ГБОУ СПО СПб Колледж информационных технологий
МДК 01.01. Системное программирование
Учебная практика

Отчет по лабораторной работе

По дисциплине МДК 01.01 Системное программирование

Работу выполнил:

Студент группы 22

Кеворков Алексей

Преподаватель

Смирнова Ирина Петровна

Санкт-Петербург
2014 год

Лабораторная работа 1

Цель: Изучение архитектуры микропроцессора КР580.

Содержание отчета:

1. Используя справку к эмулятору МП-системы изучить структуру МП и описать его компоненты в виде:

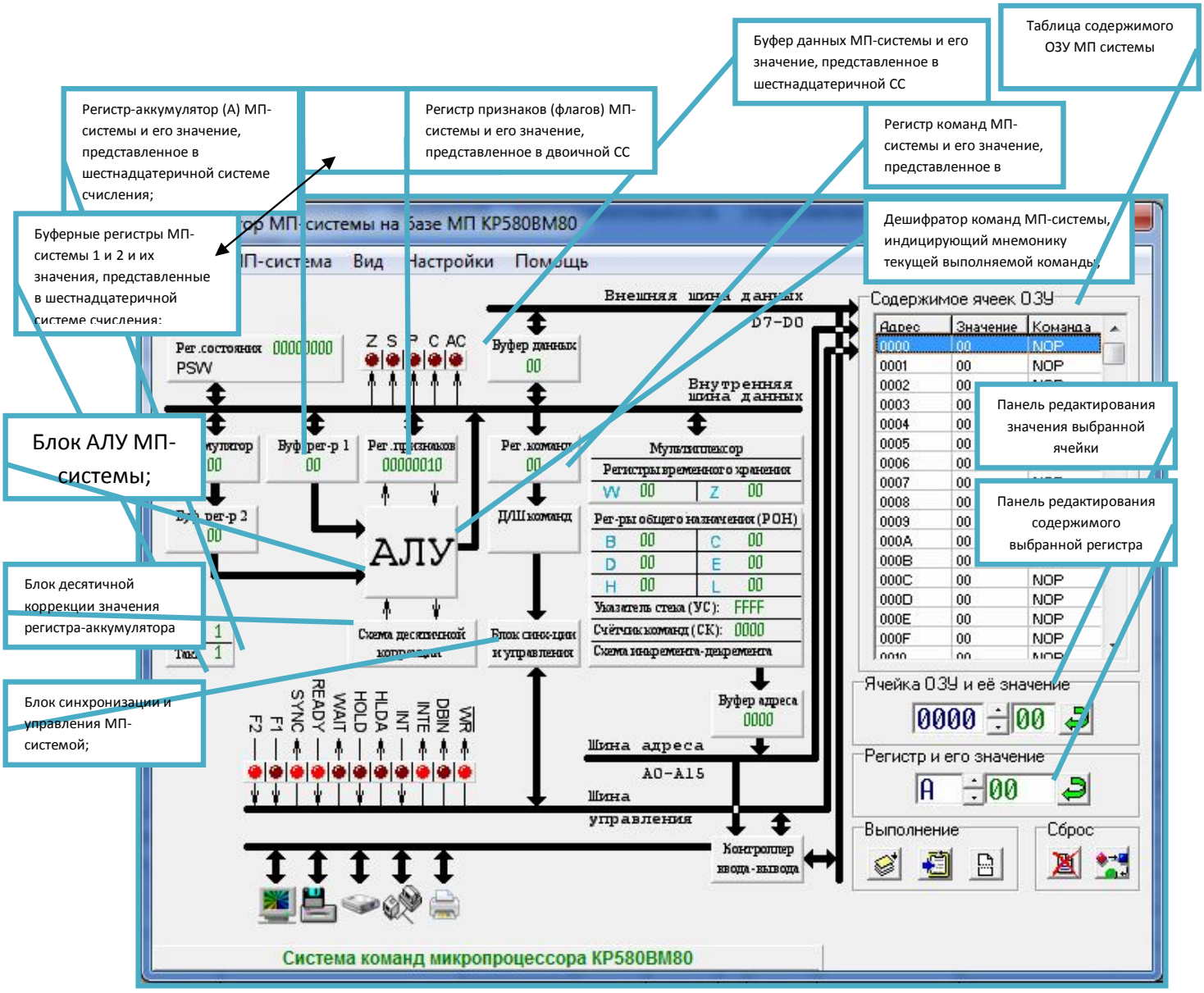
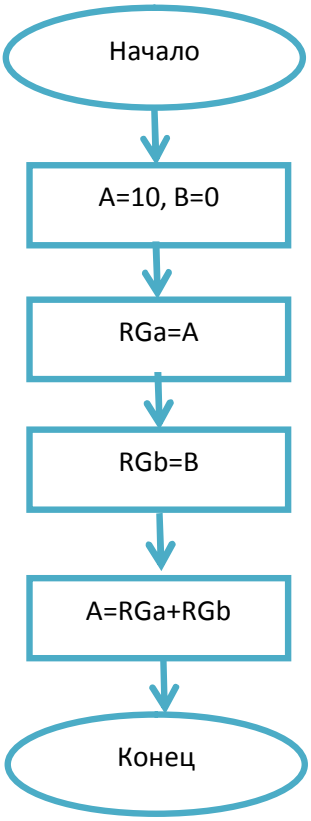


Рисунок 1- справка к эмулятору МП

1. Таблица регистров, которые имеет МП:

Регистр	Назначение	Разрядность
A	Используется в подавляющем большинстве команд логической и арифметической отработки. Обычно он адресуется неявно и служит как источником операнда, так и приемником результата.	8
РОН: B, C, D, E,	Регистры общего назначения	
HL	Адресный регистр	16
SP	Регистр счетчика команд и указателя стека	16
PC	Регистр счетчика команд и указателя стека	16

2. Блок-схема функционирования МП во время выполнения команды сложения содержимого аккумулятора и регистра B, имеющая мнемоническое обозначение ADD B.



3. Таблица флагов:

Обозначение флага	Признак флага	Условие установки флагов
M	M (Minus) - признак отрицательного результата;	устанавливается, если знаковый бит результата операции (седьмой разряд аккумулятора) равен 1, иначе сбрасывается;
Z	Z (Zero) - признак нуля;	устанавливается, если результат операции в аккумуляторе равен нулю, иначе сбрасывается;
AC	AC (Auxiliary Carry) - признак половинного переноса.	устанавливается при наличии переноса из третьего разряда аккумулятора в четвертый, иначе сбрасывается;
P	P (Parity) - признак паритета/четности;	устанавливается, если результат операции в аккумуляторе содержит четное число единиц, иначе сбрасывается;
C	C (Carry)- признак переноса;	устанавливается при наличии переноса (при сложении) или заема (при вычитании) из старшего разряда аккумулятора, иначе сбрасывается.

4. Вывод

В данной лабораторной работе мы изучили архитектуру микропроцессора КР580.

Лабораторная работа 2. Исследование команд прямой адресации

1.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для прямой адресации

Пример: LDA 000A (загрузить в аккумулятор байт, заданный адресом 000B (Load Accumulator)).

Составим программу:

```
0000  3A     LDA Ah
0001  0A
0002  00
0003  32     STA Bh      ;загрузить содержимое аккумулятора в
0004  0B                 ;ячейку памяти с адресом Bh
0005  00
0000  00
…
0009  00
000A 10
```

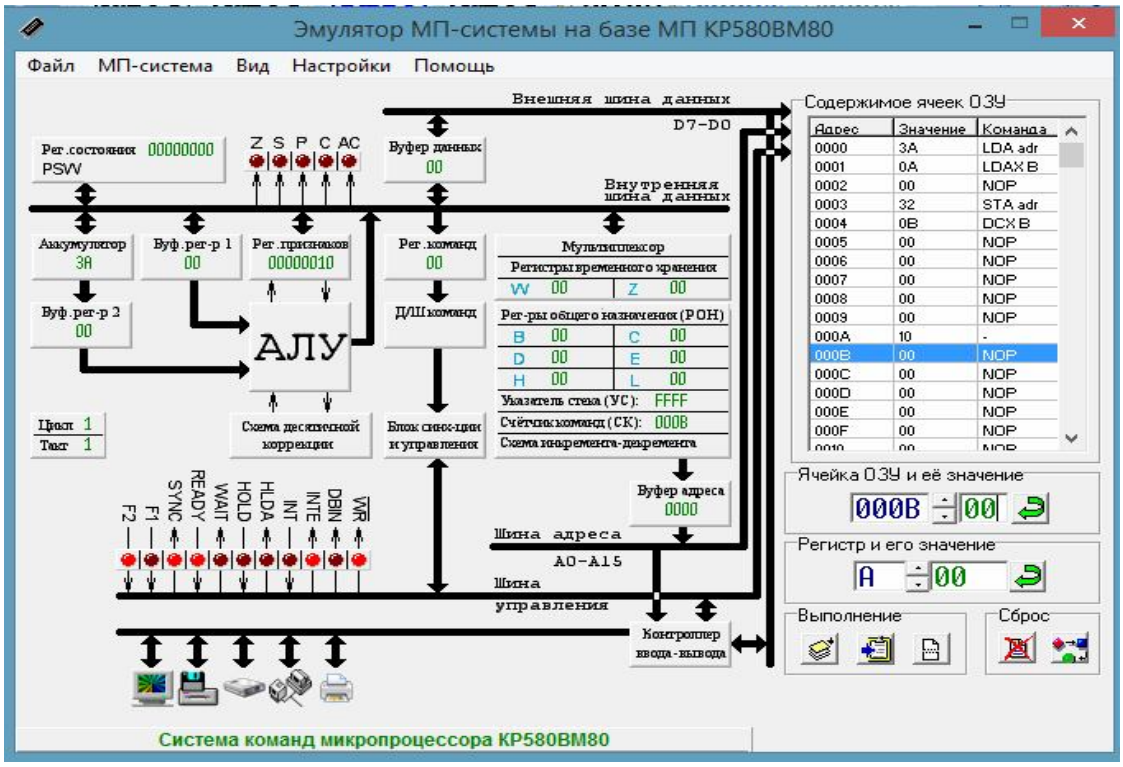


Рисунок 2 – Ввод операндов при прямой адресации

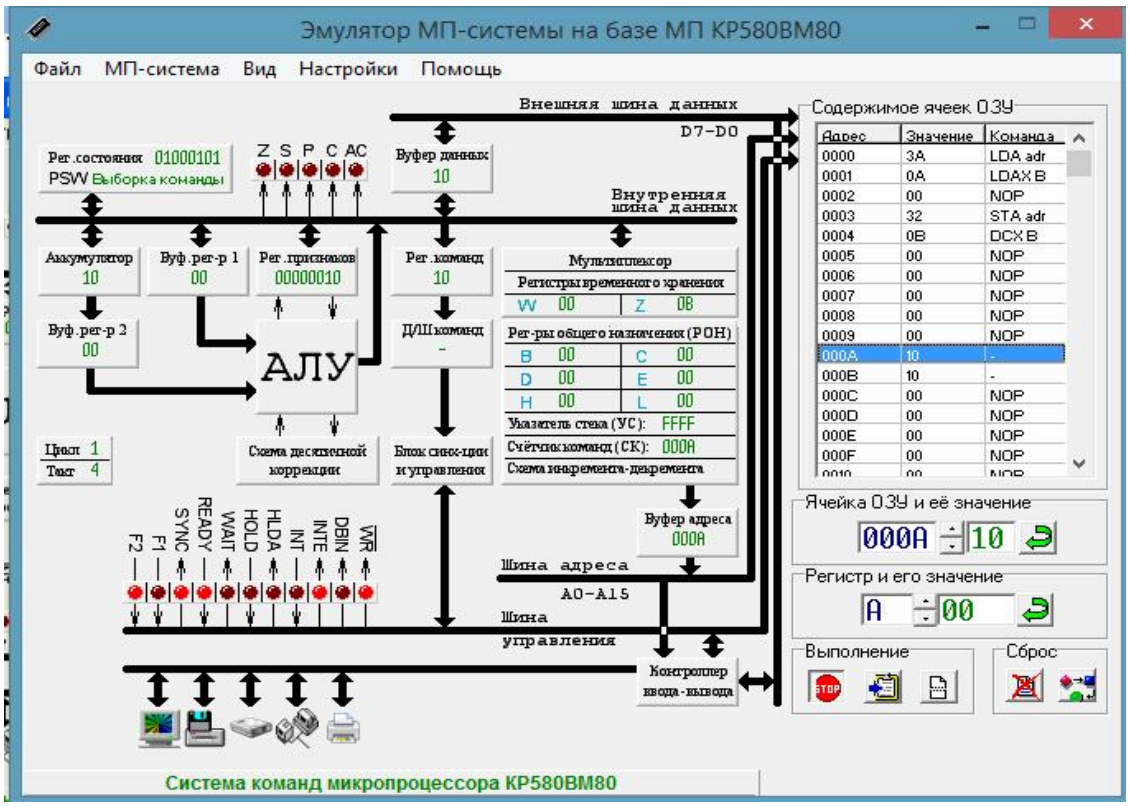


Рисунок 3 – Результат работы модели при прямой адресации

(в аккумуляторе число 10h, в ячейке 000B число 10h)

Лабораторная работа 3

Исследование команд непосредственной адресации

2.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для непосредственной адресации

MVI A, 20h. (возможно использование регистров A, B, C, D, E, H, L)

Составим программу:

0000 26 MVI A, 20h

0001 20

0002 06 MVI B, 5h

0003 5

0004 26 MVI H, d1h

0005 D1

0006 FF RST 7

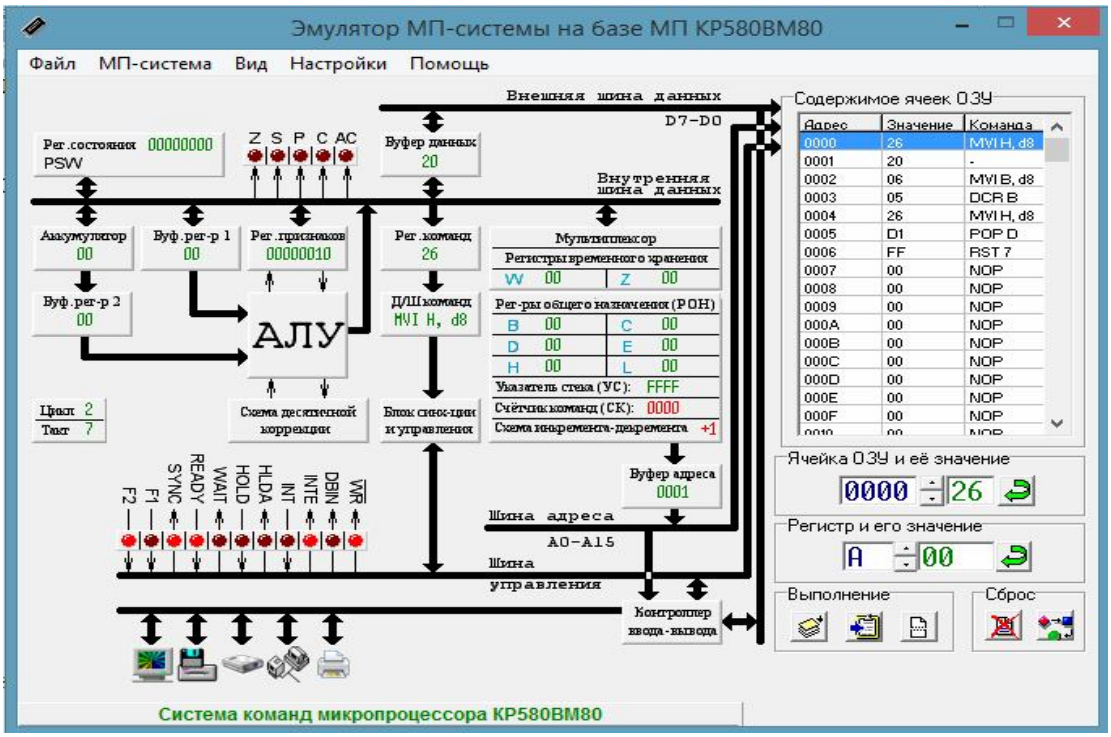


Рисунок 4 – Ввод операндов при непосредственной адресации

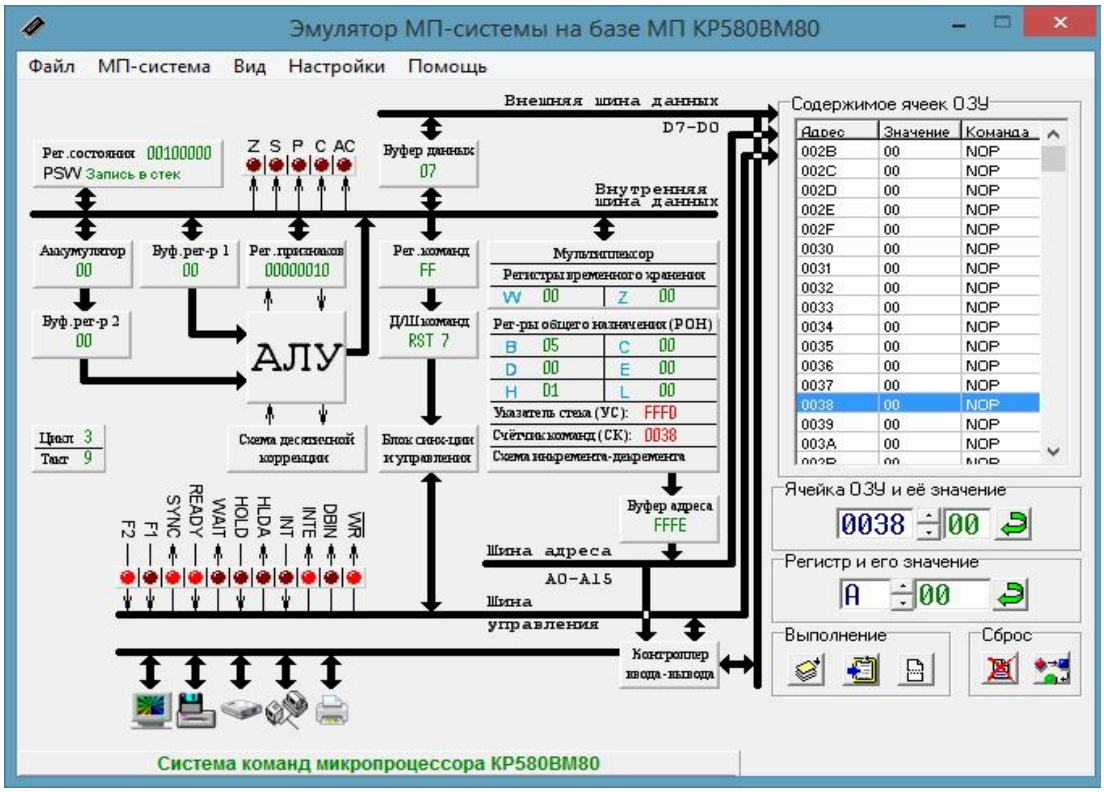


Рисунок 5 – Результат работы модели при непосредственной адресации

Лабораторная работа 4

Исследование команд косвенной адресации

3.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для косвенной адресации

Составим программу:

0000 06 MVI B, 00h

0001 00

0002 0E MVI C, 0Ah

0003 0A

0004 0A LDAX B

0005 FF RST 7

…

000A 10

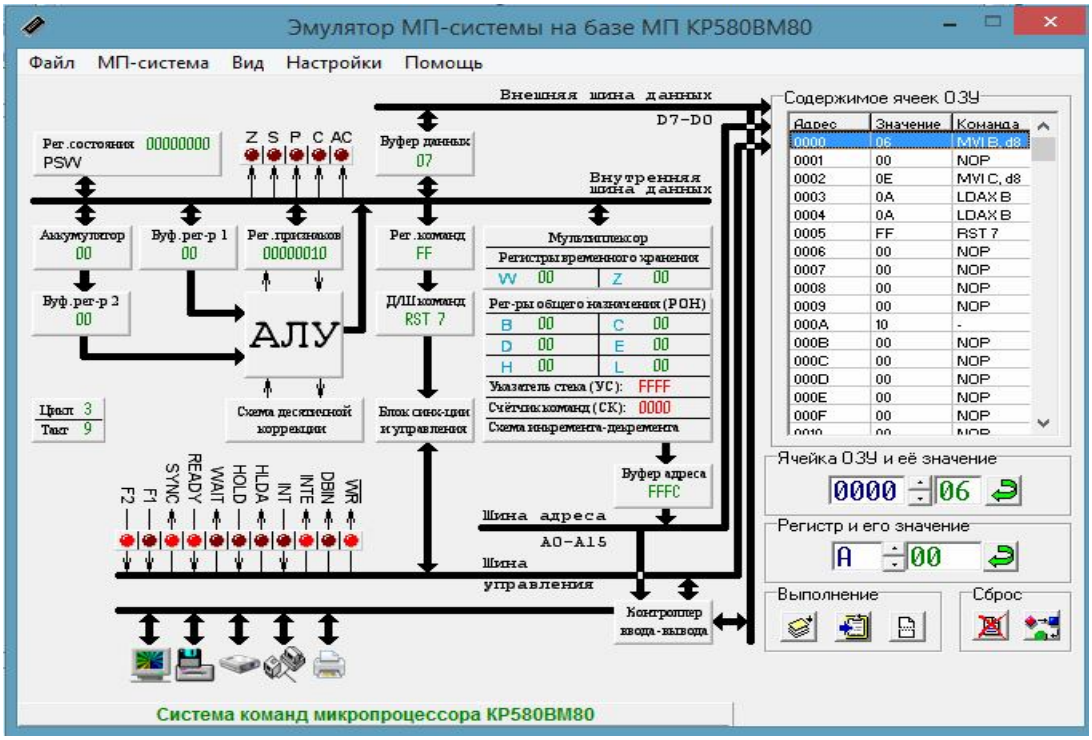


Рисунок 6 – Ввод операндов при непосредственной адресации

Значение адреса загружено в регистровую пару HL (00|0A), после чего содержимое адреса загружается в аккумулятор

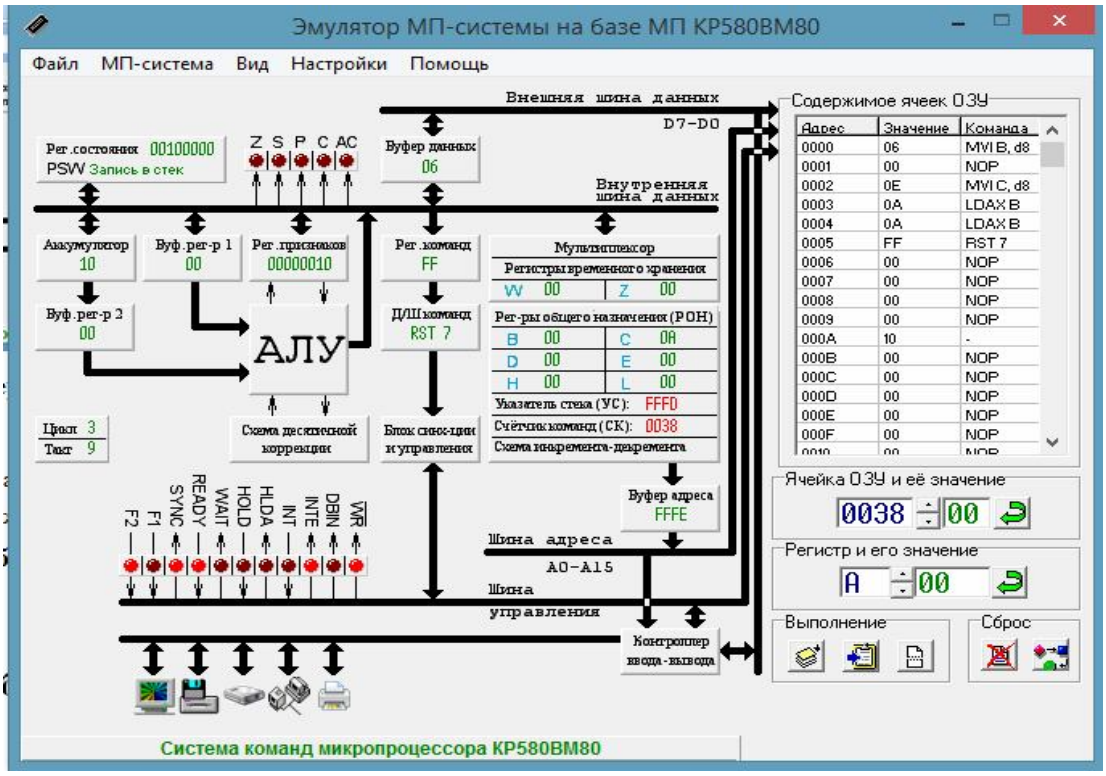


Рисунок 7 – Результат работы модели при непосредственной адресации

Лабораторная работа 5

Исследование команд стековой адресации

4.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для стековой адресации

4.2 Содержание работы

Источником или приёмником операнда является ячейка стековой памяти. Адрес вершины стека находится в указателе стека SP. При заполнении следующей ячейки указатель стека инкриминируется

Составим программу:

0000 06 MVI B, 10h

0001 10

0002 16 MVI D, 5Ah

0003 5A

0004 C5 PUSH B

0005 D5 PUSH D

0006 50 MOV D, B

0007 D1 POP D

0008 C1 POP B

0009 FF RST7

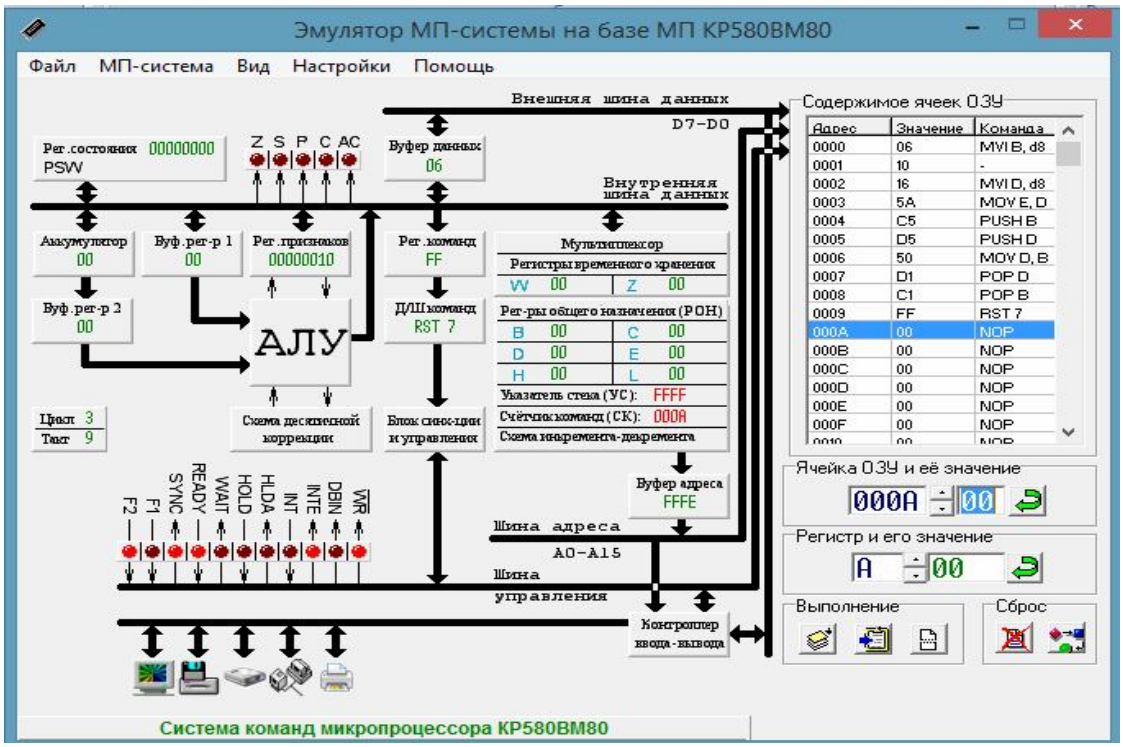


Рисунок 8 – Ввод операндов при стековой адресации

Данные помещаются в регистры B и D, после чего их содержимое посылается в стек. После этого производится обмен значениями между регистрами и значения регистров возвращаются из стека.

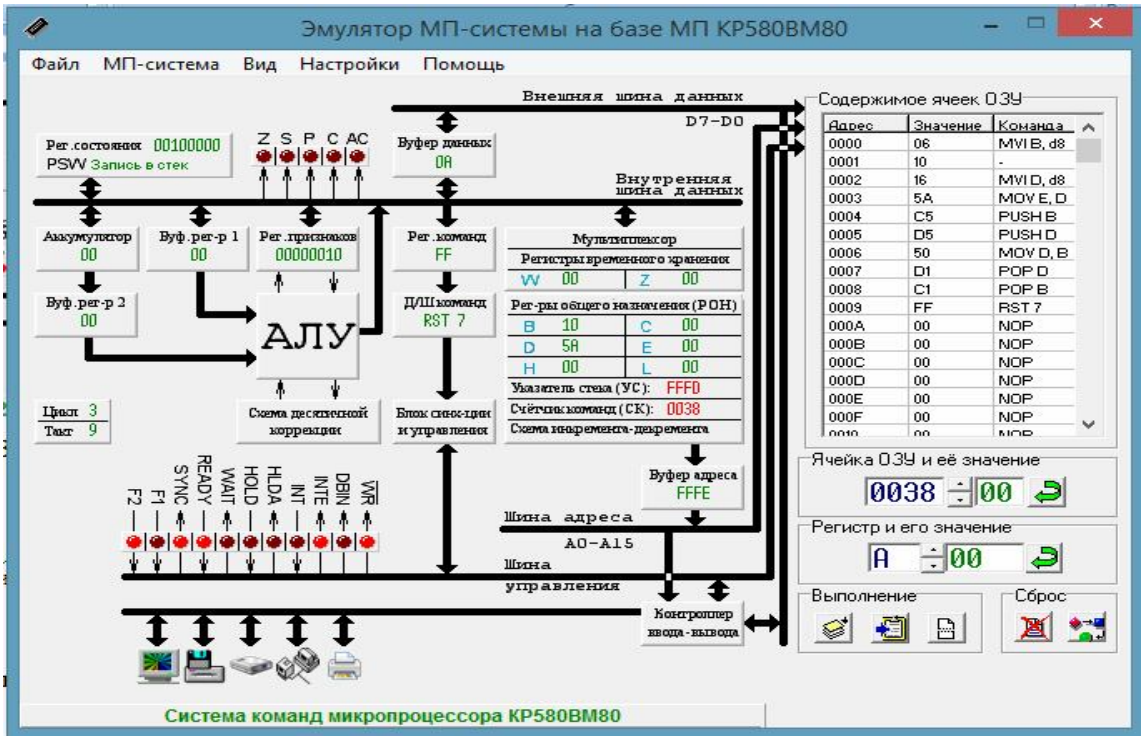


Рисунок 9 – Результат работы модели при стековой адресации

Лабораторная работа 6

Пример программы для микропроцессора

5.1 Цель лабораторной работы

Целью данной работы является ознакомление с простейшими арифметическими действиями на микропроцессоре КР580

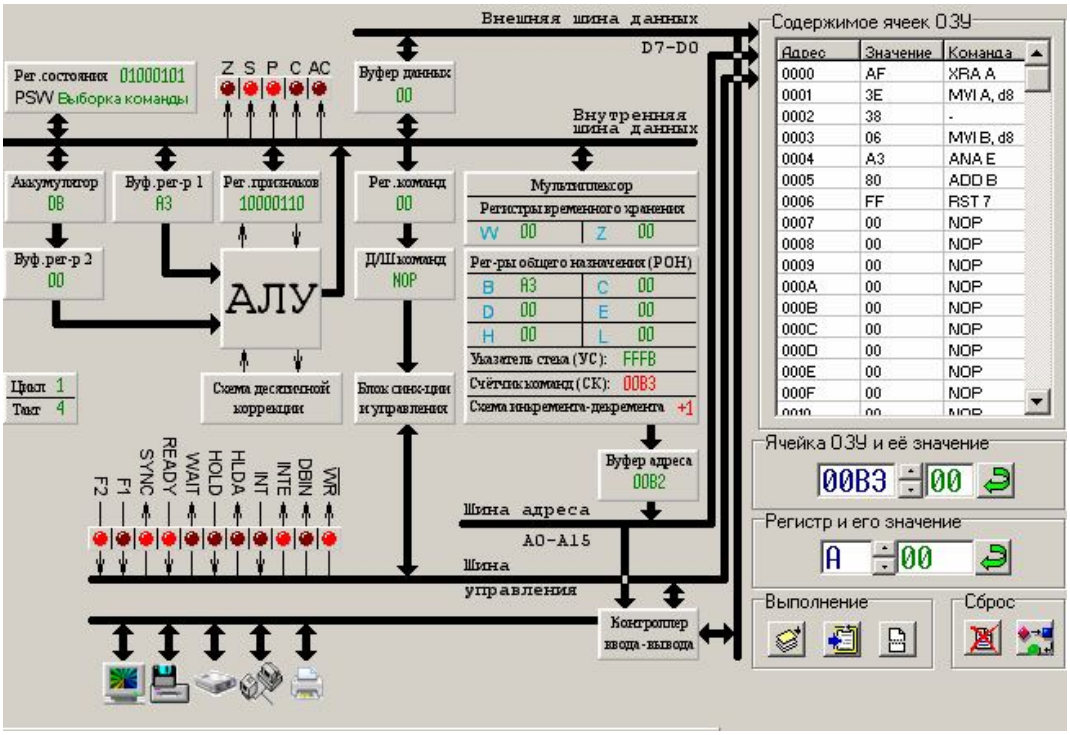
5.2 Содержание работы

Составим программу сложения (табл. 1).

Таблица 1 − Программа сложения двух однобайтных чисел

Адрес	Код команды	Метка	Мнемокод	Комментарий
0000	AF	-	XRA A	Очистить аккумулятор
0001	3E	-	MVI A, 38	Записать в аккумулятор число $56_{10} = 38h_{16}$
0002	38			
0003	06	-	MVI B,	Записать в регистр B число $163_{10} =$

0004	A3		A3	$a3h_{16}$
0005	80	-	ADD 8	Сложить $38h_{16}$ и $a3h_{16}$ ($56_{10} + 163_{10} = 219_{10} = db_{16}$)
0006	E7	-	RST 7	Прервать выполнение программы



(в аккумуляторе значение суммирования – число $219_{10} = db_{16}$)

Рисунок 10 – Простое сложение двух однобайтных чисел

Для получения разности двух чисел X и Y можно использовать эту же программу, заменив в ней по адресу 0005 код команды 80 (ADD B) кодом 90 (SUB B) команды вычитания содержимого регистра B из содержимого аккумулятора, разместив предварительно в регистрах B и A соответственно вычитаемое и уменьшаемое. Разность будет записана в аккумуляторе.

Цель лабораторной работы - рассмотреть особенности выполнения простейших арифметических операций над целыми числами без знака на эмуляторе МП К580, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и кнопками эмулятора.

Лабораторная работа 7

Задача: выполнить запись однобайтных чисел в регистры A и B, сложить их и поместить результат сложения в аккумулятор.

$66_{10}(42_{16}) + 129_{10}(81_{16}) = 195_{10}(C3_{16})$

Таблица с текстом программы и комментариями

Адрес	Код команды	Метка	Мнемокод	Комментарий
0000	AF	-	XRA A	Очистить аккумулятор
0001	3E	-	MVI A, 38	Записать в аккумулятор
0002	42	-		число X ($66_{10} = 42_{16}$)
0003	06	-	MVI B, A3	Записать в регистр B
0004	81	-		число У ($129_{10} = 81_{16}$)
0005	80	-	ADD B	Сложить X и Y Сложить 42_{16} и 81_{16} ($66_{10} + 129_{10} = 195_{10} = C3_{16}$)
0006	32	-	STA adr	Записать содержимое аккумулятора в
0007	20	-		ячейку 0020
0008	00	-		-
0009	E7	-	RST 7	Прервать выполнение программы

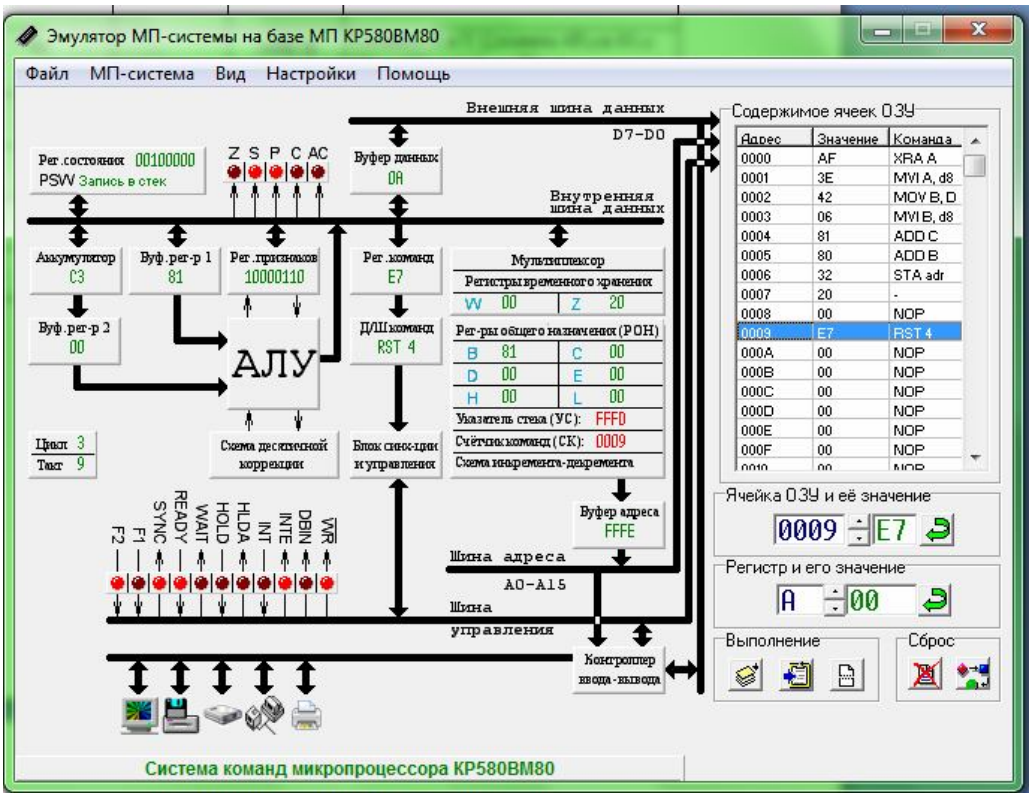


Рисунок 11 – Ввод значений

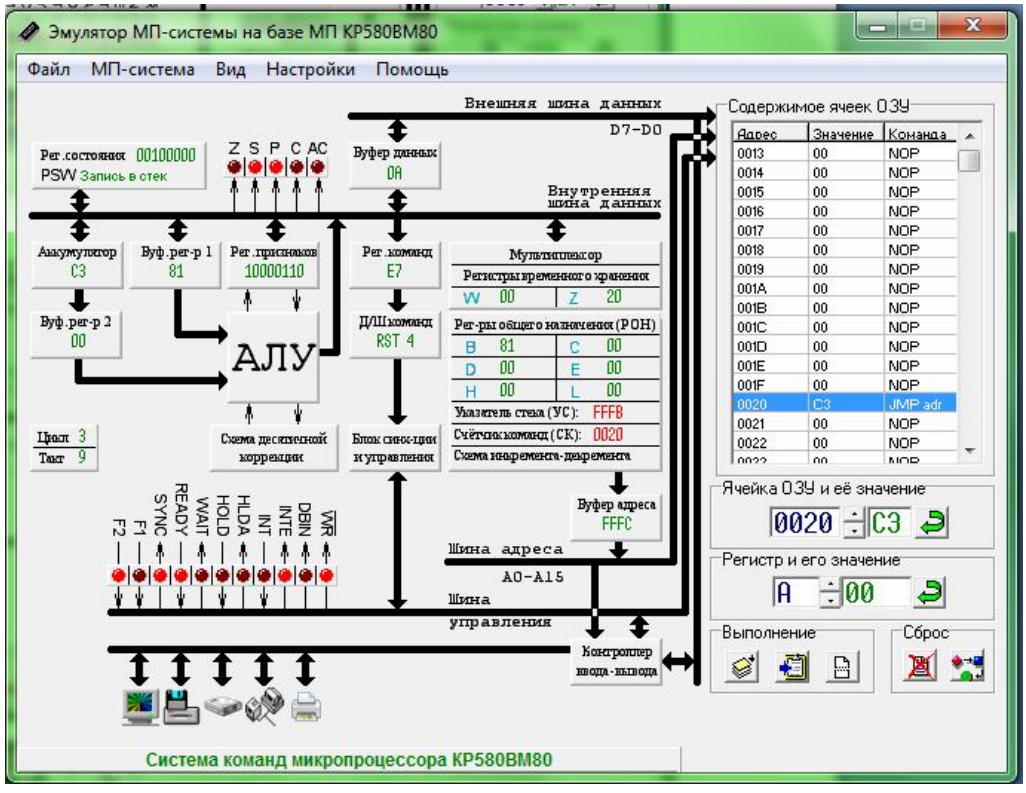


Рисунок 12 - Результат работы, помещённый в ячейку 0020

Лабораторная работа 8

Задача: выполнить запись однобайтных чисел в регистры A и B, получить их разность и поместить результат разности в аккумулятор.

$129_{10}(81_{16})$ - $67_{10}(43_{16})$= $62_{10}(3E_{16})$

Таблица с текстом программы и комментариями

Адрес	Код	Метка	Мнемок	Комментарий
0009	E7	-	RST 4	Прервать выполнение программы
0001	3E	-	MVI A, d8	Записать в аккумулятор
0002	81	-		Число X ($129_{10} = 81_{16}$)
0003	06	-	MVI B, d8	Записать в регистр B
0004	43	-		Число У (67_{10}=43_{16})
0005	90	-	ADD B	Вычесть Y из X Вычесть 43_{16} из 81_{16} (129_{10} - 67_{10}= 62_{10}=$3E_{16}$)
0006	32	-	STA adr	Записать содержимое аккумулятора в
0007	20	-		ячейку 0020
0008	00	-		-

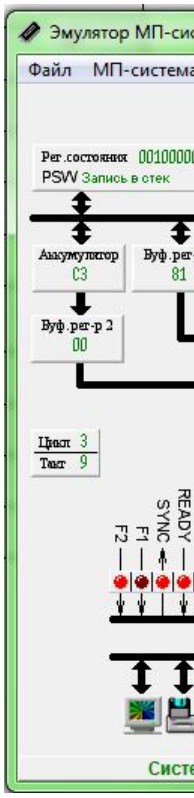


Рисунок 13 - Ввод данных

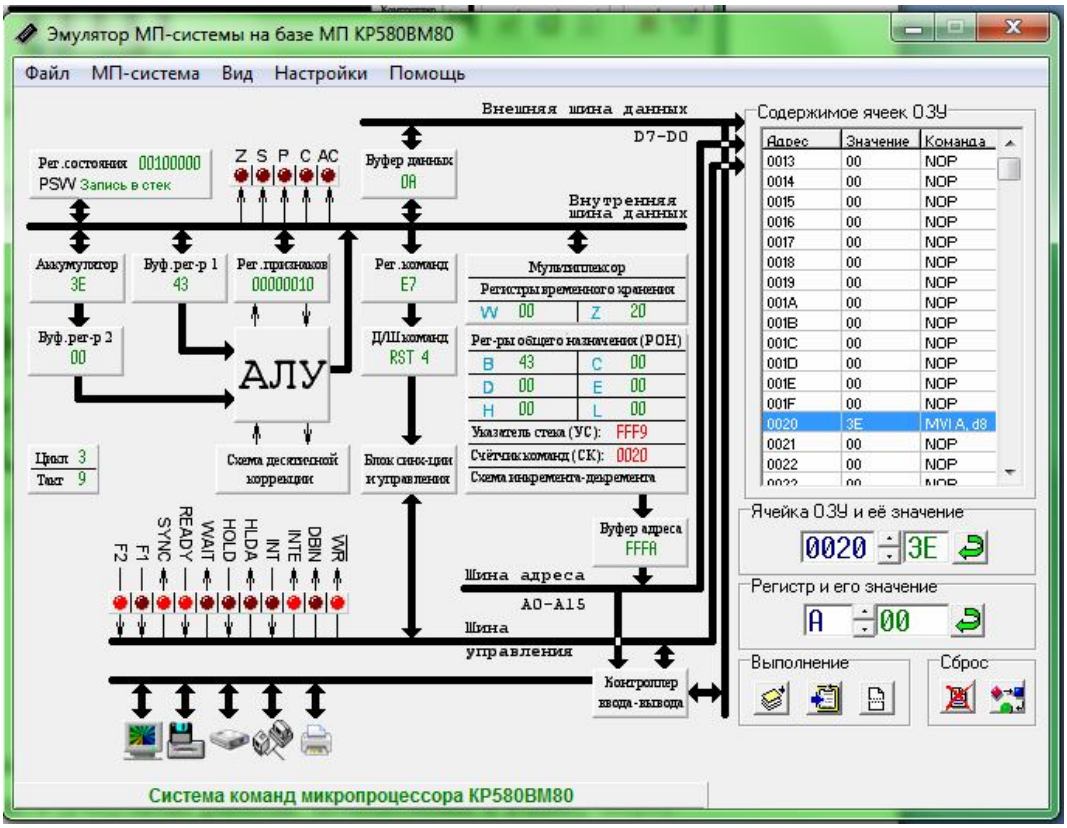


Рисунок 14 - Результат работы, помещённый в ячейку 0020

Лабораторная работа 9

Задание: Арифметические вычисления суммы в десятичной и шестнадцатеричной системе счисления с помощью массива

Цель лабораторной работы - рассмотреть особенности выполнения простейших арифметических операций над целыми числами без знака на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

$66_{10}(42_{16}) + 129_{10}(81_{16}) + 41_{10}(29_{16}) + 103_{10}(67_{16}) + 16_{10}(10_{16}) = 355_{10}(163_{16})$

Таблица с текстом программы и комментариями

Адрес	Код	Метка	Мнемоника	Комментарий

	команды			
0000 0001 0002	21 30 00	PRG 2:	LXI H,d16	Загрузить в регистры HL, адрес первого слагаемого
0003 0004	0E 05		MVI C, d8	Загрузить в регистр С количество слагаемых
0005	AF		XRA A	Очистить аккумулятор
0006	47		MOV B, A	Очистить регистр В
0007	86	M1:	ADD M	Прибавить к содержимому аккумулятора число из массива слагаемых
0008 0009 000A	D2 0D 00		INC M2	Если переноса нет, то идти на M2
000B 000C	04 B7		INR B ORA A	Увеличить содержимое регистра В на 1 Очистить флаг переноса
000D	23	M2:	INX H	Указать на следующий адрес слагаемого
000E	0D		DCR C	Уменьшить содержимое регистра С на 1
000F 0010 0011	C2 07 00		INZ	Если не все слагаемые, то идти на M1
0012	32		STA adr	Сохранение аккумулятора
0013	21		LXI H, d16	Записать содержимое аккумулятора в
0014	00			ячейку 0021
0015	AF		XRA A	Очистить аккумулятор

0016	80		ADD B	Прибавить к содержимому аккумулятора число из регистра B
0017	32		STA adr	Сохранение аккумулятора
0018	20			Записать содержимое аккумулятора в
0019	00			ячейку 0020
001A	FF		RST 7	Прервать выполнение программы
. . .				
0030	42		MOV B, D	
0031	81		ADD C	
0032	29		DAD H	Массив
0033	67		MOV H, A	
0034	10			

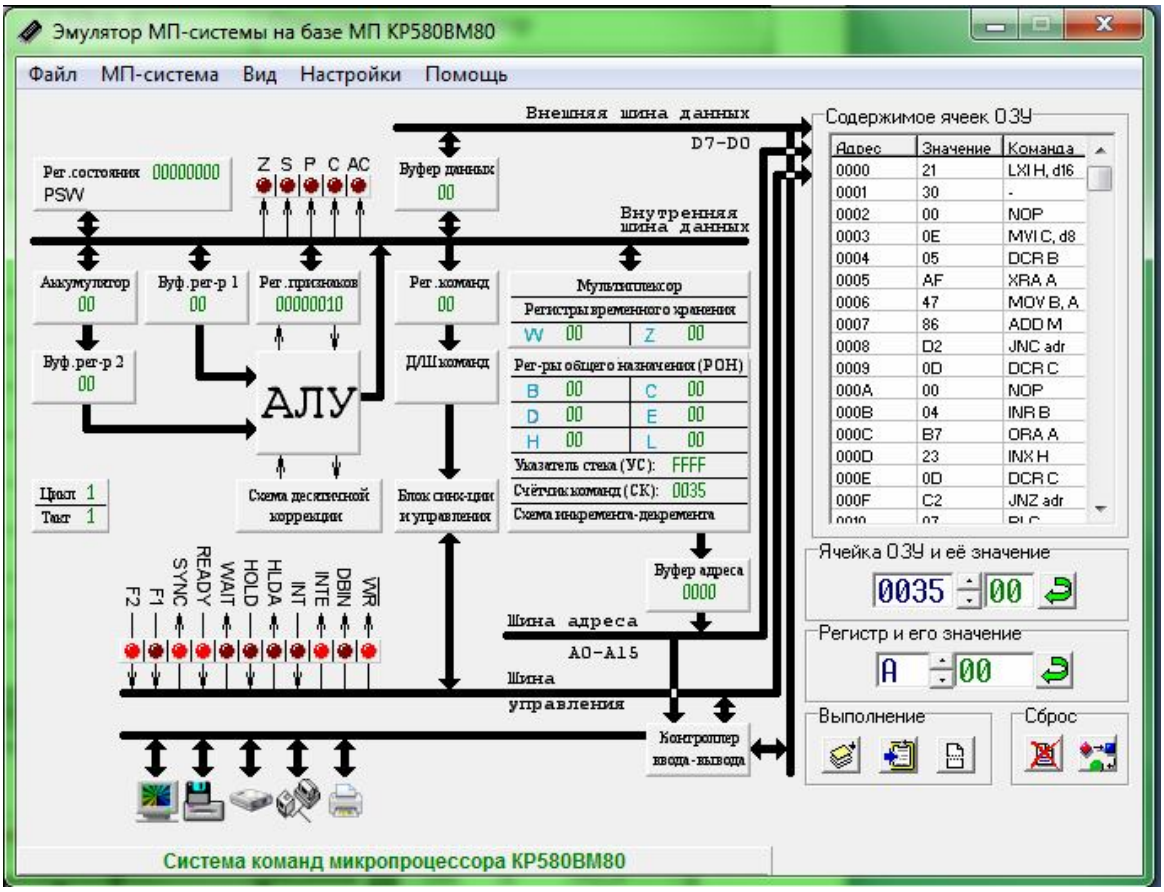


Рисунок 15 – Ввод данных

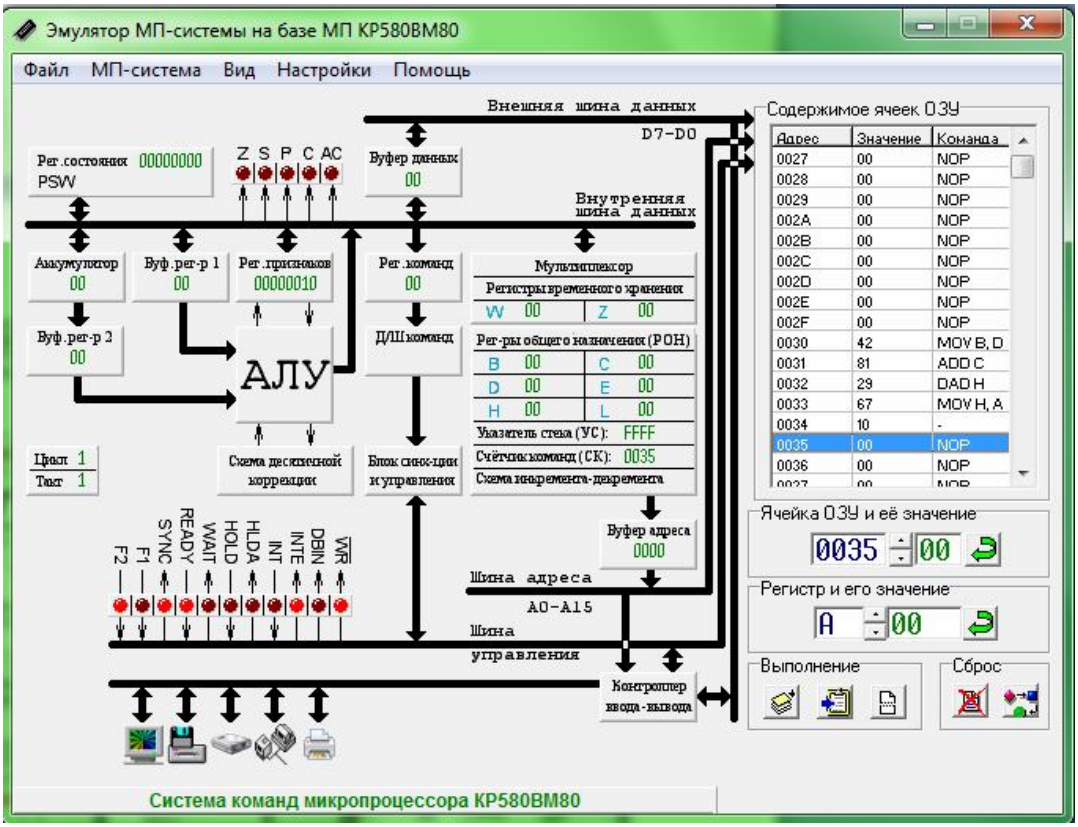


Рисунок 16 – Ввод данных

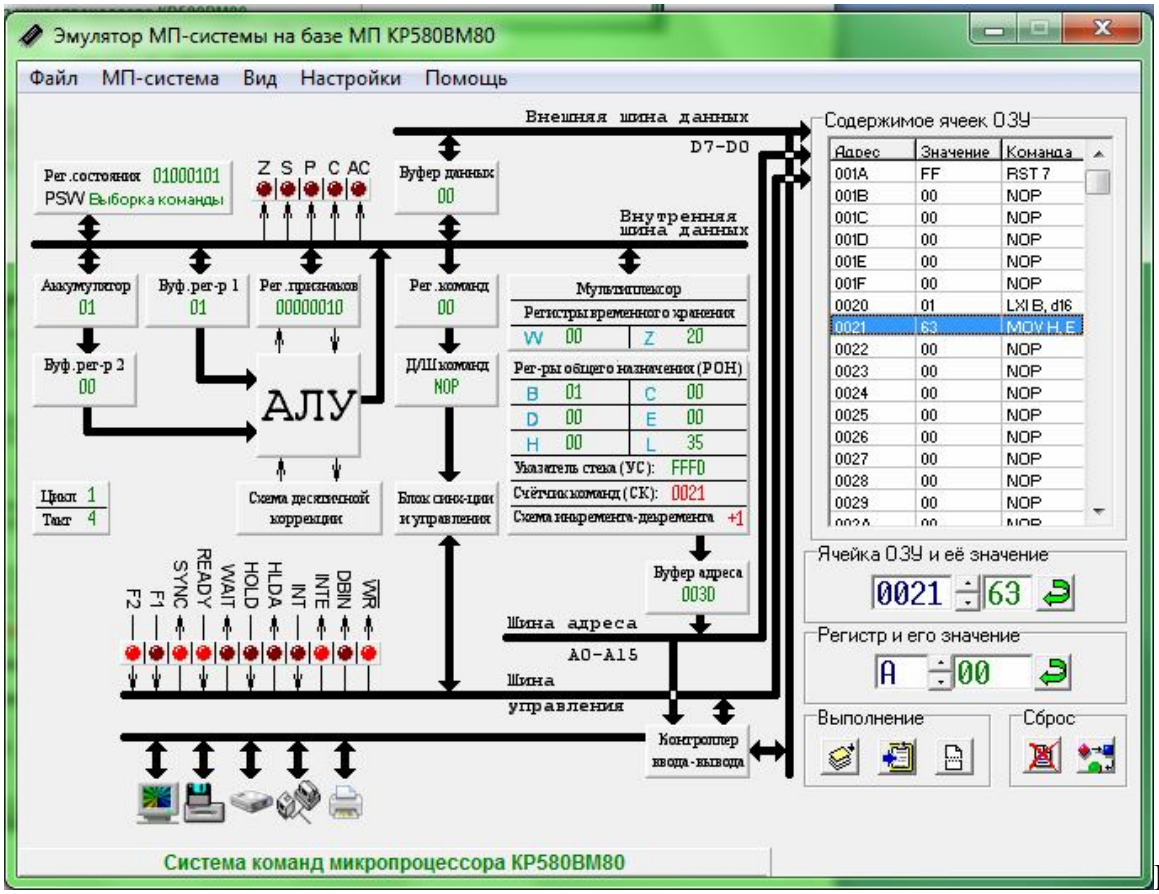


Рисунок 17 – Результат работы, помещённый в ячейки 0020 и 0021

Лабораторная работа 10

Цель: Выполнить операцию сложения двух двухбайтных чисел

$4309_{10}(10D5_{16})+ 2189_{10}(88D_{16})=6498_{10}(1962_{16})$

Таблица с текстом программы и комментариями

Адрес	Код	Мнемоника	Комментарий
0000	3E	MVI A,d8	Младший байт первого слагаемого заносится в аккумулятор $0C_{16}$
0001	D5		

			→A
0002	06	MVI B,B4	Младший байт второго слагаемого заносится в регистр B 04_{16} →B
0003	8D		
0004	80	ADD B	Регистр B + аккумулятор
0005	32	STA adr	Содержимое аккумулятора засылается
0006	20		в ячейку 0020
0007	00		
0008	3E	MVI A,d8	Старший байт первого слагаемого заносится в аккумулятор $0C_{16}$ →A
0009	10		
000A	06	MVI B,d8	Старший байт второго слагаемого заносится в регистр B 07_{16} →B
000B	08		

000C	88	ADC B	Регистр B+ аккумулятор+ перенос
000D	32	STA adr	Содержимое аккумулятора засылается
000E	21	LXI H, d16	в ячейку 0021
000F	00		
0010	76	HLT	Остановка процесса

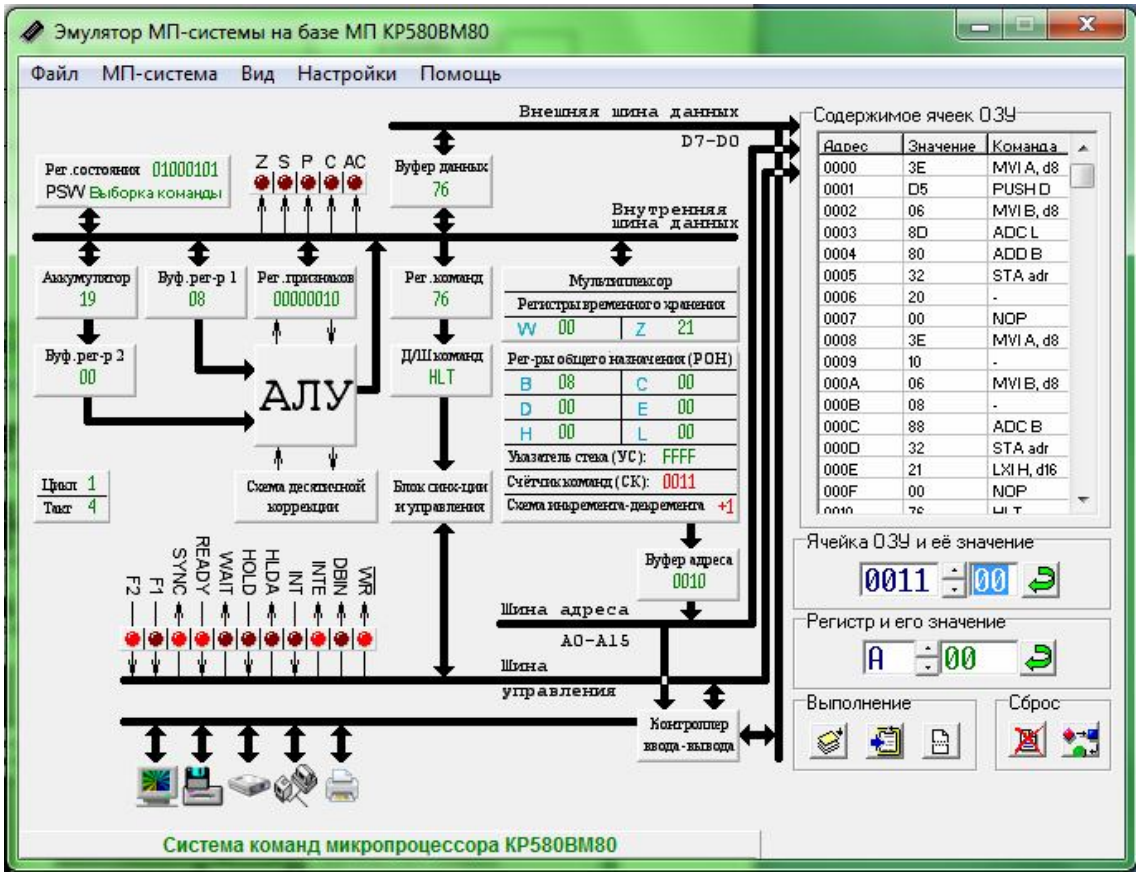


Рисунок 18 – Ввод данных

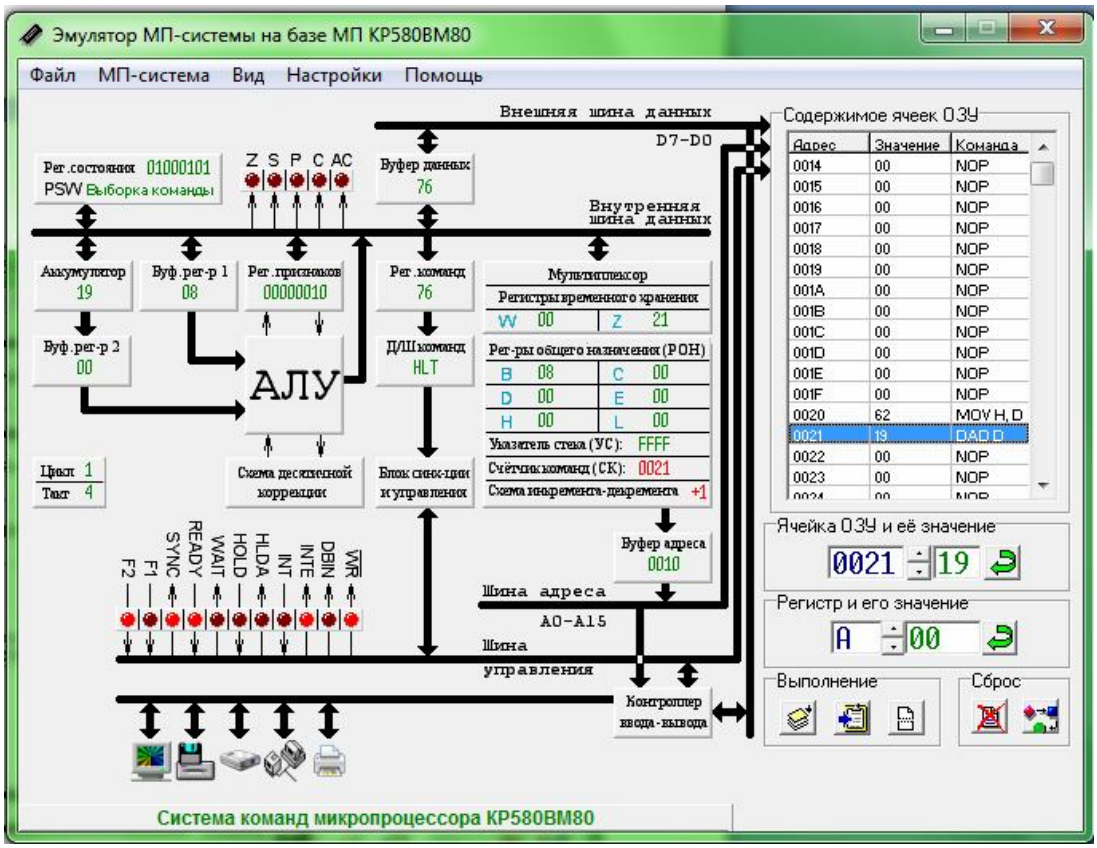


Рисунок 19 – Результат работы, помещённый в ячейки 0020 и 0021

Лабораторная работа 11

Цель: Выполнить операцию вычитания двух двухбайтных чисел.

$45900_{10}(B34C_{16})$- $21098_{10}(526A_{16})$= $24802_{10}(60E2_{16})$

Таблица с текстом программы и комментариями

Адрес	Код	Мнемоника	Комментарий
0000	3E	MVI A,d8	Младший байт уменьшаемого заносится в аккумулятор $B8_{16} \rightarrow$A
0001	4C		
0002	D6	SUI d8	Содержимое вычитается из содержимого аккумулятора
0003	6A		

0004	32	STA adr	Содержимое аккумулятора засылается в
0005	20		ячейку 0020
0006	00		
0007	3E	MVI A,8A	Старший байт уменьшаемого заносится в аккумулятор $4D_{16} \rightarrow A$
0008	B3		
0009	DE	MVI B,B4	Содержимое и перенос вычитаются из содержимого аккумулятора
000A	52		
000B	32	STA adr	Содержимое аккумулятора засылается в
000C	21		ячейку 0021
000D	00		
000E	76	HLT	Остановка процесса

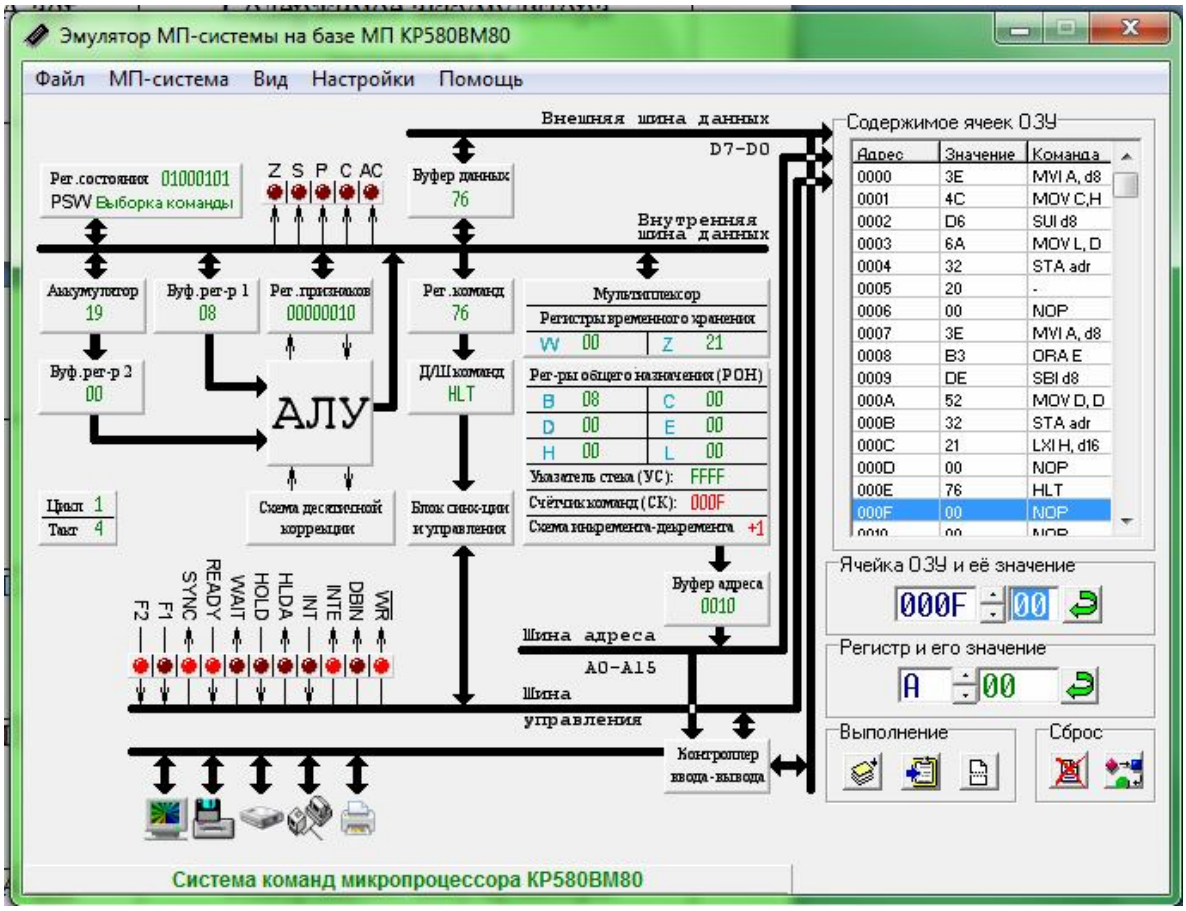


Рисунок 20 – Ввод данных

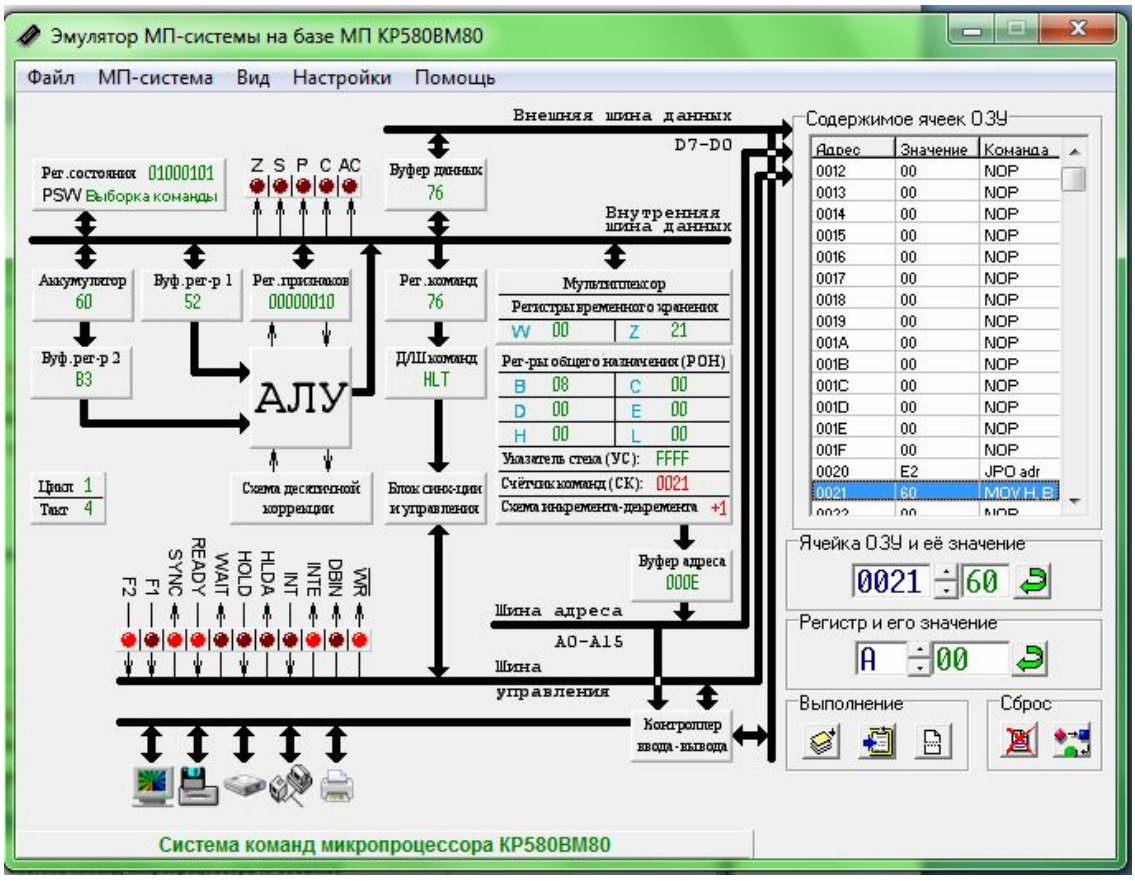


Рисунок 21 – Результат работы, помещённый в ячейки 0020 и 0021

Лабораторная работа 12

Цель: Изучение алгоритма ветвления

Задание 1:

Выполнить на эмуляторе МП КР580 программу:

Вычислить значение y:

если x>10, то y=x-4. если x ≤10, то y=3x. x=ячейка [0020], y=ячейка [0021].

```
LDA 0020
   CPI ,OA
    JM {addr}
    MOV B, A
    SUI,04
    STA 0021
    HLT
{addr}:
    MOV B,A
    ADD B
    ADD B
    STA 0021
     HLT
```

СОСТАВИТЬ БЛОК - СХЕМУ, ЛИСТИНГ (таблицу). СДЕЛАТЬ ПРОВЕРКУ ПРИ

X=04, 0A, 0D.

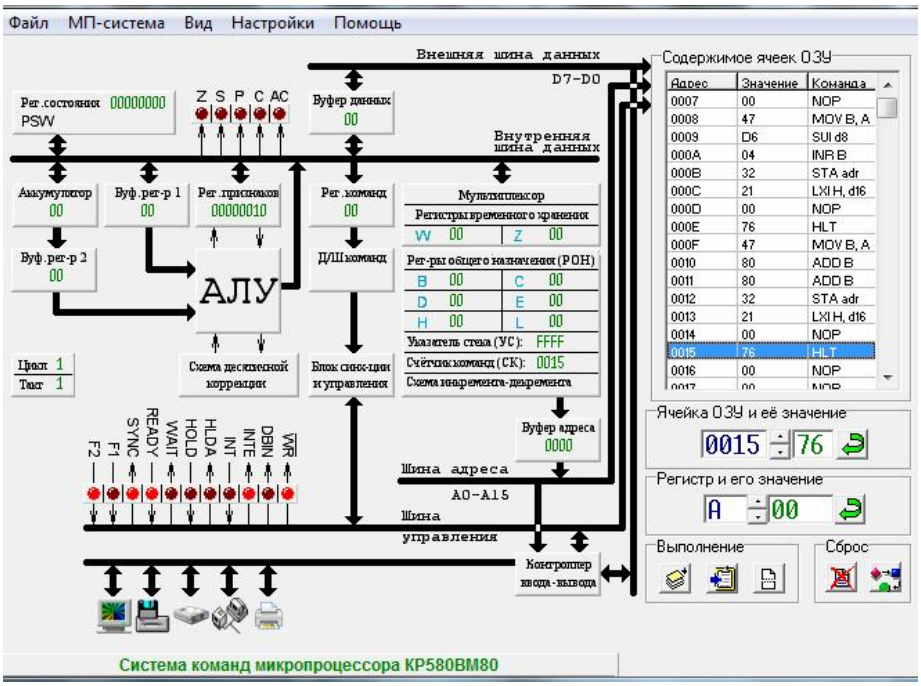


Рисунок 22 – Ввод данных

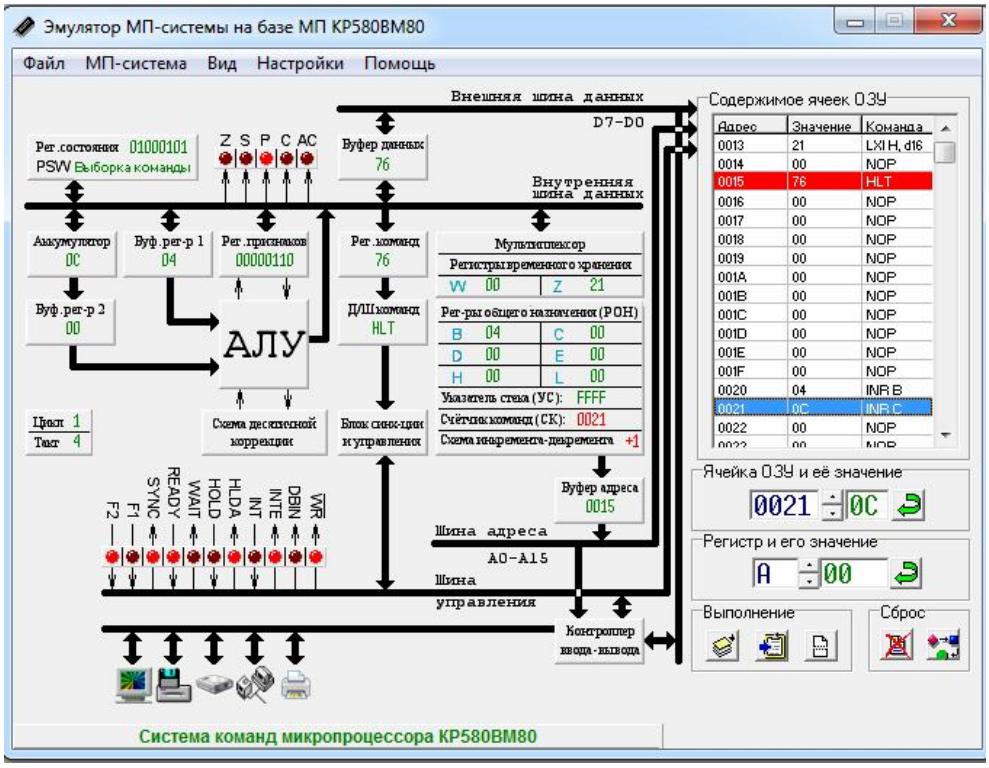


Рисунок 23 – Результат работы при x=04, помещённый в ячейку 0021

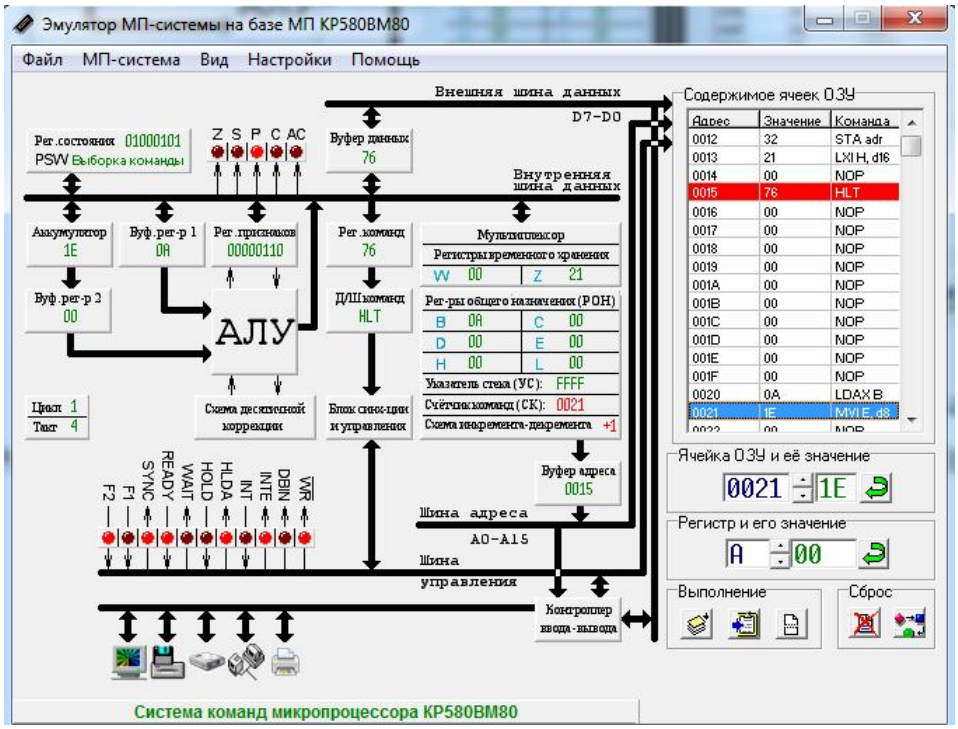


Рисунок 24 – Результат работы при x=0A, помещённый в ячейку 0021

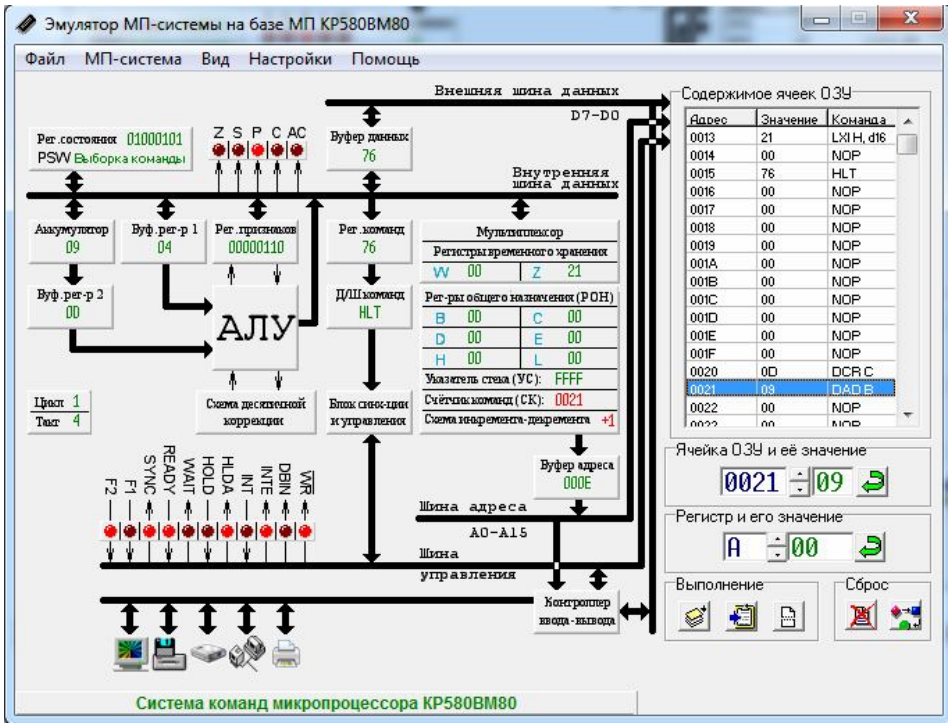


Рисунок 25 – Результат работы при x=0D, помещённый в ячейку 0021

Таблица с текстом программы и комментариями

Адрес	***Значение***	***Команда***	***Расшифровка***
0000	3A	LDA adr	Загрузка аккумулятора
0001	20		
0002	00		
0003	FE	CPI d8	Содержимое 2-го байта команды вычитается из содержимого аккумулятора
0004	0A	INR A	Число
0005	FA	JM adr	Условный переход по условию «минус»
0006	0F	RRC	
0007	00		
0008	47	MOV B,A	Передать из регистра А в регистр В
0009	D6	SUI d8	Содержимое 2-го байта команды вычитается из содержимого аккумулятора
000A	04	MVI B,d8	Содержимое 2-го байта команды засылается в регистр В
000B	32	STA adr	Сохранение аккумулятора
000C	21	LXISP,d16	
000D	00		
000E	76	HLT	Остановка процесса
000F	47	MOV B,A	Передать из регистра А в регистр В
0010	80	ADD B	Регистр В + аккумулятор
0011	80	ADD B	Регистр В + аккумулятор
0012	32	STA adr	Сохранение аккумулятора

0013	21	LXISP,d16	
0014	00		
0015	76	HLT	Остановка процесса
….			
0020	**04, 0A, 0D**		Значения X
0021			Результат

Лабораторная работа 13

Задание:

1. Вывести на экран монитора КР580 номер группы, свою фамилию и имя разными цветами.
2. Написать листинг полученной программы

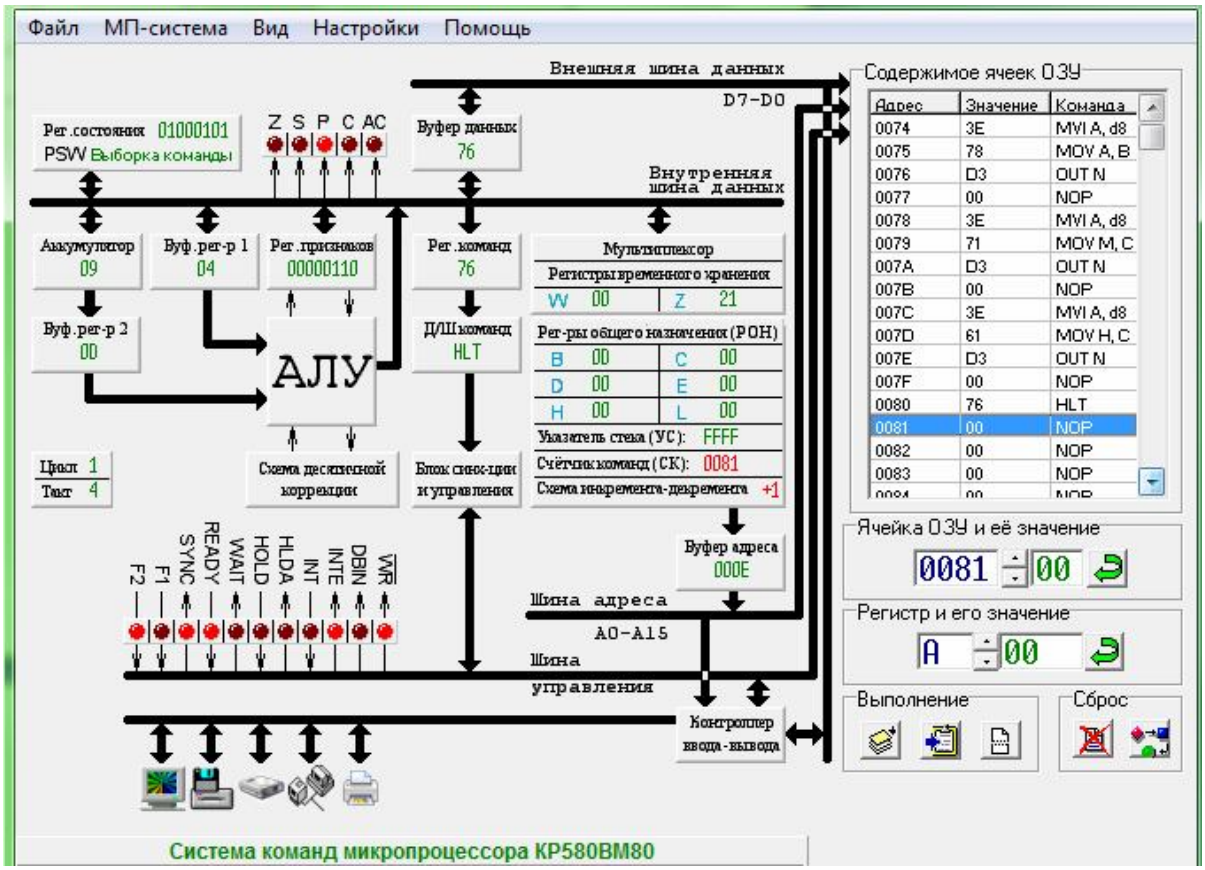


Рисунок 26 – Ввод данных

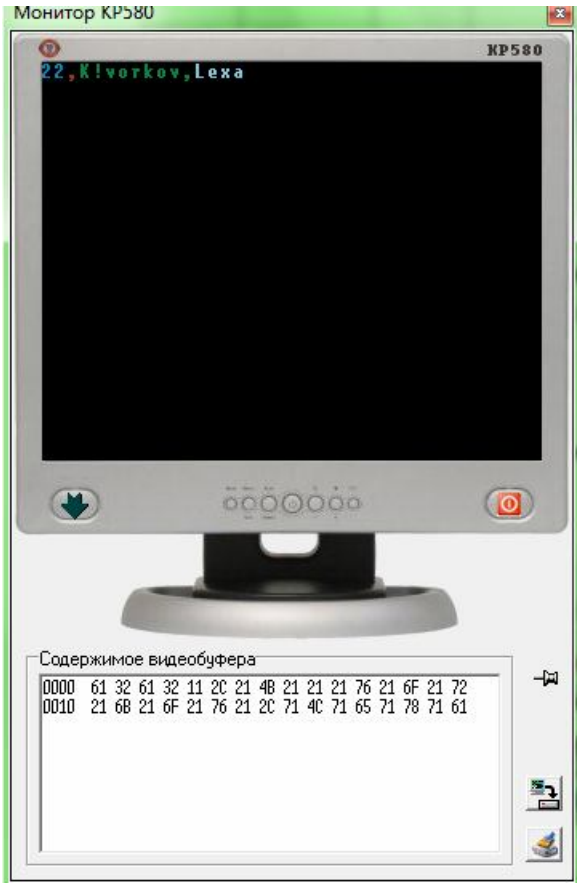


Рисунок 27 – Результат работы, выведенный на экран

Таблица с текстом программы и комментариями

адрес	код команды	метка	мнемоника и операнд	комментарии
0000	3E		MVI A,d8	21 => A, A=21
0001	61			{цвет символа в Акк}
0002	D3		OUT N	A->PORT 00,PORT 00=21
0003	00			{цвет символа в порт монитора}
0004	3E		MVI A,d8	32 => A, A=32
0005	32			{номер символа в Акк,

				2->A}
0006	D3		OUT N	A =>PORT 00,PORT 00=32
0007	00			{номер символа в порт монитора}
0008	3E		MVI A, d8	21 => A, A=21
0009	61			{цвет символа в Акк}
000A	D3		OUT 00	A =>PORT 00,PORT 00=21
000B	00			{цвет символа в порт монитора}
000C	3E		MVI A, d8	32 => A, A=32
000D	32			{номер символа в Акк, **2**->A}
000E	D3		OUT N	A->PORT 00,PORT 00=32
000F	00			{номер символа в порт монитора}
0010	3E		MVI A, 11	11 => A, A=11
0011	11			{цвет символа в Акк}
0012	D3		OUT 00	A =>PORT 00,PORT 00=11
0013	00			{цвет символа в порт монитора}
0014	3E		MVI A,6B	2C => A, A=2C
0015	2C			{номер символа в Акк, **,**->A}
0016	D3		OUT 00	A =>PORT 00,PORT

				00=2C
0017	00			{номер символа в порт монитора}
0018	3E		MVI A, 51	51 => A, A=51
0019	21			{цвет символа в Акк}
001A	D3		OUT 00	A =>PORT 00,PORT 00=51
001B	00			{цвет символа в порт монитора}
001C	3E		MVI A,56	56 => A, A=56
001D	4B			{номер символа в Акк, **K**->A}
001E	D3		OUT 00	A =>PORT 00,PORT 00=65
001F	00			{номер символа в порт монитора}
0020	3E		MVI A, 51	51 => A, A=51
0021	21			{цвет символа в Акк}
0022	D3		OUT 00	A =>PORT 00,PORT 00=51
0023	00			{цвет символа в порт монитора}
0024	3E		MVI A, 61	61 =>A, A=61
0025	65			{номер символа в Акк, **e**->A}
0026	D3		OUT 00	A =>PORT 00,PORT 00=61
0027	00			{номер символа в порт

				монитора}
0028	3E		MVI A,51	51 => A, A=51
0029	21			{цвет символа в Акк}
002A	D3		OUT 00	A->PORT 00,PORT 00=51
002B	00			{цвет символа в порт монитора}
002C	3E		MVI A,6E	6E => A, A=6E
002D	76			{номер символа в Акк, **v**->A}
002E	D3		OUT 00	A =>PORT 00,PORT 00=6E
002F	00			{номер символа в порт монитора}
0030	3E		MVI A, 51	51 => A, A=51
0031	21			{цвет символа в Акк}
0032	D3		OUT 00	A =>PORT 00,PORT 00=51
0033	00			{цвет символа в порт монитора}
0034	3E		MVI A, 6B	6B => A, A=6B
0035	6F			{номер символа в Акк, **o**->A}
0036	D3		OUT 00	A->PORT 00,PORT 00=6B
0037	00			{номер символа в порт монитора}
0038	3E		MVI A, 51	51 => A, A=51

0039	21			{цвет символа в Акк}
003A	D3		OUT 00	A =>PORT 00,PORT 00=51
003B	00			{цвет символа в порт монитора}
003C	3E		MVI A,6B	6F => A, A=6F
003D	72			{номер символа в Акк, **r**->A}
003E	D3		OUT 00	A =>PORT 00,PORT 00=6F
003F	00			{номер символа в порт монитора}
0040	3E		MVI A, 51	51 => A, A=51
0041	21			{цвет символа в Акк}
0042	D3		OUT 00	A =>PORT 00,PORT 00=51
0043	00			{цвет символа в порт монитора}
0044	3E		MVI A,76	76 => A, A=76
0045	6B			{номер символа в Акк, **k**->A}
0046	D3		OUT 00	A =>PORT 00,PORT 00=76
0047	00			{номер символа в порт монитора}
0048	3E		MVI A, 11	11 => A, A=11
0049	21			{цвет символа в Акк}
004A	D3		OUT 00	A =>PORT 00,PORT

				00=11
004B	00			{цвет символа в порт монитора}
004C	3E		MVI A, 2C	2C =>A, A=2C
004D	6F			{номер символа в Акк, **o**->A}
004E	D3		OUT 00	A =>PORT 00,PORT 00=2C
004F	00			{номер символа в порт монитора}
0050	3E		MVI A,71	71 => A, A=71
0051	21			{цвет символа в Акк}
0052	D3		OUT 00	A->PORT 00,PORT 00=71
0053	00			{цвет символа в порт монитора}
0054	3E		MVI A,52	52 => A, A=52
0055	76			{номер символа в Акк, **v**->A}
0056	D3		OUT 00	A =>PORT 00,PORT 00=52
0057	00			{номер символа в порт монитора}
0058	3E		MVI A, 71	71 => A, A=71
0059	21			{цвет символа в Акк}
005A	D3		OUT 00	A =>PORT 00,PORT 00=71
005B	00			{цвет символа в порт

				монитора}
005C	3E		MVI A, 6F	6F => A, A=6F
005D	2C			{номер символа в Акк, **,**->A}
005E	D3		OUT 00	A->PORT 00,PORT 00=6F
005F	00			{номер символа в порт монитора}
0060	3E		MVI A, 71	71 => A, A=71
0061	71			{цвет символа в Акк}
0062	D3		OUT 00	A =>PORT 00,PORT 00=71
0063	00			{цвет символа в порт монитора}
0064	3E		MVI A,6D	6D => A, A=6D
0065	4C			{номер символа в Акк, **L**>A}
0066	D3		OUT 00	A =>PORT 00,PORT 00=6D
0067	00			{номер символа в порт монитора}
0068	3E		MVI A, 61	71 => A, A=71
0069	71			{цвет символа в Акк}
006A	D3		OUT 00	A =>PORT 00,PORT 00=71
006B	00			{цвет символа в порт монитора}
006C	3E		MVI A,61	61 => A, A=61

006D	65			{номер символа в Акк, **e**->A}
006E	D3		OUT 00	A =>PORT 00,PORT 00=61
006F	00			{номер символа в порт монитора}
0070	3E		MVI A, 61	71 => A, A=71
0071	71			{цвет символа в Акк}
0072	D3		OUT 00	A =>PORT 00,PORT 00=71
0073	00			{цвет символа в порт монитора}
0074	3E		MVI A,61	61 => A, A=61
0075	78			{номер символа в Акк, **x**->A}
0076	D3		OUT 00	A =>PORT 00,PORT 00=61
0077	00			{номер символа в порт монитора}
0078	3E		MVI A, 61	71 => A, A=71
0079	71			{цвет символа в Акк}
007A	D3		OUT 00	A =>PORT 00,PORT 00=71
007B	00			{цвет символа в порт монитора}
007C	3E		MVI A,61	61 => A, A=61
007D	61			{номер символа в Акк, **a**->A}

007E	D3		OUT 00	A =>PORT 00,PORT 00=61
007F	00			{номер символа в порт монитора}
0080	76		HLT	Остановка программы

Лабораторная работа 14

Цель: изучение возможностей МП для умножения целых чисел без знака

$16_{10}(10_{16})*20_{10}(14_{16})=320_{10}(140_{16})$

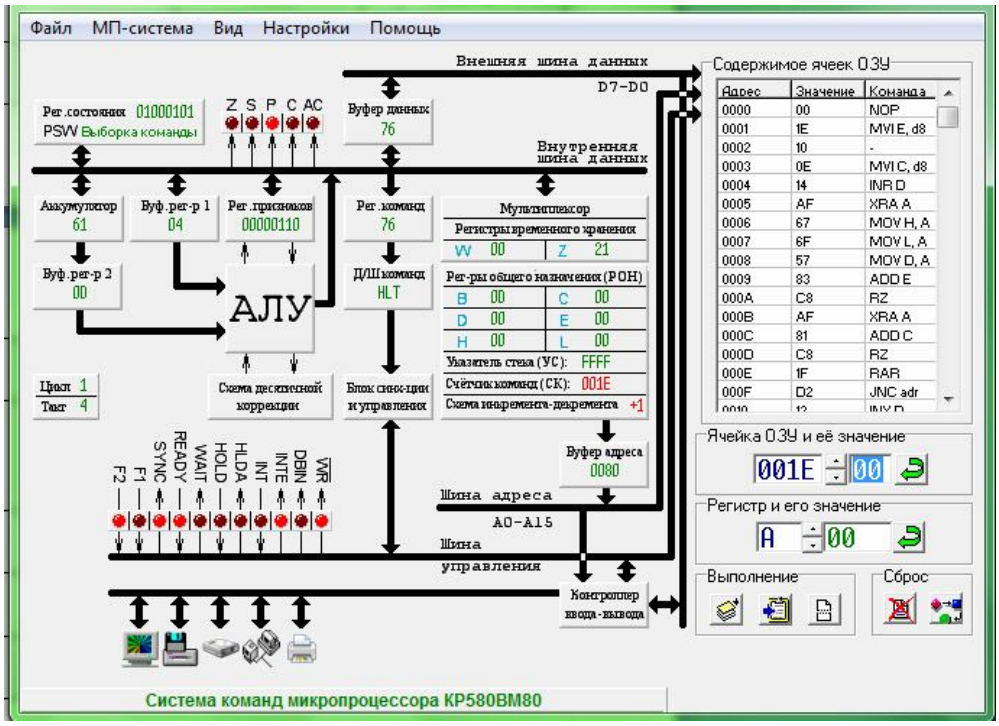


Рисунок 28 – Ввод данных

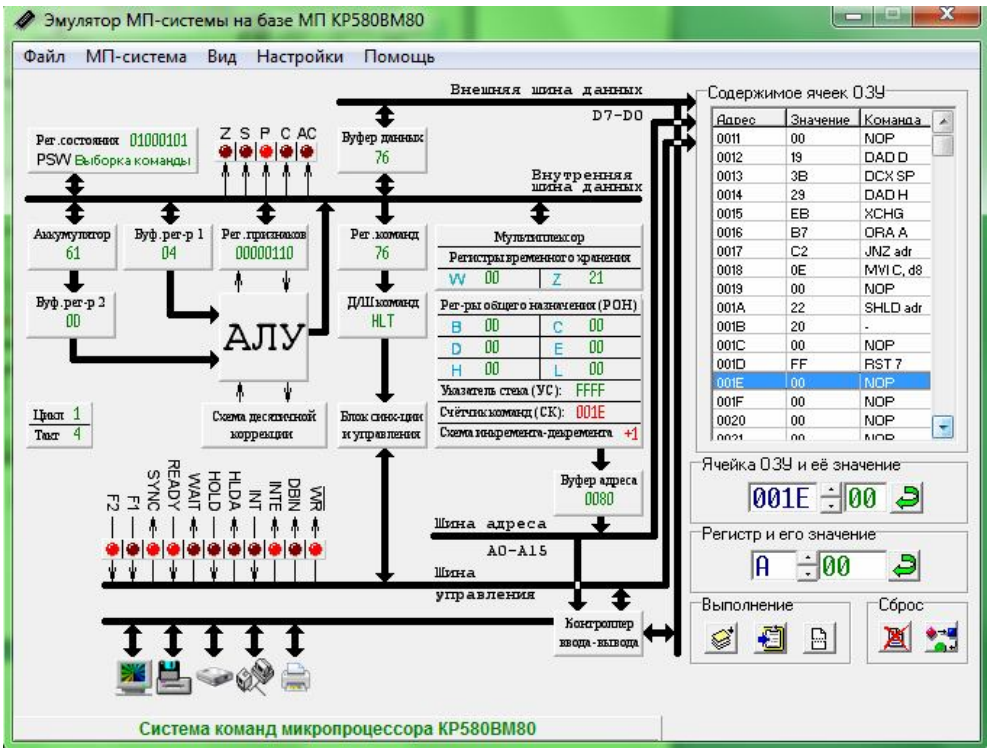


Рисунок 29 – Ввод данных

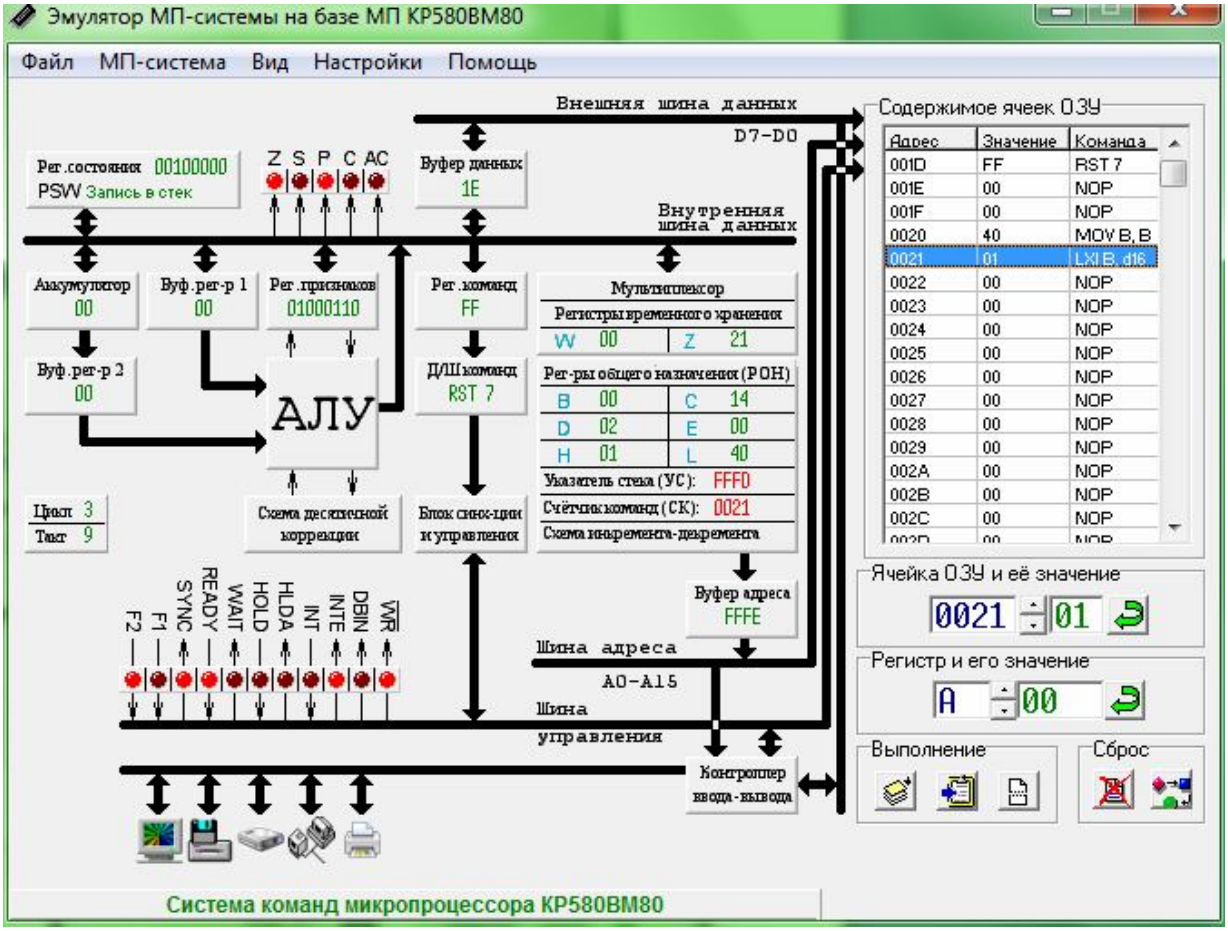


Рисунок 30 – Результат работы, записанный в ячейки 0020 и 0021

Адрес	Код команды	Метка	Мнемоника	Комментарий
0000	00			Не используется
0001	1E		MVI E,10h	Записать в регистр E число 12h (18_{10})
0002	10			
0003	0E		MVI C,1Ah	Записать в регистр C число 21h (33_{10})
0004	14			
0005	AF		XRA A	Очистить аккумулятор
0006	67		MOV H,A	Очистить регистр H
0007	6F		MOV L,A	Очистить регистр L
0008	57		MOV D,A	Очистить регистр D
0009	83		ADD E	A = A + E
000A	C8		RZ	Если A = 0, то выход
000B	AF		XRA A	Очистить аккумулятор
000C	81		ADD C	A – множитель
000D	C8		RZ	Если A = 0, то выход
000E	1F	C1:	RAR	Арифметический сдвиг аккумулятора вправо
000F	D2		JNC C2	Флаг переноса не установлен(JUMP IF NO CARRY)
0010	13			
0011	00			

0012	19		DAD D	Сдвиг множимого влево
0013	EB	C2:	XCHG	Обменять пару регистров DE с парой HL (EXCHANGE)
0014	29		DAD H	Прибавить к паре регистров HL содержимое пары регистров H
0015	EB		XCHG	Обменять пару регистров DE с парой HL (EXCHANGE)
0016	B7		ORA A	Логическое сложение «ИЛИ» аккумулятора. В данном случае очистить флаг переноса
0017	C2		JNZ C1	Флаг нуля не установлен(JUMP IF NOT ZERO)
0018	0E			
0019	00			
001A	22		RST 7	Записать результат в
001B	20			ячейку 0020
001C	00			
001D	FF			Прервать выполнение программы
…	…		…	…
0020				Результат
0021				Результат

Лабораторная работа №15

Цель: изучение алгоритмов деление двух однобайтных чисел, составление и выполнение программы деления двух однобайтных чисел без знака.

$208_{10}(D0_{16}):16_{10}(10_{16})=13_{10}(D_{16})$

0000	11	DIV B:	LXI D, D16	; Загрузка делимого и делителя в пару DE
0001	D0		RNC	Загружаемое число X
0002	10			Загружаемое число Y
0003	21		LXI H, D16	; Загрузка счетчика в регистр L и очистка регистра H
0004	08			;
0005	00		NOP	;
0006	0E		MVI C, 0	; Очистка регистра промежуточного остатка
0007	00		NOP	;
0008	7B	NXT B:	MOV A, E	; Загрузка делимого в аккумулятор
0009	17		RAL	; Смещение делимого влево на один бит
000A	5F		MOV E, A	; Пересылка делимого в регистр E
000	79		MOV A,	; Загрузка промежуточного остатка в A

B			C	
000C	17		RAL	; Смещение промежуточного остатка влево
000D	92		SUB D	; Вычитание делителя от A
000E	D2		JNC NOADD	; Если C = 0 - переход на метку NOADD
000F	12		STAX D	;
0010	00			;
0011	82		ADD D	; Восстановление остатка
0012	4F	NOADD:	MOV C, A	; Пересылка промежуточного остатка в регистр C
0013	3F		CMC	; Инверсия признака переполнения
0014	7C		MOV A, H	; Запись доли в A
0015	17		RAL	; Смещение доли
0016	67		MOV H, A	; Сохранение доли
0017	2D		DCR L	; Уменьшение регистра L
0018	C2		JNZ NXTB	; Организация цикла
0019	08			;

001A	00			;
001B	7C		MOV A, H	; Пересылка младшего байта результата в A
001C	32		STA 0050	; Сохранение содержимого A в памяти по адресу 0030
001D	30			;
001E	00			;
001F	79		MOV A, H	; Пересылка старшего байта результата в A
0020	32		STA 0051	; Сохранение содержимого A в памяти по адресу 0031
0021	31			;
0022	00			;
0023	CF		RST1	; Прекращение выполнения программы

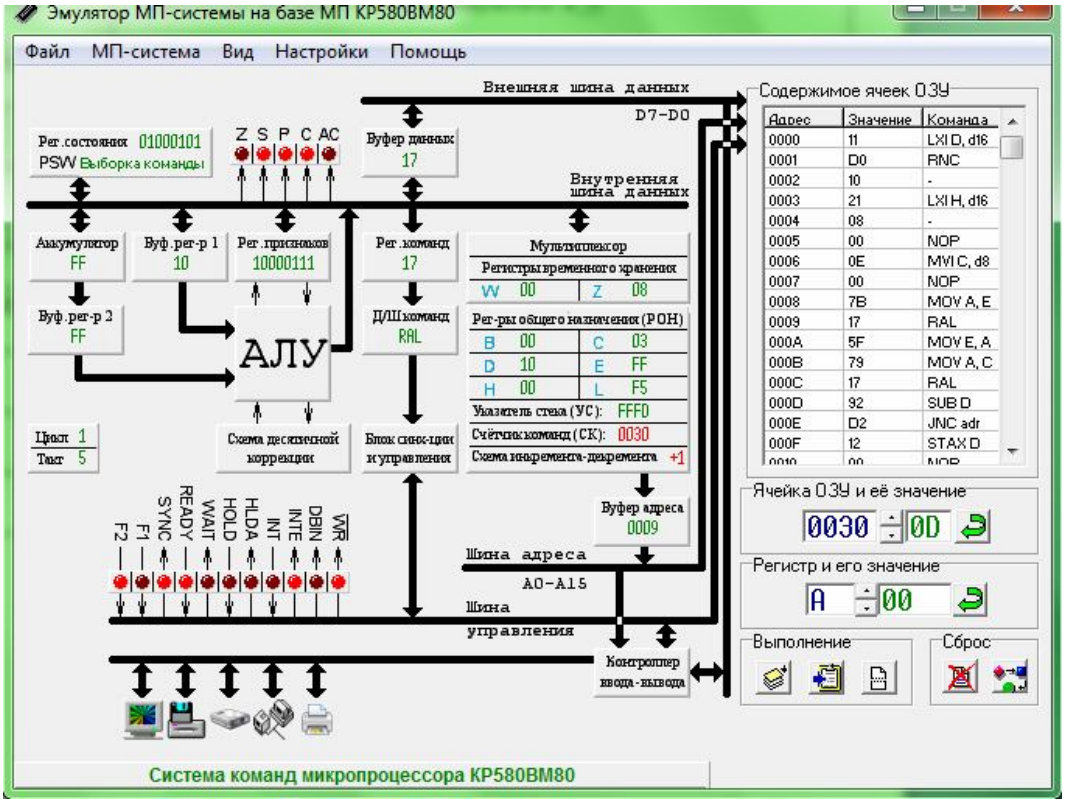


Рисунок 31 – Ввод данных

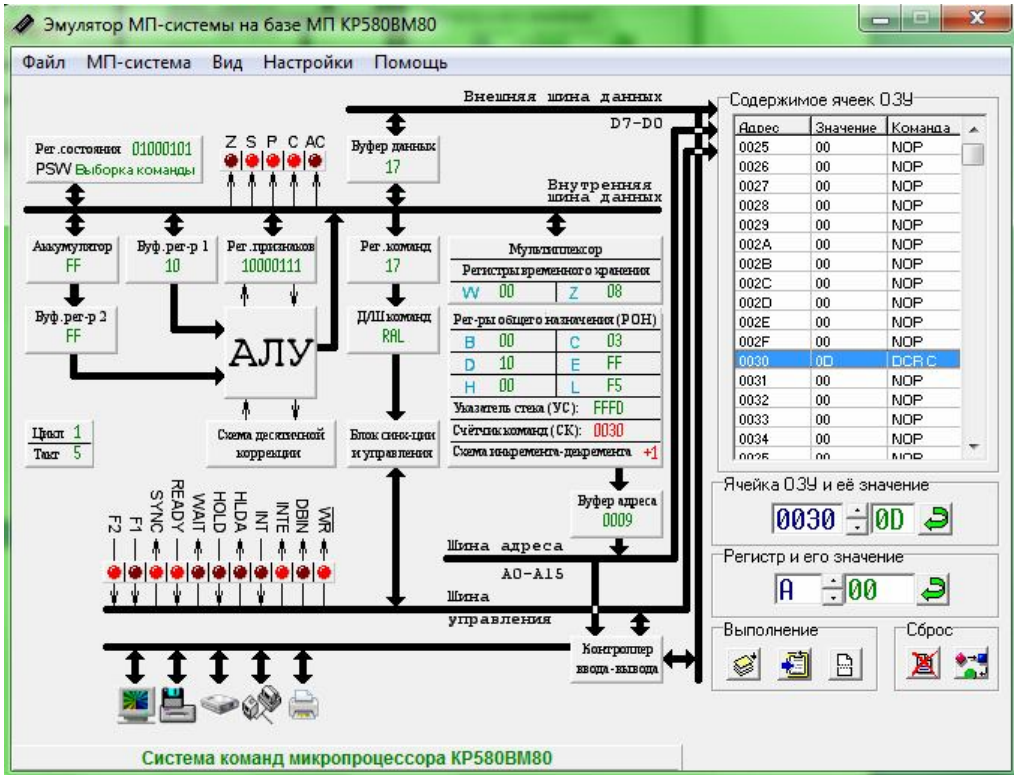


Рисунок 32 – Результат записанный в ячейку 0030