIB - with an immediate operand, B - byte.

EXAMPLES

1. Write down to memory codes of the program, according to the table 5.5, realizing formula A = A XOR A XOR E XOR M XOR AAH

Table 5.5

Address	Command	Code	Comment
800	XRA A	AF	A = A XOP A, A = 0
801	XRA E	AB	A = A XOR E
802	LXI H, 900H	21 00 09	Load. HL =900 H, address M
805	XRA M	AE	A = A XOR M
806	XRI AAH	EE AA	A= A XOR AAH, res.

Execute the program with a command: <CT> 800 __ 808 <ВП> previously taking the initial data of your own height. Checkup obtained results.

2. Write down to memory, according to the table 5.6 codes of the program realizing formula HL = (A OR B) AND DE XOR (HL XOR C)

Table 5.6

Address	Command	Code	Comment
800	ORA B	B0	A ORB = A
801	ANA E	AE	A = A AND E
802	MOV E, A	5F	Transfer E-A, record of the subresult
803	MOV A, C	79	Transfer A < - C
804	XRA L	A	A = A XOR L
805	XRA E	AB	A = A XOR E
806	MOV L, A	6F	Transfer L, A, lower byte of result
807	MOV A, H	7C	Transfer A < - H, higher byte of result
808	XRA D	AA	A = A XOR D
809	MOV H, A	67	Transfer H < - A, higher byte of result

Execute the program with a command: <СТ> 800 __ 80A <ВП> previously taking the initial data of your own height. Checkup obtained results.

5.2.4 A command of inversion.

CMA - inversion of the accumulator.

EXAMPLES

1. Write down in memory the program from in the table 5.7 realizing formula A = NOT ((NOT B) AND (NOT C))

Table 5.7

Address	Command	Code	The comment
800	MOV A, B	78	Transfer A < - B
801	CMA	2F	F = NOT A
802	MOV B, A	47	Transfer B < - A (NOT B)
803	MOV A, C	79	Transfer A < - C
804	CMA	2F	A = NOT A (NOT C)
805	ANA B	A0	A = A ANR B
806	CMA	2F	A = NOT A, result

Execute the program with a command: <СТ> 800 __ 807 <ВП> previously taking the initial data of your own height. Checkup obtained results.

2. Write down to memory the program according to the table 5.8 realizing formula : M3 = NOT (NOT Ml) OR (NOT M2)

Addresses of cells of memory: Ml =900 H, M2 = 901 H, M3 = 902 H

Table 5.8

Address	Command	Code	Comment
800	LXI H, 900 H	21 00 09	Load HL = 900 H, address H
803	MOV A, N	7E	Reading A = M1
804	CMA	2F	A= NOT A (NOT M1)
805	MOV C, A	4F	Transfer C < - A, subresult
806	INX H	23	HL = HL + 1, address M2 (901)
807	MOV A, N	7E	Reading A = M2
808	CMA	2F	A = NOT A (NOT M2)
809	ORA C	B1	A = A OR C
80A	CMA	2F	A = NOT A, result
80B	INX H	23	HL = HL + 1, address M3
80C	MOV M, A	77	Record M3 = A, result

Execute the program with a command: <СТ> 800 __ 80D <ВП> previously taking the initial data of your own height. Checkup obtained results.

5.3 Steps of work execution

1. Using command of logic addition writes and execute the programs of realization of the given formula s.
2. On the basis of commands of logic multiplication do the programs of the given formula s.
3. Using the command excluding OR write and execute the programs of realization of the specified formula s.
4. On the basis of inversion of the accumulator make and execute the programs of realization of the given formula s.

5.4 Methodical guide

5.4.1 Execute item 1 of the task for formula HL = (B OR C OR DE) OR 8800H

Fill in the table for values B, C, DE, HL, F. Initial values set arbitrary.

5.4.2 Execute the task 1 for the formula : M3 = Ml OR M2

Addresses of cells of memory accordingly Ml = 900H, M2 = 901H, M3 = 902H.

Input data Ml and. M2 take arbitrary. Results present in the table: Ml, M2, M3, F

5.4.3. Execute task 2 for formula HL = (HL AND BC OR DE) AND 03FFH

Initial value take arbitrary. Fill in the table: HL(in.), BC, DE, HL(res.), F.

5.4.4 Execute formula M2 =A AND Ml OR C AND D

Addresses of cells of memory M1= 900H, M2 = 901H. Initial values take arbitrary. In the table reflect values A, C, D, Ml, M2, F.

5.4.5 According to the task 3 realize formula H = ((L XOR D) OR E AND H) XOR (B OR C) XOR OFH

Initial values take arbitrary. In the table reflect Content of registers: HL(in.), D, E, B, C, HL (res.), F.

5.5 Content of the report

1. Features of logic operations.
2. Essence of elementary logic operations.
3. Technique and commands of logic addition.
4. Technique and commands of logic multiplication.
5. Features and commands of addition by module two.
6. Sense and practical value of inversion.

6.LABORATORY WORK № 6
OPERATIONS AND COMMANDS OF SHIFT.

Purpose of work is to study operations and commands of simple and cyclical shift.

Subject of a research: features and abilities of information processing of the increased formats on the basis of operations and commands of simple and cyclical shifts in a system KP580.

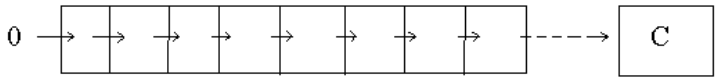
6.1 Brief theoretical information

The commands of shift allow on the basis of elementary arithmetical operations to do a necessary set of operations on numbers. Besides do the completed operations on data processing. Are used mainly at bit testing and comparison.

The commands of shift of MP K580 (Intel 8080) operate only with data of the accumulator and do not require other operands located in memory or the registers. The addressing in this case is named implicit, and frequently the name is not specified. Many MP have others, than given MP, types of commands of shift. For example, MP Motorola 68000 is doing the shift of Content not only of accumulator, but also of memory cells, and command of operations of shift influence not only an attribute of carry, as it is in given MP, but also on all other attributes.

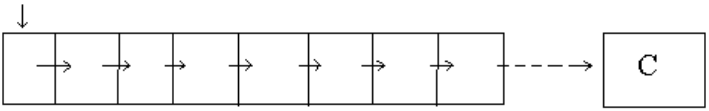
Let's consider the pictures with possible types of shift:

a) shift to the right:



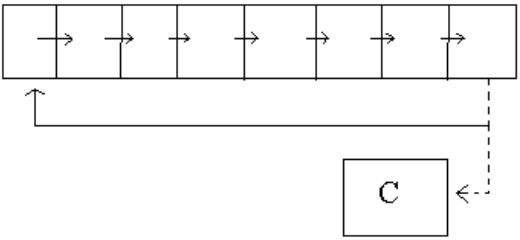
- extreme right bit is transferred to C (register of carry);
- all other bits are moved together to the right;
- in extreme left digit is transferred 0.

b) Arithmetical shift to the right:



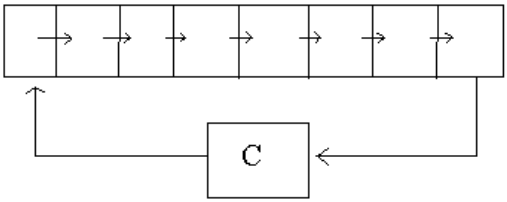
- extreme right bit is transferred to C;
- other bits are moved together to the right;
- extreme left bit remains without change.

c) Cyclical shift to the right:



- extreme right bit is transferred to C;
- other bits are moved together to the right;
- to extreme left digit initial Content of extreme right digit are transferred.

d) Cyclical shift to the right with carry:



- extreme right bit is transferred to C;
- other bits are moved together to the right;
- to extreme left digit initial Content of the register of carry C are transferred;

The shifts to the left have identical technique, but are made in the opposite direction. Their diagrams and algorithms are offered to construct by yourself.

In a system of commands of MP K580 the following commands of shift are used:

a) cyclical shift to the left;

b) cyclical shift to the right;

c) shift to the left through carry;

d) shift to the right through carry.

The commands of shift are executed in the accumulator above 8-digit operands. The result is stored to the accumulator.

6.2 Commands of cyclical shift

The command of cyclical shift to the left displaces each bit in limits of byte on one digit to the left. Thus Content of the higher digit are recorded in lower digit and to a bit of carry.

The command of cyclical shift to the right will move each bit in limits of byte on one digit to the right. Thus Content of lower digit are recorded to the higher digit and a bit of carry.

RLC - command of cyclical shift to the left;

RRC - command of cyclical shift to the right;

EXAMPLES

1. Write down to memory codes of the program realizing cyclical shift of byte on 4 digits

Table 6.1

800	RLC	07	Cyclical shifts to the left on 1
801	RLC	07	---------- ‘ ’ ----------
802	RLC	07	---------- ‘ ’ ----------
803	RLC	07	---------- ‘ ’ ----------

Execute the program, previously set initial values <СТ> 800 __ 804 <ВП>.

2. Write down in memory codes of the program realizing operation of john of the higher tetrads of two bytes, contained in the registers B and C in one, using a command RRC

Table 6.2

800	MOV A, C	79	
801	RRC	0F	Migration of the higher tetrad of 1 byte
802	RRC	0F	on a place of lower
803	RRC	0F	Migration of the higher tetrad of 1 byte
804	RRC	0F	on a place of lower
805	ANI FH	E6 0F	Selection of the higher tetrad of 1 byte
807	MOV C, A	4F	Transfer C < - A
808	MOV A, B	78;	Transfer A < - B
809	ANI FH	E6 F0	Selection of the higher tetrad of 2 byte
80B	ORA C	B1	Join of two bytes in one

Execute the program, previously set initial values <СТ> 800 __ 80C <ВП>. Check up obtained results.

6.3 Commands of shift through carry

The command of shift to the left through carry displaces Content of each bit in limits of byte on one digit. Thus content of bit of carry is recorded to lower digit, and the content of higher bit is recorded to bit of carry.

The command of shift to the left through carry will move Content of each digit in limits of byte to the right on one digit. Thus in the higher digit of byte the value of byte of carry is recorded, and to it content of lower digit of byte will be recorded, using the given command, it is possible to realize a command of division on numbers, multiple 2.

RAL - command of shift to the left through carry;
RAR - command of shift to the right through carry.

EXAMPLES

1. Write down to memory codes of the program realizing cyclical shift to the left on one digit of Content of register pair HL.

Table 6.3

800	ORA A	B7	Reset of carry in 0
801	MOV A, L	7D	Shift to the left of L on 1 digit through carry
802	RAL	17	0 to lower digit of L
803	MOV L, A	6F	
804	MOV A, H	7C	Shift to the left of H on 1 digit through carry
805	RAL	17	Taking into account carry from L
806	MOV H, A	67	Higher digit of H to carry
807	MOVA, L	7D	Carry to lower digit of L
808	ACI 0	CE 00	
80A	MOV L, A	6F	Transfer L < - A

Execute the program, previously set initial values <СТ> 800 __ 80B <ВП>. Checkup obtained results

2. Write down to memory codes of the program realizing operation of multiplication by 4 of Content of the register C. B = C * 4

Table 6.4

800	MOV A, C	79	Transfer A < -C
801	ORA A	B7	Reset of bit of carry
802	RAL	17	Multiplication by 2
803	RAL	17	Multiplication by 2
804	MOV B, A	47	Transfer B < - A

Execute the program, previously set reference values <СТ> 800 __ 805 <ВП>. Check up obtained results.

3. Write down to memory codes of the program of division by 8 Content the register H. The whole part of result should be placed in D, rest in E.

Table 6.5

800	MOV A, M	7C	
801	ORA A	B7	Reset carry to 0
802	RAR	1F	A = A/2
803	ORA A	B7	The carry is equal 0
804	RAR	1F	A = A/2
805	ORA A	B7	The carry is equal 0
806	RAR	1F	A = A/2
807	MOV D, A	57	D = H/8 - whole part
808	MOV A, H	7C	Transfer A < - H
809	ANI 07H	E6 07	Selection of the result rest
80B	MOV E, A	5F	The rest to E

Execute the program, previously set initial values <СТ> 800 __ 80C <ВП>. Checkup obtained results.

6.4 Steps of work execution

1. Make by a program way joining of separate tetrads into byte.
2. Execute logic multiplication of the given bits of one byte by the given bits of another;
3. Make by a program way - cyclical shift on the given number of digits of Content of registers pair.
4. Multiply Content of registers pair by the given number.
5. Divide Content of registers pair by the given number.

6.5 Methodical guide

1. Write and execute the program of joining of lower tetrads of two bytes into one. A lower tetrad of 2nd byte place to the higher tetrad of resulting byte. Use: E – 1st byte, D- 2nd byte, A - result. Initial values of combined bytes take arbitrary.

2. Write and execute the program executing the operation of logic multiplication of 3 bits (5,6,7) of one byte and 3 bits (2,3,4) of the second byte, previously those bit should be moved to the lower digits and other positions zeroed. Take the next values: L – 1st byte, H – 2nd byte and C - result. Initial values take arbitrary.

3. Write and execute the program of cyclical shift on 3 digits of Content of registers pair DE. Input data take arbitrary.

4. Write and execute the program of multiplication of Content of registers pair HL by 4. Input data take arbitrary.

5. Write and execute the program of division of Content of registers pair BC by 8. BC = BC/8. Input data take arbitrary.

6.5 Content of the report

1. Fundamentals of the theoretical information.
2. The initial programs, data and results of the tasks.
3. Conclusions on the done work.

6.6 Questions for self-verification

1. Characteristics of shift to the right or to the left.
2. Features of arithmetical shifts to the right or to the left.
3. Algorithms of cyclical shifts.
4. Cyclical shifts with carry. Their features.
5. Operations and commands of shift of МПК 580
6. Practical use of operations of shift, their usefulness.

7.LABORATORY WORK № 7
OPERATIONS AND COMMANDS OF COMPARISON.

The purpose of work is to study operations and commands of comparison of microprocessor KP 580.

Subject of the research: principles, technique of data monitoring during its processing.

7.1 Brief theoretical information

In practice there are situations, when it is a necessity to compare binary numbers. They are most typical at diagnostics of correctness of MP operation (MPS) and decision making during data processing.

The operation of comparison consists in comparing of any two data elements.

The fact of equality of two numbers can be revealed with the help of operations «Excluding OR" and "Subtraction". However the bad side of those methods is part or complete loss of input data during execution of the specified operations. The operation COMPARISON doesn't have those shortcomings. It based on a subtraction of Content of the register or cell of memory from Content of the accumulator without change of input data, i.e. Content of compared operating elements (cell of memory, register, accumulator) after execution of the specified operation.

In a system of commands of MP KP 580 three types of commands of comparison are used:

- comparison of Content of the accumulator and register,
- comparison of Content of the accumulator and cell of memory,
- comparison of Content of the accumulator with an immediate operand.

The commands of comparison are executed by means of an internal subtraction from Content of the accumulator accordingly, Content of the register, cell of memory or immediate operand. Content of the accumulator thus do not vary. Result of comparison is setting of bits of attributes. The possible variants of set of attributes are shown in the table .7.1.

Table 7.1

Result Comparison	Attributes	
	Zero	Carry
Equally	1	0
It is more	0	0
It is less	0	1

Except for the listed attributes bits of parity and sign are established by results of an internal subtraction. Bit of parity is equal 1 when an amount of units in the result is even, and is equal 0, if an amount of units is odd.

Bit of the sign is set equal to a value of the higher digit of result.

It allows process of execution of the program to be dependent on a value of result of execution of the previous operation. For calling to the information on results of evaluations in MP there is a set of flips making the program-accessible register (Flags) F, which are set to unit or are reset to zero depending on result of executed evaluations. The attributes are formed by results of arithmetical or logic operations. Each flip stores one of condition bits. The values of four bits of conditions can be checked with a help of a program:

1. S - sign of result (1 - minus, 0 - plus);
2. Z - zero result (1 - zero, 0 - not zero);
3. C - carry (1 – there is carry, 0 - the carry is not present);
4. P - parity, (1 - number of units is even, 0 - odd).

Fifth attribute is the additional carry AC from a lower tetrad in the higher during the processing of binary-decimal numbers, which is used together with a command DAA.

Arrangement of attributes in digits of RP the following:

Table 7.2

S	Z	0	AC	0	P	1	C
7	6	5	4	3	2	1	0

The check of attributes of a result is carried out in the binary form of Content of the register F.

7.2 Commands of comparison with Content of the register

Commands of comparison with Content of the register CMP are following: A (BF), CMP B (B8), CMP C (B9), CMP D (BA), CMP E (BB), CMP H (BC), CMP L (BD).

EXAMPLE.

Write down to memory codes of the program of comparison of Content of the registers C and B.

Table 7.3

800	MOV A, C	79	Transfers A < - C
801	CMP B	B8	Comparison with register B

Execute the program, previously set initial values <СТ> 800 __ 802 <ВП>.

For checking the attributes, specified in the table, the content of F should be translated from hex to binary form and Content of required bits is checked.

Command of comparison with Content of a cell of memory CMP M (code BE) - comparison of Content of the register A and cell of memory, which address is given in pair HL.

Example:

Table 7.4

800	LXI H, 900H	21 00 09	Loads HL, address
803	CMP M;	BE	Comparison A and M

Execute the program, previously set the initial values.

7.3 Command of comparison with an immediate operand

Command of comparison with an immediate operand (code FE) are following:

CPI B - comparison of Content of the register A with number given in second byte of a command, where B - byte.

EXAMPLE.

Write down to memory codes of a command of comparison with an immediate operand.

Table 7.5

800	CPI 7FH;	FE 7FH	Comparison of 7FH with A

Execute the program, previously set initial values <СТ> 800 __ 802 <ВП>

7.4 Steps of work execution

1. Make and execute the program of comparison of Content of the registers.
2. Make and execute the program of comparison of Content of the register and cell of memory.
3. Make and execute the program of comparison of Content of the register with an immediate operand.

7.5 Methodical guide

1. Execute item 1 of the task for the registers H and L. Make and fill in the table of input data and results. Input data take arbitrary. Specify condition of basic attributes.
2. Execute item 2 of the task for the register B and cell of memory with address 906H. Fill in the table of input data and results of attributes. Initial values of compared values take arbitrary.
3. Execute item 3 of the task for the register H and immediate operand 5AH. Fill in the table of input data and results of a condition of bits of attributes. Initial values of the register take arbitrary.

7.5 Content of the report

1. Basic information on an investigated question.
2. Programs of executing the tasks in language of Assembler and machine codes.
3. Tables of input data and results, conclusions on the done work.

7.6 Questions for self-verification

1. Features of comparison operations, their practical value.
2. Equivalent operations of comparison and their deficiencies.
3. Results of comparison and their definition.
4. To list attributes with possible values of bits and their brief performance.
5. Structure of the register of attributes and its practical use.

LIST OF REFFERENCES

1. Микропроцессоры. В 3-х книгах. Учебник для ВТУЗов/ П.В. Нестеров, В.Ф. Шаньгин, В.Д. Горбунов и др.; под ред. Л.Н. Преснухина. – М.: Высш. шк., 1986

2. Стрыгин Б.В., Щарев Л.С. Основы вычислительной, микропроцессорной техники и программирования. – М.: Высш. шк., 1989. – 479с.

3. Каган Б.М., Сташин В.В. Микропроцессоры в цифровых системах. - М.: Энегрия, 1979.

4. Грибанов В.П. и др. Операционные системы: Учеб. пособие М.: Финансы и статистика, 1990.-239с.

5. Справочник по микропроцессорным устройствам. /А.А. Молчанов, В.И. Корнейчук, В.П. Тарасенко, Д.А. Россошинский. - К.: Техника, 1987. – 288с.

6. Погорелый С.Д., Слободянюк Т.Ф. Программное обеспечение микропроцессорных систем: Справочник. – 2-е изд. К.: Техника, 1989. – 301с.

7. Токхайм Р. Микропроцессоры: курс и упражнения, /пер. с англ., под ред. В.Н. Грасевича. - М.: Энергоатомиздат, 1987. – 336с.

8. Гилмор Ч. Введение в микропроцессорную технику. - М.: Энергоатомиздат, 1983.-464с.

9. Майоров С.А. и др. Введение в микроЭВМ. - Л.: Машиностроение, 1988.- 304с.

10. Майоров В.Г., Гаврилов А.И. Практический курс программирования микропроцессорных систем. - М.: Машиностроение, 1989. – 272с.

FIRST TETRAD	SECOND TETRAD																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NOP	LXI BC	STAX BC	INX BC	INR B	DCR B	MVI B	RLC	--	DAD BC	LDA X BC	DCX BC	INR C	DCR C	MVI C	RRC	0
1	--	LXI DE	STAX DE	INX DE	INR D	DCR D	MVI D	RAL	--	DAD DE	LDA X DE	DCX DE	INR E	DCR E	MVI E	RAR	1
2	--	LXI HL	SHLD HL	INX HL	INR H	DCR H	MVI H	DAA	--	DAD HL	LHLD HL	DCX HL	INR L	DCR L	MVI L	CMA	2
3	--	LXI SP	STA adr	INX SP	INR M	DCR M	MVI M	STC	--	DAD SP	LDA adr	DCX SP	INR M	DCR M	MVI M	CMC	3
4	MOV B, B	MOV B, C	MOV B, D	MOV B, E	MOV B, H	MOV B, L	MOV B, M	MOV B, A	MOV C, B	MOV C, C	MOV C, D	MOV C, E	MOV C, H	MOV C, L	MOV C, M	MOV C, A	4
5	MOV D, B	MOV D, C	MOV D, D	MOV D, E	MOV D, H	MOV D, L	MOV D, M	MOV D, A	MOV E, B	MOV E, C	MOV E, D	MOV E, E	MOV E, H	MOV E, L	MOV E, M	MOV E, A	5
6	MOV H, B	MOV H, C	MOV H, D	MOV H, E	MOV H, H	MOV H, L	MOV H, M	MOV H, A	MOV L, B	MOV L, C	MOV L, D	MOV L, E	MOV L, H	MOV L, L	MOV L, M	MOV L, A	6
7	MOV M, B	MOV M, C	MOV M, D	MOV M, E	MOV M, H	MOV M, L	HLT	MOV M, A	MOV A, B	MOV A, C	MOV A, D	MOV A, E	MOV A, H	MOV A, L	MOV A, M	MOV A, A	7
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC	ADC E	ADC H	ADC L	ADC M	ADC A	8
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A	9
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A	A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A	B
C	RNZ	POP BC	JNZ adr	JMP adr	CNZ adr	PUSH BC	ADI data	RST0	RZ	RET	JZ adr	--	CZ adr	CAL L adr	ACI data	RST1	C
D	RNC	POP DE	JNC adr	OUT port n	CNC adr	PUSH DE	SUI data	RST2	RC	--	JC adr	IN port n	CC adr	--	SBI data	RST3	D
E	RPO	POP HL	JPO adr	XTHL	CPO adr	PUSH HL	ANI data	RST4	RPE	PCH L	JPE adr	XCH G	CPE adr	--	XRI data	RST5	E
F	RP	POP psw	JP adr	DI	CP adr	PUSH psw	ORI data	RST6	RM	SPH L	JM adr	EI	CM adr	--	CPI data	RST7	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

65