**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,**

**СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Санкт-Петербургский колледж телекоммуникаций им Э.Т. Кренкеля**

**Отчёт о выполнении лабораторной работы №3**

**по дисциплине "Системное программирование"**

**Тема:** Низкоуровневое программирование.

Работа с программами и данными в машинном представлении.

Принял преподаватель: Кривоносова Н.В

Выполнил: студент группы ЗФ-054

Фомина Е.А.

Санкт-Петербург

2022 год

**Низкоуровневое программирование.**

**Работа с программами и данными в машинном представлении.**

1. **Цель работы**

В соответствии с рабочей программой по дисциплине «Системное программирование»

в результате выполнения заданий по лабораторным работам студент должен:

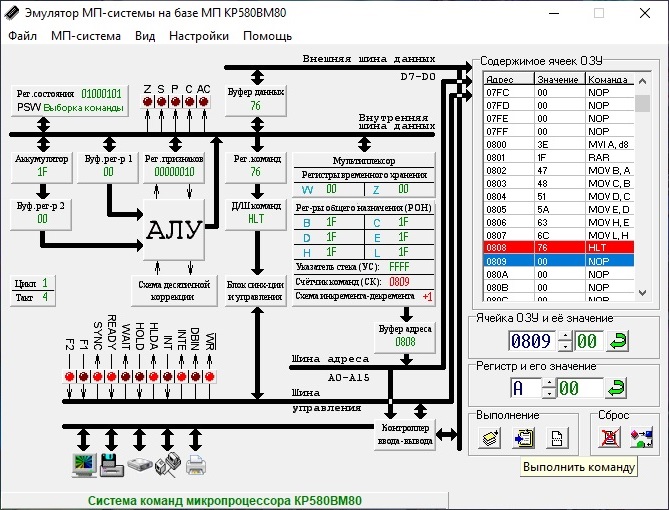
уметь:

* освоить способы записи программ в машинных кодах, перевода ассемблерной программы в машинный код;
* приобрести навыки ввода, отладки и выполнения программ;
* закрепить теоретические знания об архитектуре и ресурсах микропроцессоров.

1. **Задания**
   1. **Исследование команд обмена данными регистров и ячеек памяти.**

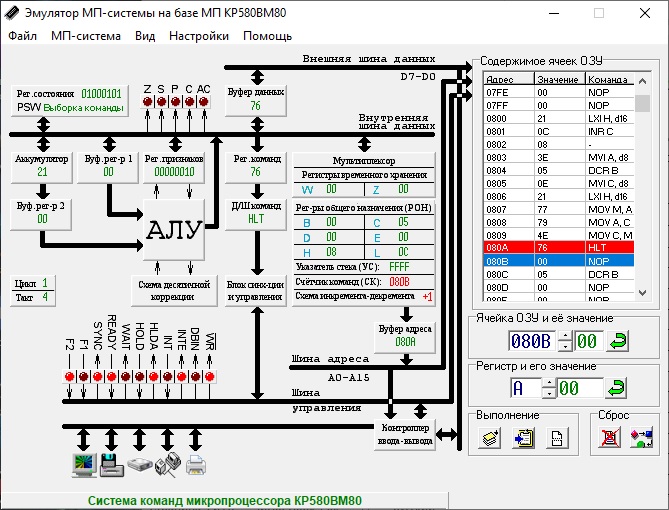
* Составить программу последовательной записи числа 1F16 в регистры A, B, C, D, E, H, L, пользуясь командами MOV r1, r2 и MVI r, D8. Программу выполнить в режиме трассировки, фиксируя в отчете содержимое регистров до и после выполнения каждой команды.

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Мнемоника | Содержимое регистров |
|  |  |  | A=00 ; B=00 ; C=00 ; D=00 ; E=00 ; H=00 ; L=00 ; |
| 0800 | 3E | MVI A, d8 |  |
| 0801 | 1F | RAR |  |
|  |  |  | **A=1F ;** B=00 ; C=00 ; D=00 ; E=00 ; H=00 ; L=00 ; |
| 0802 | 47 | MOV B, A |  |
|  |  |  | A=1F ; **B=1F ;** C=00 ; D=00 ; E=00 ; H=00 ; L=00 ; |
| 0803 | 48 | MOV C, B |  |
|  |  |  | A=1F ; B=1F ; **C=1F ;** D=00 ; E=00 ; H=00 ; L=00 ; |
| 0804 | 51 | MOV D, C |  |
|  |  |  | A=1F ; B=1F ; C=1F ; **D=1F ;** E=00 ; H=00 ; L=00 ; |
| 0805 | 5A | MOV E, D |  |
|  |  |  | A=1F ; B=1F ; C=1F ; D=1F ; **E=1F ;** H=00 ; L=00 ; |
| 0806 | 63 | MOV H, E |  |
|  |  |  | A=1F ; B=1F ; C=1F ; D=1F ; E=1F ; **H=1F ;** L=00 ; |
| 0807 | 6C | MOV L, H |  |
|  |  |  | A=1F ; B=1F ; C=1F ; D=1F ; E=1F ; H=1F ; **L=1F ;** |
| 0808 | 76 | HLT |  |
| 0809 | 00 | NOP |  |

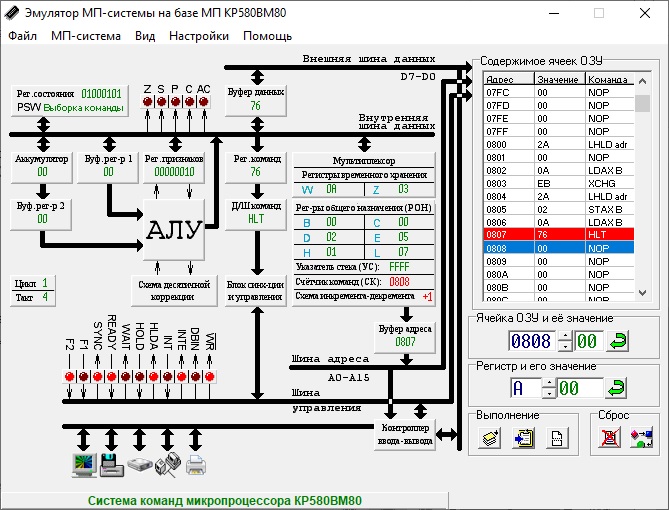


* Составить программу обмена данными между регистрами A и C с применением регистра М. Программу выполнить в режиме трассировки, фиксируя в отчете содержимое регистров до и после выполнения каждой команды.

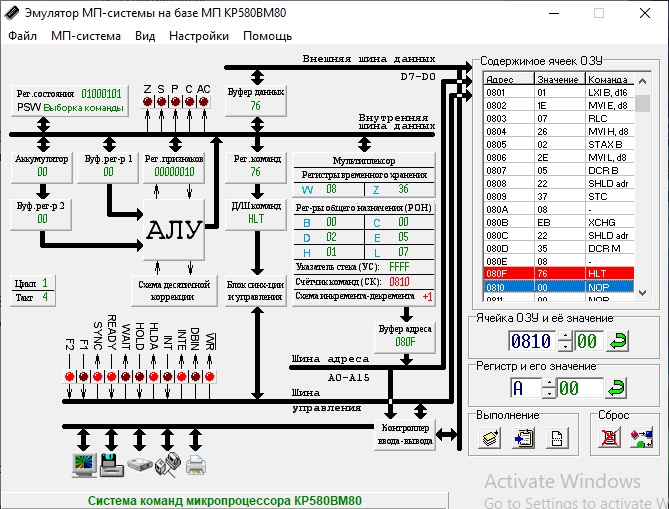
|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Мнемоника | Коментарий |
| 0800 | 21 | LXI H, d16 | A=00; C=00; H=00; L=00; |
| 0801 | 0C | INR C |  |
| 0802 | 08 | - |  |
| 0803 | 3E | MVI A, d8 | A=00; C=00; **H=08; L=0С;** |
| 0804 | 05 | DCR B |  |
| 0805 | 0E | MVI C, d8 | **A=05;** C=00; H=08; L=0С; |
| 0806 | 21 | LXI H, d16 |  |
| 0807 | 77 | MOV M, A | A=05; **C=21;** H=08; L=0С; (M=05) |
| 0808 | 79 | MOV A, C |  |
| 0809 | 4E | MOV C, M | **A=21;** C=21; H=08; L=0С; |
| 080A | 76 | HLT | A=21; **C=05;** H=08; L=0С; |
| 080B | 00 | NOP |  |
| 080C | 05 | DCR B |  |



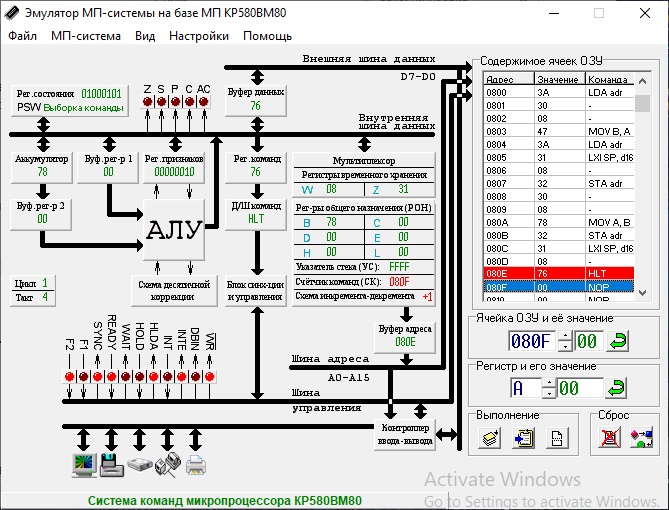
* Записать последовательно элементы массива в регистры E, D, L, H. Исходный массив: 0А00 (0516); 0А01 (0216); 0А02 (0716); 0А03 (0116). Предварительно записать исходный массив в память.



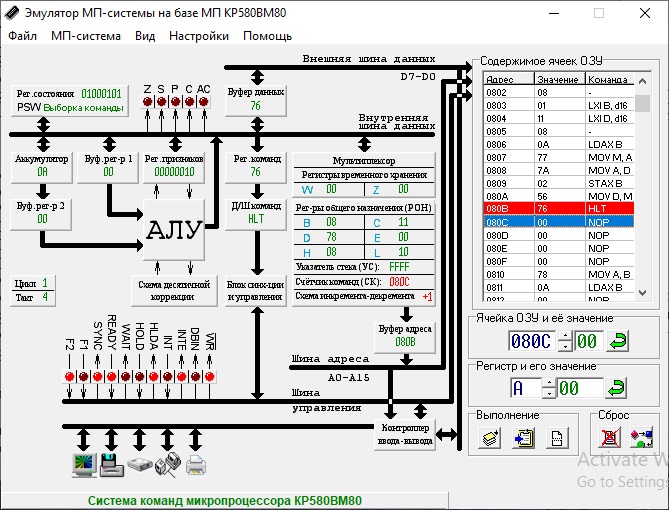
* В регистрах записаны числа: D – 0116, E – 0716, H –0216, L – 0516. Образовать массив {07, 01, 05, 02} в памяти, начиная с ячейки 0835. Предварительно записать числа в регистры.



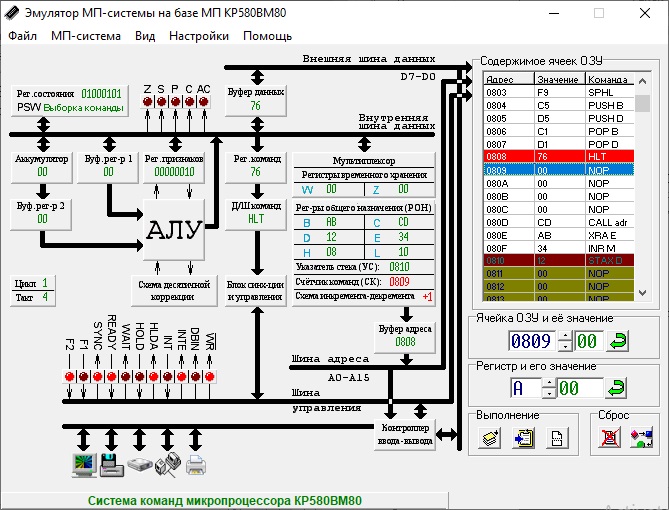
* Составить программу обмена данными между двумя ячейками памяти 0830 (7816) и 0831 (Е316). Предварительно записать числа в память.



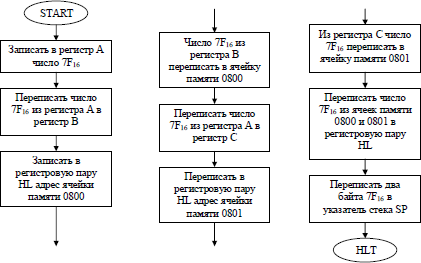
* Составить программу обмена данными между ячейками памяти 0820 (7816) и регистром D (0А16). Предварительно записать данные в ячейку памяти и регистр. Для промежуточного хранения данных использовать регистр М.

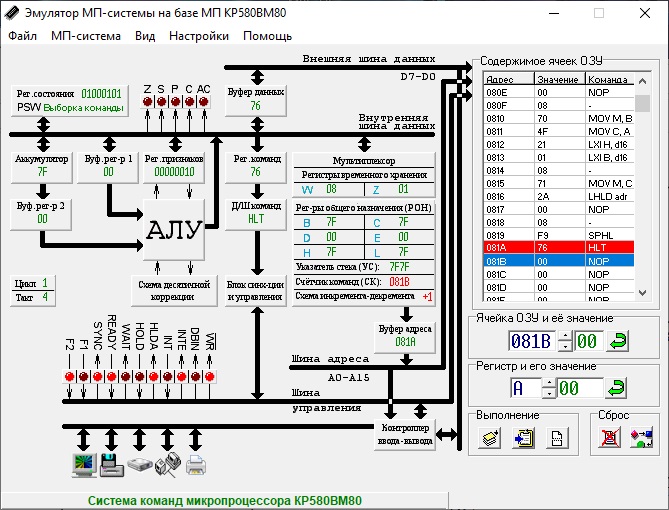


* 1. **Исследование команд обмена данными со стеком.**
* Организовать стековую область памяти, начиная с ячейки 0820. Поменять местами содержимое регистровых пар В-С и D-E, используя команды обмена данными со стеком. Предварительно заполнить регистры данными: B – 1216, C – 3416, D –AB16, E – CD16.

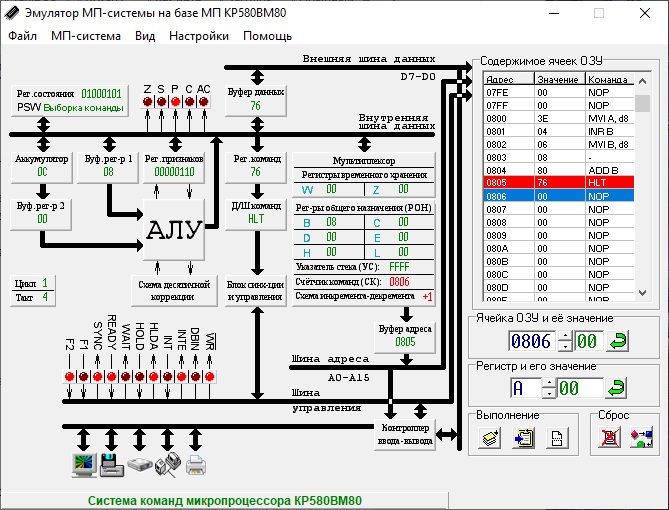


* 1. **Составление программ управления.**
* Составить программу в соответствии с предложенной блок-схемой проверить корректность ее работы.

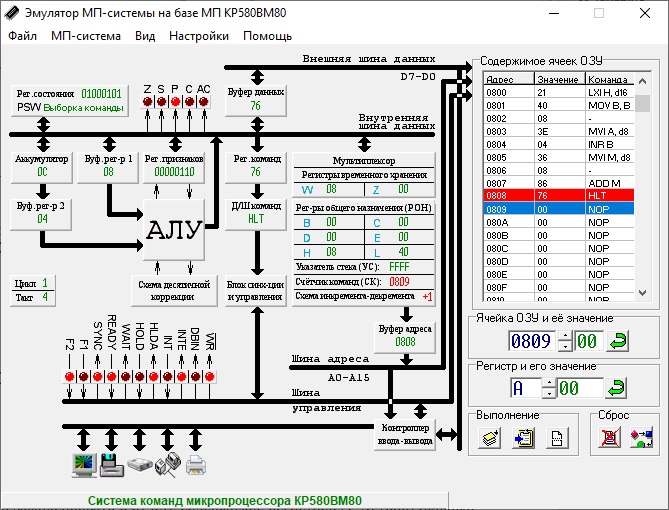


****

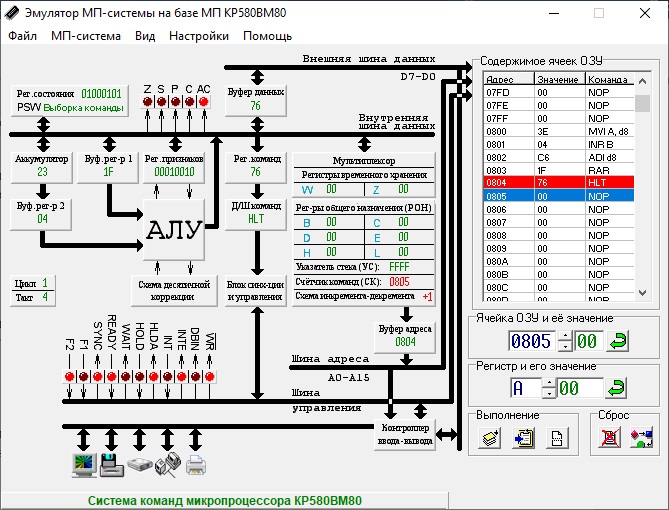
* 1. **Изучение арифметических операций.**
* Выполнить операцию сложения двух однобайтных чисел с применением команд: ADD r, ADD M, ADI D8.
* Просмотреть и зафиксировать в отчете содержимое регистров А (аккумулятора), В, F (регистра состояний).



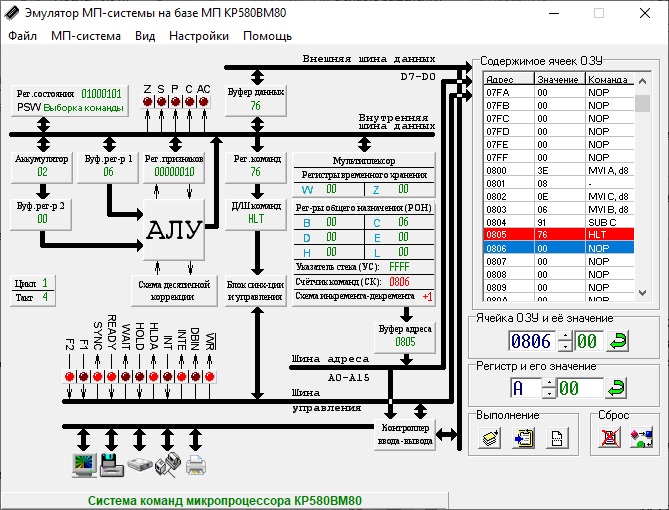
* Просмотреть и зафиксировать в отчете содержимое регистров А, М (виртуальный регистр), F, регистровой пары HL.



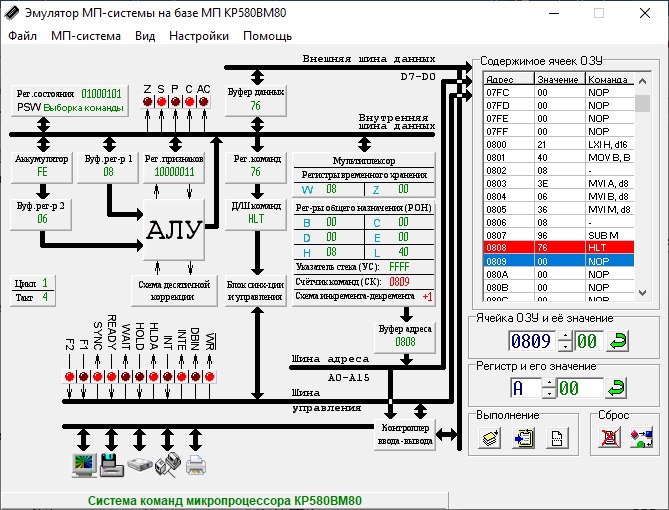
* Просмотреть и зафиксировать в отчете содержимое регистров А, F.



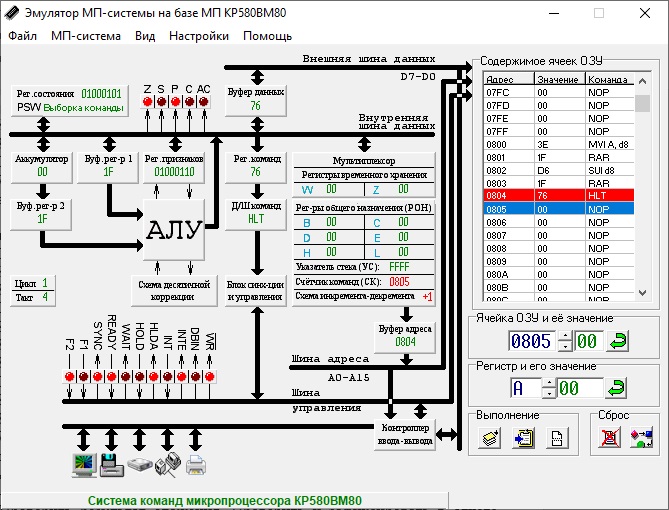
* Выполнить операцию вычитания двух однобайтных чисел с применением команд: SUB r, SUB M, SUI D8.
* Просмотреть и зафиксировать в отчете содержимое регистров A, C, F.



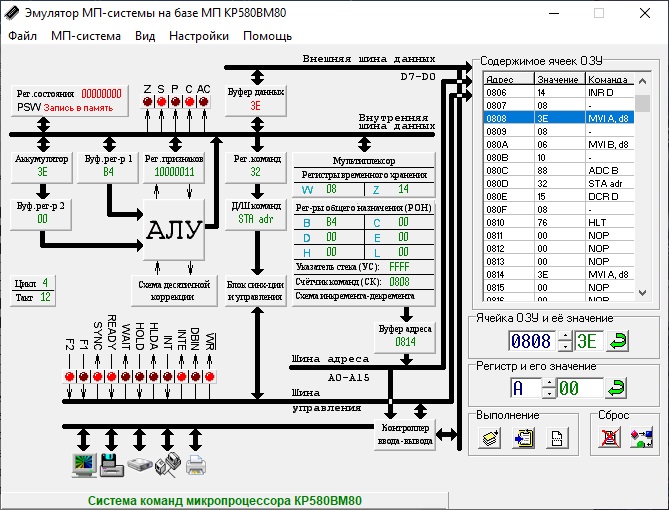
* Просмотреть и зафиксировать в отчете содержимое A, M, F регистровой пары HL.

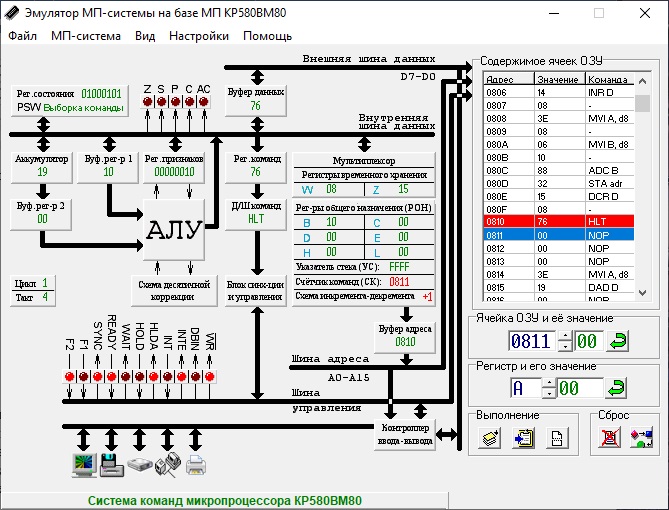


* Просмотреть и зафиксировать в отчете содержимое регистров A, F.

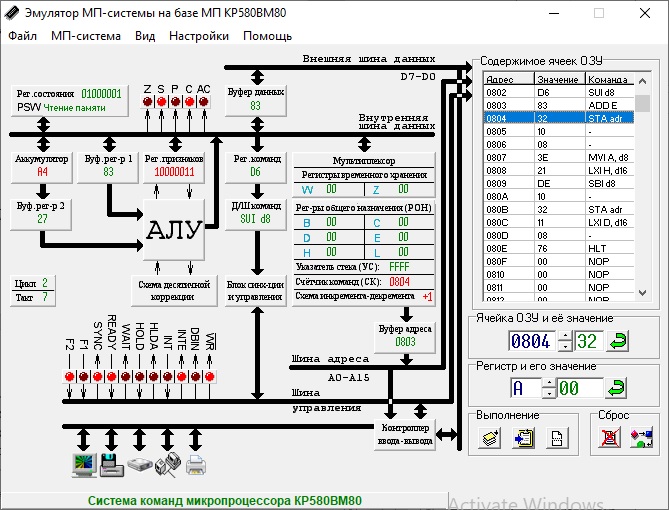


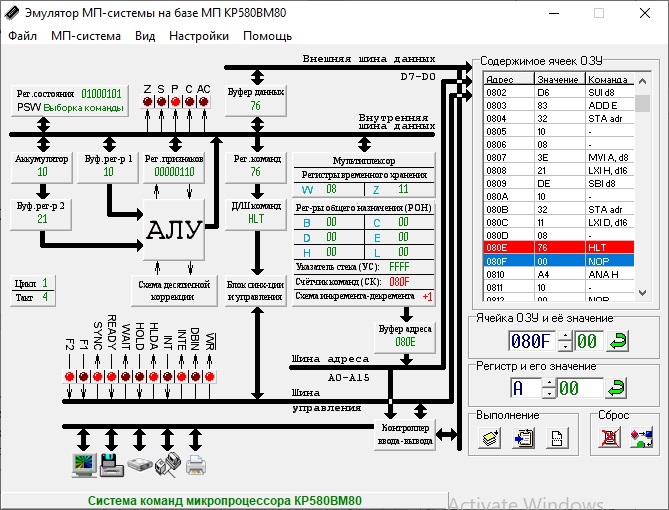
* Выполнить операцию сложения двух двухбайтных чисел с применением одной из команд: ADС r, ADС M, AСI D8.
* Проверить и зафиксировать в отчете содержимое регистра F после сложения младших и старших байтов, а также результат сложения.



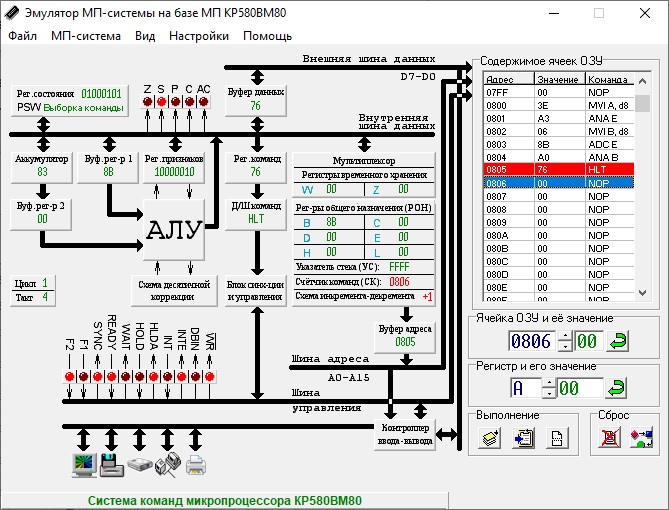


* Выполнить операцию вычитания двух двухбайтных чисел с применением одной из команд: SВB r, SВB M, SВI D8.
* Проверить и зафиксировать в отчете содержимое регистра F после вычитания младших и старших байтов, а также результат вычитания чисел – ячейки памяти 0850 (младший байт разности) и 0851 (старший байт разности).



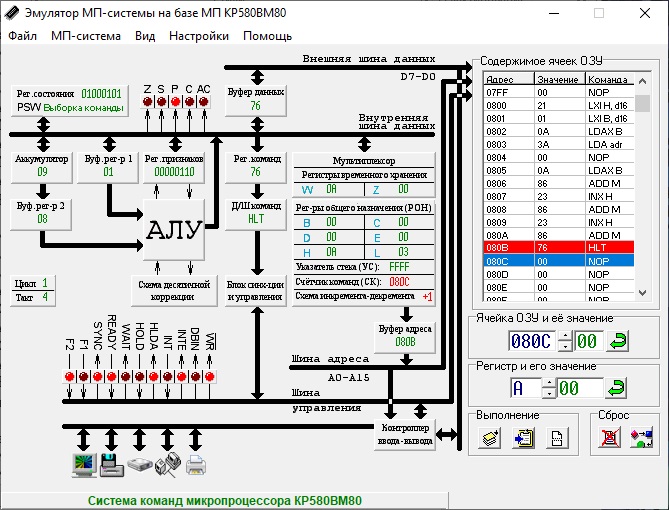


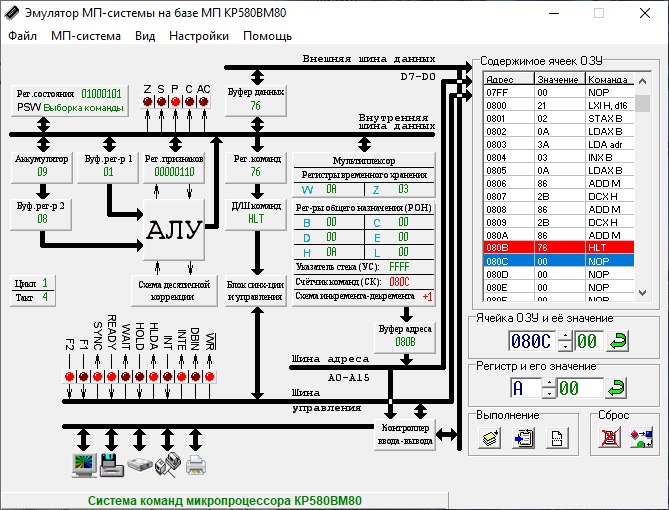
* 1. **Изучение логических операций.**
* Выполнить операцию логического умножения чисел101000112 (A316) и 100010112 (8B16) с помощью любой команды: ANA r, ANA M, ANA D8. Проверить правильность результата и зафиксировать в отчете содержимое регистров А, F.



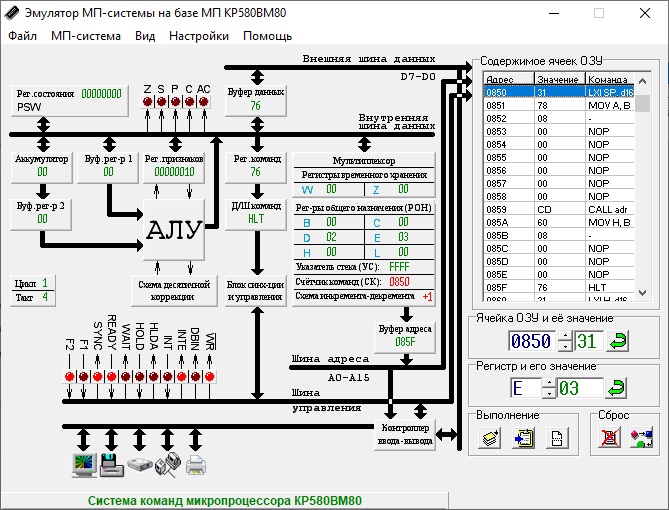
101000112 & 100010112 = 100000112 = 8316

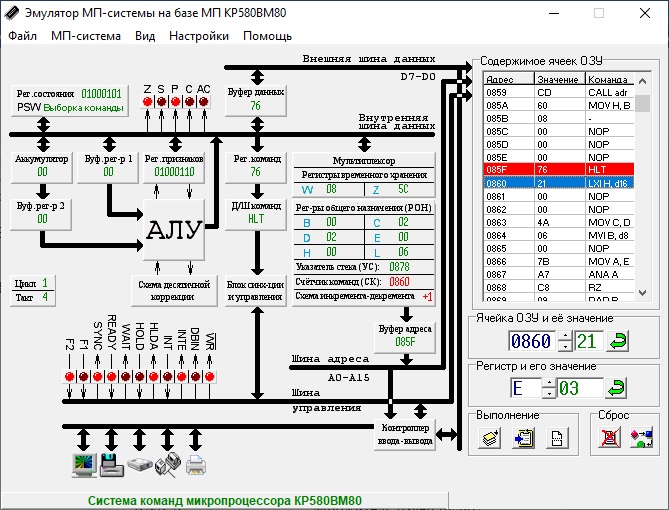
* Задан массив чисел: 0А00←0116; 0А01←0516; 0А02←0216; 0А03←0116. Пользуясь командами INX r и DCX r, определить сумму элементов массива. Перед выполнением программы необходимо ввести массив в память.



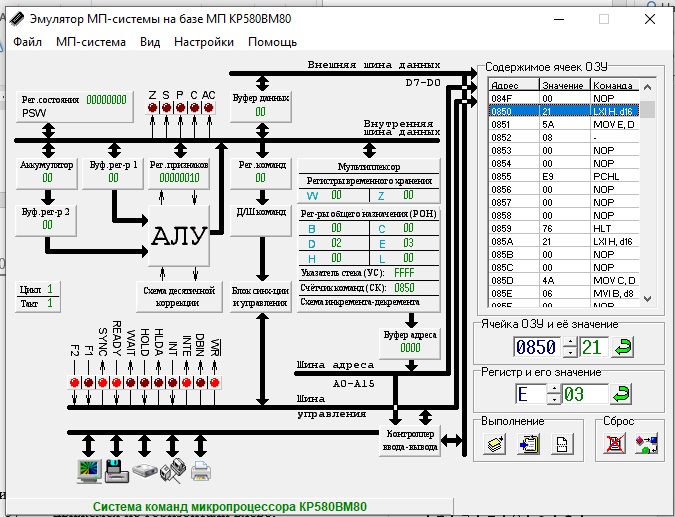


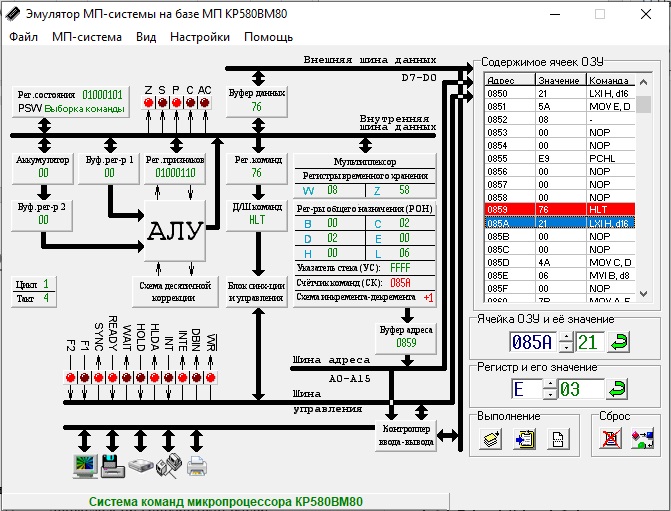
* 1. **Изучение команд переходов.**
* Для исследования команд JMP A16, CALL A16, RZ, NOP и HLT записать и пошагово выполнить подпрограмму, которая представляет собой подпрограмму умножения двух 8- разрядных двоичных чисел без знака.



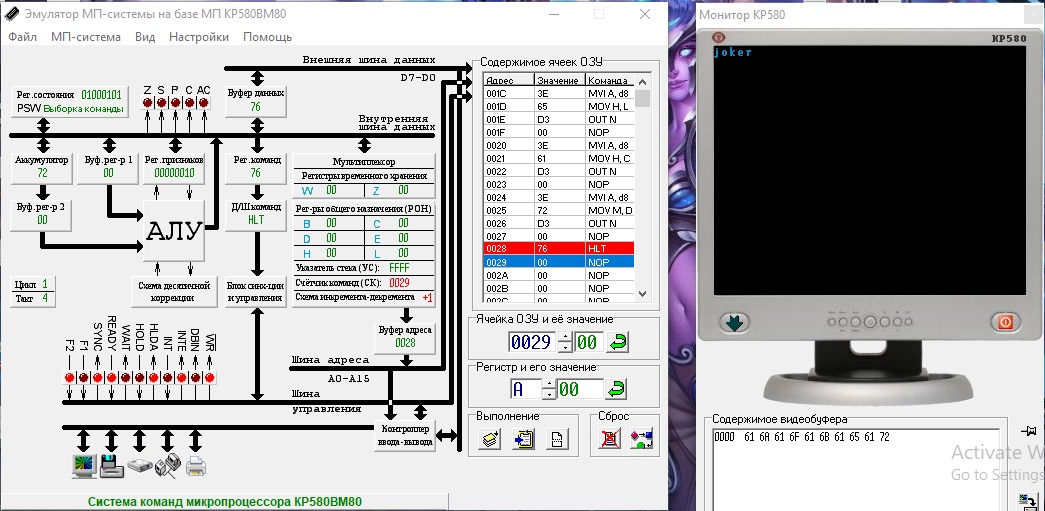


* Исследовать команды управления PCHL и JZ. Для этого предыдущую программу умножения изменить с использованием данных команд и выполнить ее в режиме трассировки



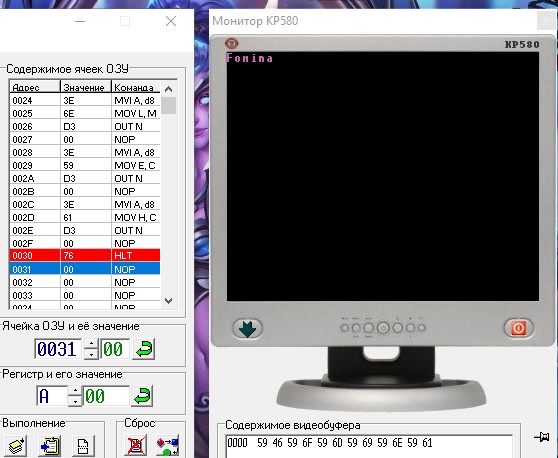


* 1. **Работа с прерываниями.**
* Написать программу для вывода на экран монитора КР580 слово joker. Цвет букв голубой.

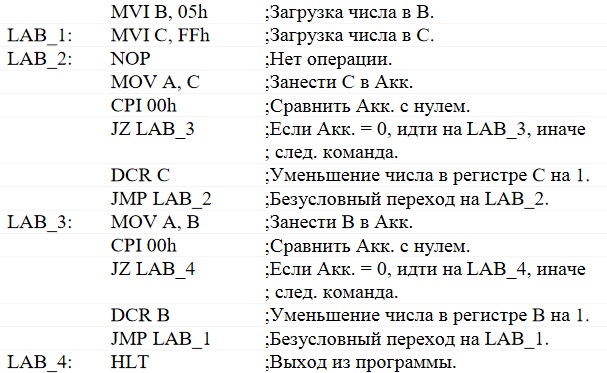


* Вывести на экран монитора КР580 свою фамилию. Записать программу в отчет вместе с кодами команд и комментариями.

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Мнемоника | Коментарий |
| 0000 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0001 | 59 | MOV E, C | {цвет символа в Акк} |
| 0002 | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 0003 | 00 | NOP | {цвет символа в порт монитора} |
| 0004 | 3E | MVI A, d8 | 46 => A, A=46 |
| 0005 | 46 | SUB H | {номер символа в Акк, **F**->A} |
| 0006 | D3 | OUT N | A =>PORT 00,PORT 00=46 |
| 0007 | 00 | NOP | {номер символа в порт монитора} |
| 0008 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0009 | 59 | MOV E, C | {цвет символа в Акк} |
| 000A | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 000B | 00 | NOP | {цвет символа в порт монитора} |
| 000C | 3E | MVI A, d8 | 6F => A, A=6F |
| 000D | 6F | ADC M | {номер символа в Акк, **o**->A} |
| 000E | D3 | OUT N | A =>PORT 00,PORT 00=6F |
| 000F | 00 | NOP | {номер символа в порт монитора} |
| 0010 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0011 | 59 | MOV E, C | {цвет символа в Акк} |
| 0012 | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 0013 | 00 | NOP | {цвет символа в порт монитора} |
| 0014 | 3E | MVI A, d8 | 6D => A, A=6D |
| 0015 | 6D | ADC H | {номер символа в Акк, **m**->A} |
| 0016 | D3 | OUT N | A =>PORT 00,PORT 00=6D |
| 0017 | 00 | NOP | {номер символа в порт монитора} |
| 0018 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0019 | 59 | MOV E, C | {цвет символа в Акк} |
| 001A | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 001B | 00 | NOP | {цвет символа в порт монитора} |
| 001C | 3E | MVI A, d8 | 69 => A, A=69 |
| 001D | 69 | ADC B | {номер символа в Акк, **i**->A} |
| 001E | D3 | OUT N | A =>PORT 00,PORT 00=69 |
| 001F | 00 | NOP | {номер символа в порт монитора} |
| 0020 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0021 | 59 | MOV E, C | {цвет символа в Акк} |
| 0022 | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 0023 | 00 | NOP | {цвет символа в порт монитора} |
| 0024 | 3E | MVI A, d8 | 6E => A, A=6E |
| 0025 | 6E | ADC L | {номер символа в Акк, **n**->A} |
| 0026 | D3 | OUT N | A =>PORT 00,PORT 00=6E |
| 0027 | 00 | NOP | {номер символа в порт монитора} |
| 0028 | 3E | MVI A, d8 | 61 => A, A=59 |
| 0029 | 59 | MOV E, C | {цвет символа в Акк} |
| 002A | D3 | OUT N | A->PORT 00,PORT 00=59 |
| 002B | 00 | NOP | {цвет символа в порт монитора} |
| 002C | 3E | MVI A, d8 | 61 => A, A=61 |
| 002D | 61 | ADD B | {номер символа в Акк, **a**->A} |
| 002E | D3 | OUT N | A =>PORT 00,PORT 00=61 |
| 002F | 00 | NOP | {номер символа в порт монитора} |
| 0030 | 76 | HLT | остановка |



* 1. **Организация временной задержки.**

****

* Записать коды команд программы временной задержки.

|  |  |  |
| --- | --- | --- |
| № Ячейки | Значение | Мнемоника |
| 0400 | 06 05 | MVI B, d8 |
| 0402 | 0E FF | MVI C, d8 |
| 0404 | 00 | NOP |
| 0405 | 79 | MOV A, C |
| 0406 | FE 00 | CPI d8 |
| 0408 | CA OF 04 | JZ adr |
| 040B | 0D | DCR C |
| 040C | C3 05 04 | JMP adr |
| 040F | 78 | MOV A, B |
| 0410 | FE 00 | CPI d8 |
| 0412 | CA 19 04 | JZ adr |
| 0415 | 05 | DCR B |
| 0416 | C3 02 04 | JMP adr |
| 0419 | 76 | HLT |

* Подбором содержимого регистров B и C в первых двух строках программы добиться временной задержки, приблизительно равной 2 – 3 минутам.
  + При C = FF и B = 6 задержка будет чуть более двух минут
* Определить, при каких числах в регистрах B и C программа будет осуществлять минимальное и максимальное время задержки, приблизительно оценить минимальное и максимальное время задержки.
  + Минимальное время будет при B и C равных 0(00)
  + Максимальное при 255(FF)
* Что нужно изменить в программе, чтобы время задержки увеличить до одного часа?
  + Подобрать значения B и C при которых задержка будет равна 1 минуте. Добавить еще один цикл с регистром D повторяющий полученную программу 60 раз.

1. **Контрольные вопросы.**
   1. **Перечень, назначение и процесс выполнения команд пересылки данных.**

MOV R1,R2 - Пересылка данных из регистра в регистр (R1)←(R2)

MOV R,M - Пересылка данных из памяти в регистр (R)←((H,L))

MOV M,R - Пересылка данных из регистра в память ((H,L))← (R)

MVI R,B2 - Пересылка непосредственных данных в регистр (R)←В2

MVI М,B2 - Пересылка непосредственных данных в память ((H,L))←В2

LDA – Прямая загрузка А-регистра

LDAX - Косвенная загрузка А-регистра

STA - Прямое запоминание содержимого А-регистра

SHLDY - Прямое запоминание содержимого Н-пары регистров

XCHG - 0бмен между Н- и D-парами регистров

* 1. **Понятие стековой области. Перечень, назначение и процесс выполнения команд обмена данными со стеком.**

Стековая память реализует безадрес­ное задание операндов. Стек представляет собой груп­пу последовательно, пронумерованных регистров или ячеек памяти, снабженных указателем стека, в котором автоматически при записи и считывания устанавливается номер (адрес) последней занятой ячейки стека (вершины стека).

PUSH - Запись в стек

POP - Чтение из стека

XTHL - Обмен между вершиной стека и Н-парой регистров

SPHL - Запись содержимого Н пары регистров в указатель стека

IN - Ввод данных

OUT - Вывод данных

* 1. **Перечислите команды сложения с указанием их назначения.**

ADD R - Сложение содержимого А-регистра

ADI B2 - Сложение непосредственных данных и содержимого А-регистра

ADC R - Сложение содержимых регистра и А-регистра с переносом

ADC M - Сложение содержимых ячейки памяти и А-регистра с переносом

ACI B2 - Сложение непосредственных данных н А-регистра с переносом

* 1. **Перечислите команды вычитания с указанием их назначения.**

SUB R - Вычитание содержимого регистра из содержимого А-регистра

SUB M - Вычитание содержимого ячейки памяти из содержимого А-регистра

SUI B2 - Вычитание непосредственных данных из содержимого А-регистра

SBB R - Вычитание содержимого регистра из содержимого А-регистра с заемом

SBB M - Вычитание ячейки памяти из содержимого А-регистра с заемом

SBI B2 - Вычитание непосредственных данных из содержимого А-регистра с заемом

* 1. **В чем отличие выполнения операция с двухбайтными числами от выполнения операций с однобайтными числами?**

Операции с однобайтными выполняются в одно действие, а с двухбайтными в несколько. При сложении сначала складываются младшие байты чисел, затем старшие с учетом переноса в старший байт. При вычитании сначала из младшего байта уменьшаемого вычитается младший байт вычитаемого, далее с учетом заема в младший байт из старшего байта уменьшаемого вычитается старший байт вычитаемого

* 1. **Перечислите команды логических операций с указанием их назначения.**

ANA, ANI - Поразрядное И

CMP, CPI – Сравнение

XRA, XRI - Поразрядное исключающее ИЛИ

ORA, ORI - Поразрядное ИЛИ

* 1. **Перечислите команды сравнения с указанием их назначения.**

CMP R - Сравнение содержимых регистра и А-регистра

CMP M - Сравнение содержимых ячейки памяти и А-регистра

CPI B2 - Сравнение непосредственных данных с содержимым А-регистра

* 1. **Опишите назначение байта-маски.**

Позволяет выделить фрагмент двоичного кода, нужный бит или группу битов с помощью операции поразрядного логического «И» исходного кода и кода-маски

* 1. **Назначение команд управления JMP, CALL, PCHL, RET, EI, DI, HLT, NOP, Jc, Cc, Rc.**

JMP - Безусловный переход

CALL - Безусловный вызов подпрограммы

PCHL - Запись содержимого Н-нары регистров

RET - Возврат из подпрограммы

EI - Разрешение прерывания

DI - Запрет прерывания

HLT - Остановка

NOP - Пустая операция

Jc - Переход, если перенос

Cc - Вызов подпрограммы, если перенос

Rc - Возврат из подпрограммы, если перенос

* 1. **Назначение команд ввода-вывода IN port, OUT port?**

IN port - Ввод из порта с указаным номером

OUT port - Вывод в порт с указаным номером

* 1. **Назначение команд работы со стеком PUSH rr, POP rr, XTHL, SPHL.**

PUSH rr – Запись в стек содержимого регистровой пары

POP rr – Чтение содержимого регистровой пары из стека

XTHL – Обмен между вершиной стека и Н-парой регистров

SPHL - Запись содержимого Н пары регистров в указатель стека

* 1. **С помощью каких команд можно организовать цикл?**

С помощью команд передачи управления и условия из логических и арифметических операций

* 1. **Назначение каждой из команд организации цикла.**

MOV – Занести данные

CPI – Сравнить данные

JZ – Переход, если условие равно

DCR – Уменьшить данные на 1

JMP – Безусловный переход

* 1. **Какие группы регистров участвуют в организации циклов?**
  2. **Что такое подпрограмма?**

Это отдельная функционально независимая часть программы, содержащая определённый набор действий

* 1. **Для чего предназначены команды передачи управления?**

Для безусловного и условного управления переходами, обращения и выхода из подпрограмм

* 1. **Назначение стека.**

Стековая памятьиспользуется для безадрес­ного задания операндов.

* 1. **Какие в системе команд МП КР580 предусмотрены операции со стеком?**

Запись в стек, чтение из стека, обмен между вершиной стека и регистровыми парами, запись содержимого регистровой пары в указатель стека, ввод данных, вывод данных

* 1. **Какие команды МП КР580 используются при обращении к подпрограммам и возврате из подпрограмм?**

Для организации подпрограмм используются команды вызова подпрограммы CALL ADR и возврата из подпрограммы RET

* 1. **Для чего в программе временной задержки предназначены строки: 4 – 6, 9 – 11?**

Для формирования цикла