Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

ОТЧЕТ

по учебной практике

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Специальность 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Выполнил |  |
| Студент гр. 493 | \_\_\_\_\_\_\_\_\_\_\_Э.Н. Баланин |
|  |  |
| Оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Преподаватели УП | \_\_\_\_\_\_\_\_\_\_\_\_И.А. Матысик |
|  |  |

Санкт-Петербург

2023

СОДЕРЖАНИЕ

[Лабораторная работа №1. Изучение архитектуры процессора KP580 3](#_Toc128587602)

[Лабораторная работа № 2. Исследование команд прямой адресации 6](#_Toc128587603)

[Лабораторная работа № 3. Исследование команд непосредственной адресации 8](#_Toc128587604)

[Лабораторная работа № 4. Исследование команд косвенной адресации 10](#_Toc128587605)

[Лабораторная работа 5. Исследование команд стековой адресации 12](#_Toc128587606)

[Лабораторная работа 6. Пример программы для микропроцессора 14](#_Toc128587607)

[Лабораторная работа 7. Операция сложения 16](#_Toc128587608)

[Лабораторная работа №8. Операция вычитания 18](#_Toc128587609)

[Лабораторная работа № 9. Сложения элементов массива 20](#_Toc128587610)

[Лабораторная работа № 10. Сложение двухбайтовых десятичных чисел 22](#_Toc128587611)

[Лабораторная работа № 11. Вычитание двухбайтовых десятичных чисел 24](#_Toc128587612)

[Лабораторная работа № 12. Алгоритмы ветвления 26](#_Toc128587613)

[Лабораторная работа № 13. Вывод информации на экран 29](#_Toc128587614)

[Лабораторная работа № 14. Умножение целых чисел без знака 33](#_Toc128587615)

[Лабораторная работа № 15. Деление однобайтных чисел 35](#_Toc128587616)

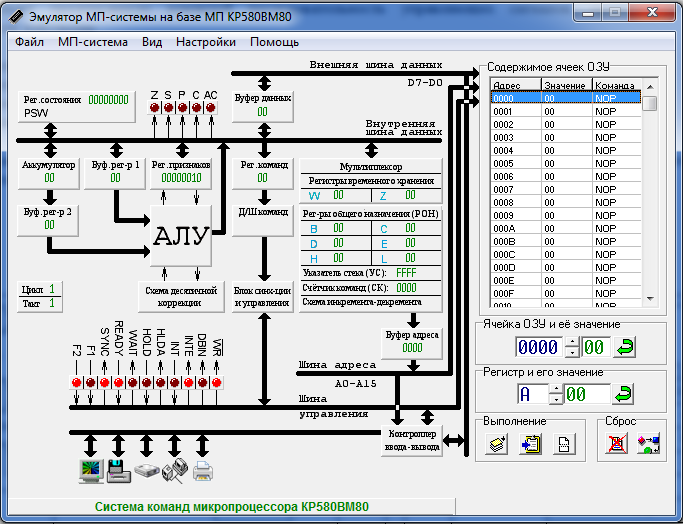
[Лабораторная работа № 16. Арифметические операции 37](#_Toc128587617)

# Лабораторная работа №1. Изучение архитектуры процессора KP580

Цель: Изучение архитектуры микропроцессора КР580.

Содержание:

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



Буфер данных

Аккумулятор

Регистр команд

Дешифратор команд

Буфер регистра 1

Регистр признаков

Буфер регистра 2

АЛУ

Схема десятичной коррекции

Блок синхронизации и управления

Содержимое ячеек ОЗУ

Ячейка ОЗУ и её значение

Регистр и её значение

Рисунок 1 – Описание компонентов программы KP580

1. Таблица регистров продемонстрирована на таблице 1.

**Таблица 1 – Регистры**

|  |  |  |
| --- | --- | --- |
| Регистр | Назначение | Разрядность |
| 8-разрядный аккумулятор А | Источник операнда и приемник результата | 8 |
| 16-разрядный регистр HL | Адресный регистр | 16 |
| PC | Счетчик команд | 16 |
| SP | Указатель стека | 16 |
| РОН | Хранения операндов и адресов ячеек памяти (4 регистра B,C,D,E) | 8 (при объединении 32) |

1. Блок-схема функционирования МП во время выполнения команды сложения содержимого аккумулятора и регистра В, имеющую мнемоническое обозначение ADD B продемонстрирована на рисунке 2.

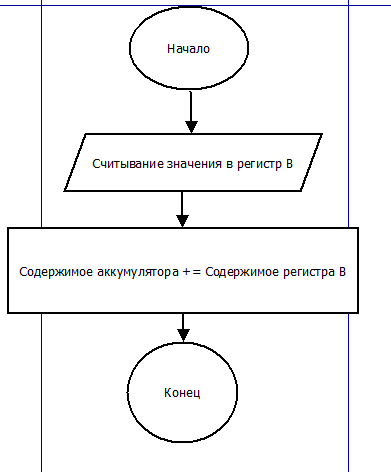


Рисунок 2 – Блок-схема команды ADD B

1. Таблица флагов продемонстрирована на таблице 2.

**Таблица 2 – Флаги**

|  |  |  |
| --- | --- | --- |
| Обозначение флага | Признак флага | Условие установки флагов |
| C (Carry) | Перенос | При переносе при сложении или заёме при вычитании из старшего разряда аккумулятора |
| M (Minus) | Отрицательный результат | Если седьмой бит аккумулятора равен 7 (знаковый бит) |
| Z (Zero) | Ноль | Результат операции в аккумуляторе равен нулю |
| P (Parity) | Четность | Четное число единиц в аккумуляторе |
| AC (Auxiliary Carry) | Половинный перенос | Перенос из третьего разряда аккумулятора в четвертый |

# Лабораторная работа № 2. Исследование команд прямой адресации

Цель: Ознакомление с командами микропроцессора КР580 для прямой адресации.

Содержание: Программный код с применением прямой адресации продемонстрирован в таблице 3.

**Таблица 3 – Программный код с применением прямой адресации**

|  |  |  |
| --- | --- | --- |
| Адрес | Значение | Команда |
| 0000 | 3A | LDA adr |
| 0001 | 0A | - |
| 0002 | 00 | NOP |
| 0003 | 32 | STA adr |
| 0004 | 0B | DCX B |
| 0005 | 00 | NOP |
| 0006 | 76 | HLT |
| 0007 | 00 | NOP |
| 0008 | 00 | NOP |
| 0009 | 00 | NOP |
| 000A | 10 | - |
| 000B | 00 | NOP |

На рисунке 3 продемонстрирован программный код программы до его исполнения.

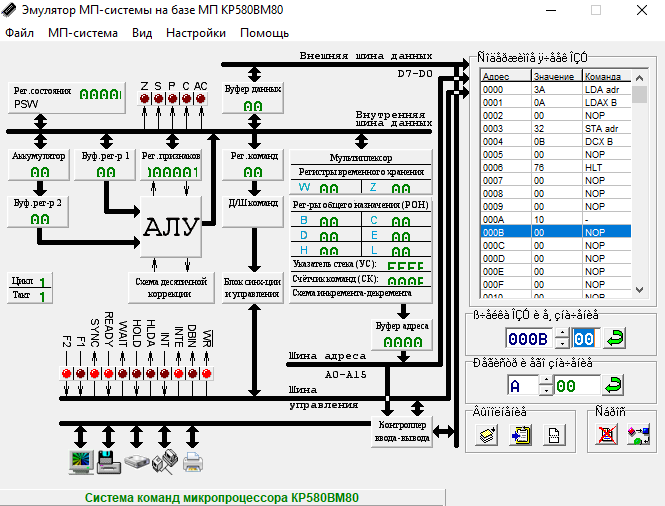


Рисунок 3 – Эмулятор перед выполнением команд

На рисунке 4 продемонстрирован результат работы программы

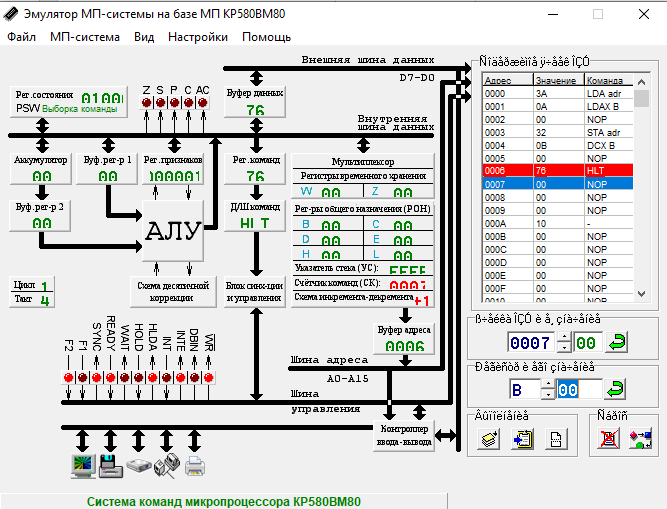


Рисунок 4 – Результат работы программы

# Лабораторная работа № 3. Исследование команд непосредственной адресации

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

Содержание: Программный код с применением команд для непосредственной адресации продемонстрирован в таблице 4.

**Таблица 4 – Программный код с применением команд для непосредственной адресации**

|  |  |  |
| --- | --- | --- |
| Адрес | Значение | Команда |
| 0000 | 3E | MVI A,20h |
| 0001 | 20 | - |
| 0002 | 06 | MVI B, 5h |
| 0003 | 5 | - |
| 0004 | 0B | MVI H, d1h |
| 0005 | D1 | - |
| 0006 | FF | RST 7 |

На рисунке 5 продемонстрирована программа до начала её выполнения.

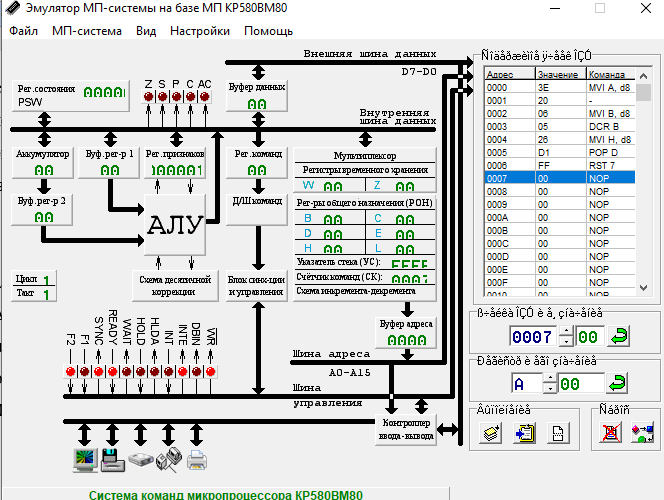


Рисунок 5 – Программа до начала выполнения

На рисунке 6 продемонстрирован результат выполнения программы.

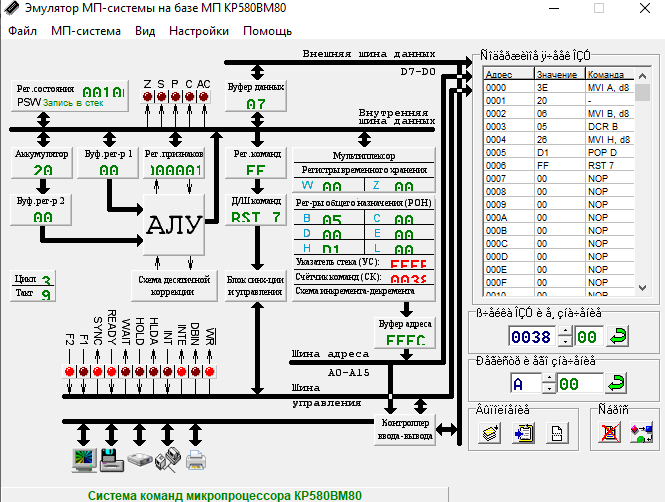


Рисунок 6 – Результат выполнения программы

# Лабораторная работа № 4. Исследование команд косвенной адресации

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

Содержание: Программный код с применением команд косвенной адресации продемонстрирован в таблице 5.

**Таблица 5 – Программный код с применением команд косвенной адресации**

|  |  |  |
| --- | --- | --- |
| Адрес | Значение | Команда |
| 0000 | 06 | MVI B, d8 |
| 0001 | 00 | NOP |
| 0002 | 0E | MVI C, d8 |
| 0003 | 0A | LDAX B |
| 0004 | 0A | LDAX B |
| 0005 | FF | RST 7 |
| 0006 | FF | RST 7 |
| 0007 | 00 | NOP |
| 0008 | 00 | NOP |
| 0009 | 00 | NOP |
| 000A | 10 | - |

На рисунке 7 продемонстрирован эмулятор процессора KP580 до начала выполнения программного кода продемонстрированного в таблице 5.

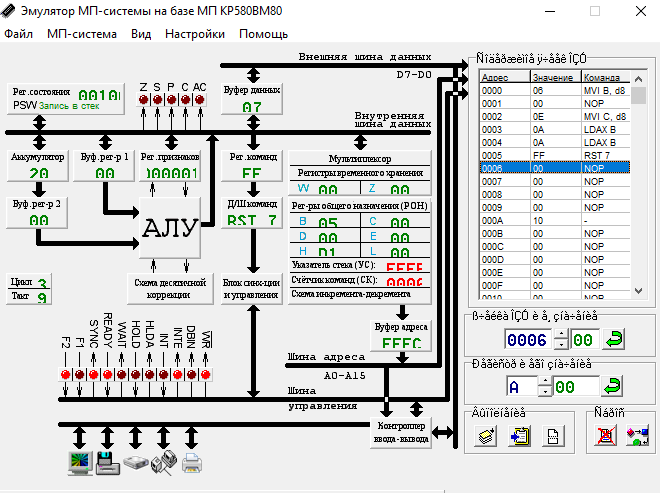


Рисунок 7 – Эмулятор перед выполнением программного кода

На рисунке 8 продемонстрирован результат выполнение программы продемонстрированной в таблице 5.

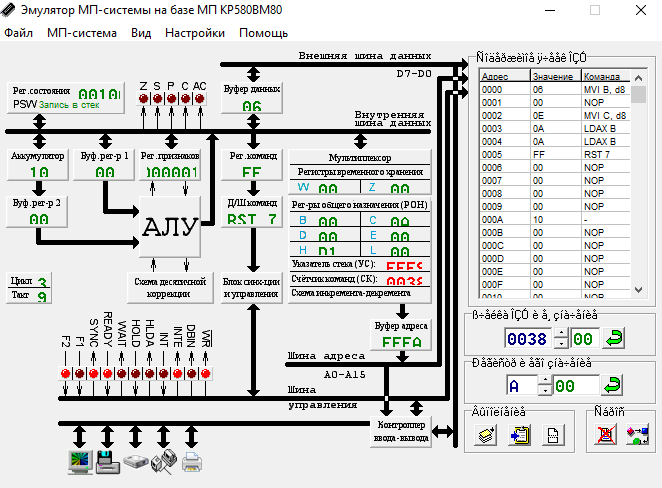


Рисунок 8 – Результат выполнения программного кода

# Лабораторная работа 5. Исследование команд стековой адресации

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

Содержание:

Программный код с применением команд стековой адресации продемонстрирован в таблице 6.

**Таблица 6 – Программный код с применением команд стековой адресации**

|  |  |  |
| --- | --- | --- |
| Адрес | Значение | Команда |
| 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 |

На рисунке 9 продемонстрирован эмулятор процессора KP580 до начала выполнения программного кода продемонстрированного в таблице 6.

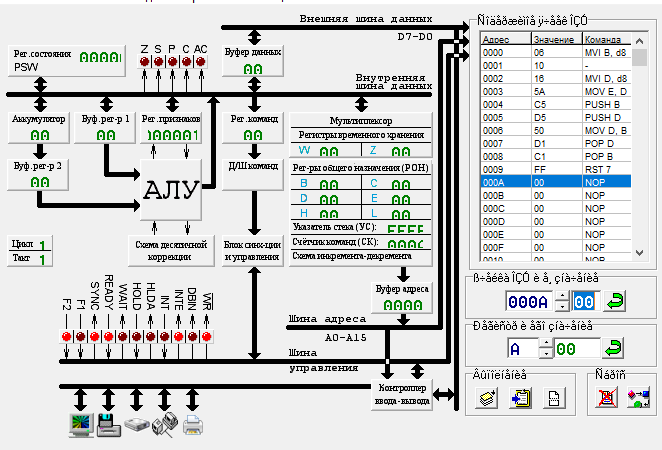


Рисунок 9 – Эмулятор перед выполнением программного кода

На рисунке 10 продемонстрирован результат выполнение программы продемонстрированной в таблице 6.

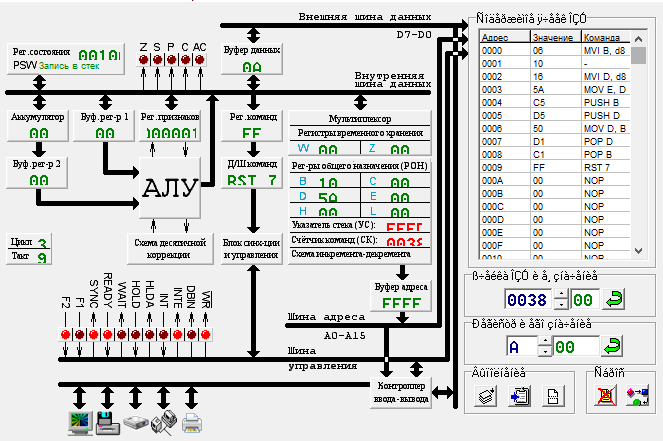


Рисунок 10 – Результат выполнения программного кода

# Лабораторная работа 6. Пример программы для микропроцессора

Цель: ознакомиться с простейшими арифметическими действиями на микропроцессоре KP580.

Содержание:

Программа приложения продемонстрирована в таблице 7

**Таблица 7 – Простейшие арифметические действия**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Метка | Мнемокод |
| 0000 | AF | - | ХRА А |
| 0001 | ЗЕ | - | MVI A, 38 |
| 0002 | 38 |
| 0003 | 06 | - | MVI B, A3 |
| 0004 | A3 |
| 0005 | 80 | - | ADD 8 |
| 0006 | E7 | - | RST 7 |

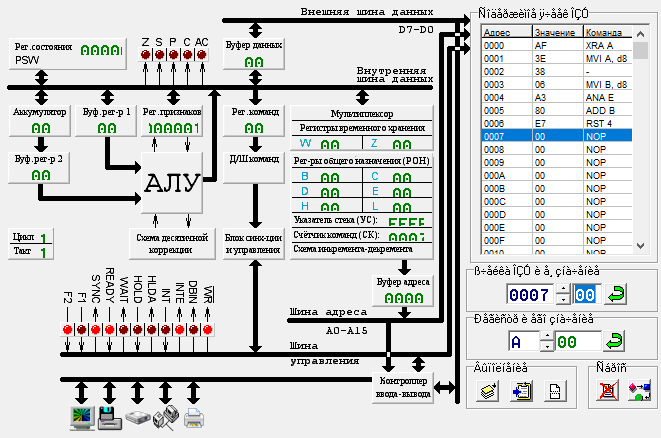
На рисунке 11 продемонстрирован эмулятор процессора KP580 до начала выполнения программного кода продемонстрированного в таблице 7.

Рисунок 11 – Эмулятор перед выполнением программного кода

На рисунке 12 продемонстрирован результат выполнение программы продемонстрированной в таблице 7.

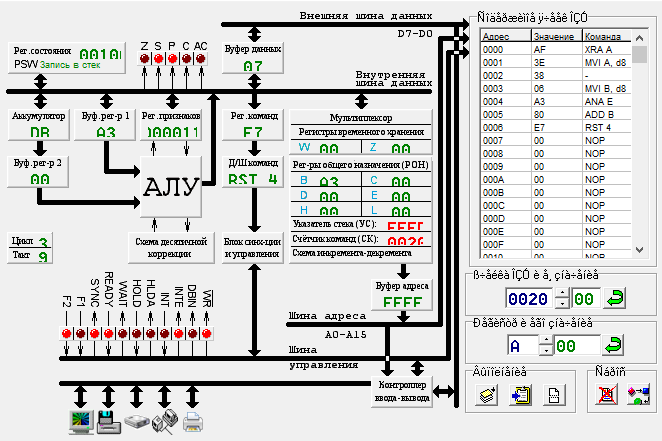


Рисунок 12 – Результат выполнения программного кода

# Лабораторная работа 7. Операция сложения

Цель: 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.

Содержание: выполнить сложение чисел: 178+23 = 201 (С916)

Программный код продемонстрирован в таблице 8.

**Таблица 8 – Программный код сложения двух чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Значение | Команда | Комментарий |
| 0000 | AF | XRA A | Очистка аккумулятора |
| 0001 | 3E | MVI A, d8 | Загрузка в регистр А значения B2 |
| 0002 | B2 | ORA D |
| 0003 | 06 | MVI B, d8 | Загрузка в регистр В значения 17 |
| 0004 | 17 | RAL |
| 0005 | 80 | ADD B | Сложение содержимого регистра В из аккумулятора |
| 0006 | 32 | STA adr | Сохранение содержимого аккумулятора в память по адресу 0020 |
| 0007 | 20 | - |
| 0008 | 00 | NOP |
| 0009 | E7 | RST 4 | Прервать выполнение программы |
| 000A | 00 | NOP |  |

На рисунке 13 продемонстрирован эмулятор до начала выполнения программного кода продемонстрированного в таблице 8.

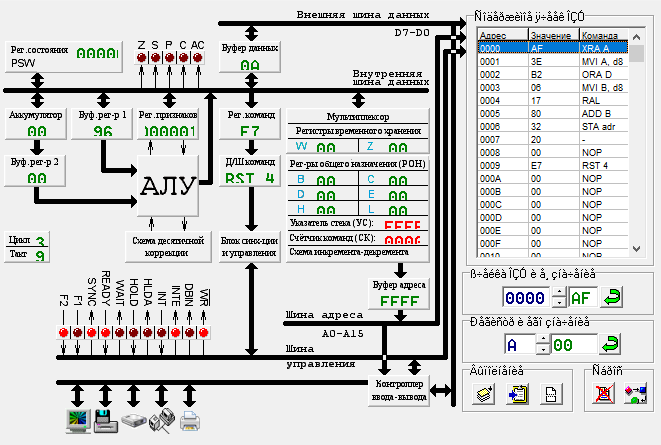


Рисунок 13 – Эмулятор перед выполнением программы

На рисунке 14 продемонстрирован эмулятор после выполнения программы (в аккумуляторе сохранилась сумма С9, она также расположена в ОЗУ по адресу 0020).

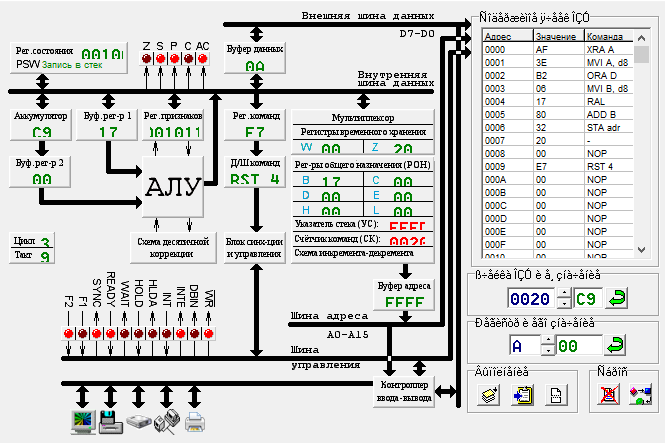


Рисунок 14 – Результат работы программы

# Лабораторная работа №8. Операция вычитания

Цель: 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.

Содержание: выполнить вычитание 234 – 56 = 178 (B2).

Программный код продемонстрирован в таблице 9.

**Таблица 9 – Программный код вычитания двух чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Значение | Команда | Комментарий |
| 0000 | AF | XRA A | Очистка аккумулятора |
| 0001 | 3E | MVI A, d8 | Загрузка в регистр А значения EA |
| 0002 | EA | JPE adr |
| 0003 | 06 | MVI B, d8 | Загрузка в регистр В значения 38 |
| 0004 | 38 | - |
| 0005 | 90 | SUB B | Вычитание содержимого регистра В из аккумулятора |
| 0006 | 32 | STA adr | Сохранение содержимого аккумулятора в память по адресу 0020 |
| 0007 | 20 | - |
| 0008 | 00 | NOP |
| 0009 | E7 | RST 4 | Прервать выполнение программы |
| 000A | 00 | NOP |  |

На рисунке 15 продемонстрирован эмулятор до начала выполнения программного кода продемонстрированного в таблице 9.

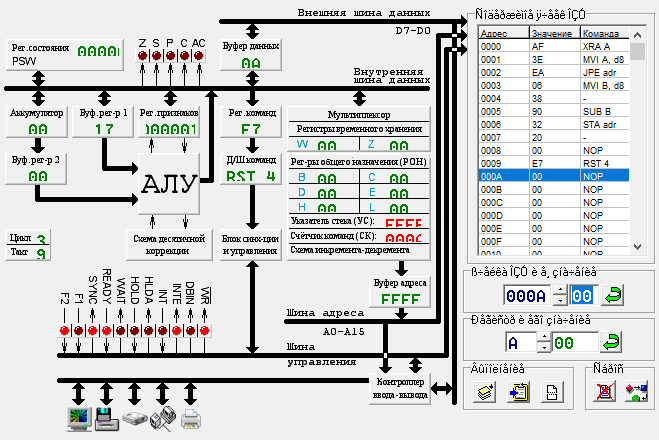


Рисунок 15 – Эмулятор до выполнения программы вычитания

На рисунке 16 продемонстрирован эмулятор после выполнения программы.

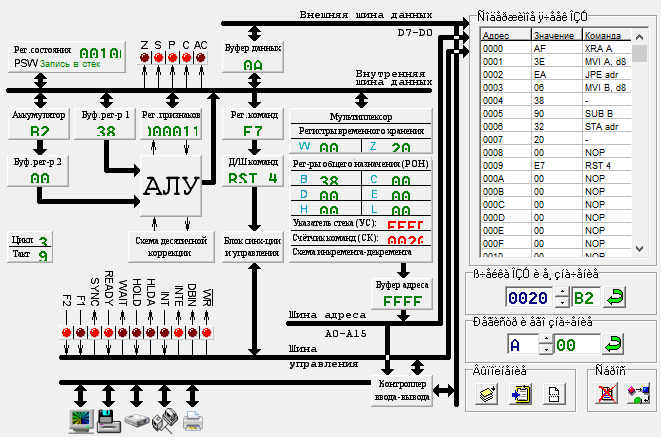


Рисунок 16 – Результат работы программы

# Лабораторная работа № 9. Сложения элементов массива

Цель: 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.

Содержание: выполнить сложение суммы элементов.

В таблице 10 продемонстрирован программный код.

**Таблица 10 – Программный код сложения элементов массива**

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 21 | LXI H, d16 | Загрузить в регистры HL, адрес первого слагаемого |
| 0001 | 50 | MOV D, B |
| 0002 | 00 | NOP |
| 0003 | 0E | MVI C, d8 | Загрузить в регистр С количество слагаемых |
| 0004 | 07 | RLC |
| 0005 | AF | XRA A | Очистить аккумулятор |
| 0006 | 47 | MOV B, A | Очистить регистр В |
| 0007 | 86 | ADD M | Прибавить к содержимому аккумулятора число из массива слагаемых |
| 0008 | D2 | JNC adr | Если переноса нет, то идти на М2 |
| 0009 | 0D | DCR C |
| 000A | 00 | NOP |
| 000B | 04 | INR B | Увеличить содержимое регистра В на 1  Очистить флаг переноса |
| 000C | B7 | ORA A |
| 000D | 23 | INX H | Указать на следующий адрес слагаемого |
| 000E | 0D | DCR C | Уменьшить содержимое регистра С на 1 |
| 000F | C2 | JNZ adr | Если не все слагаемые, то идти на M1 |
| 0010 | 07 | RLC |
| 0011 | 00 | NOP |
| 0012 | 32 | STA adr | Перенести из аккумулятора по адресу 0020 |
| 0013 | 20 | - |
| 0014 | 00 | NOP |
| 0015 | 78 | MOV A, B | Перенести из регистра В с аккумулятор |
| 0016 | 32 | STA adr | Перенести из аккумулятора по адресу 0021 |
| 0017 | 21 | LXI H, d16 |
| 0018 | 00 | NOP |
| 0019 | 76 | HLT | Конец |
| … | … | … | … |
| 0050 | 22 | SHLD adr |  |
| 0051 | 59 | MOV E, C |  |
| 0052 | 7F | MOV A, A |  |
| 0053 | 30 | - |  |
| 0054 | 20 | - |  |
| 0055 | 2F | CMA |  |
| 0056 | 2B | DCX H |  |
| 0057 | 00 | NOP |  |

На рисунке 17 продемонстрирован эмулятор до выполнения программы.

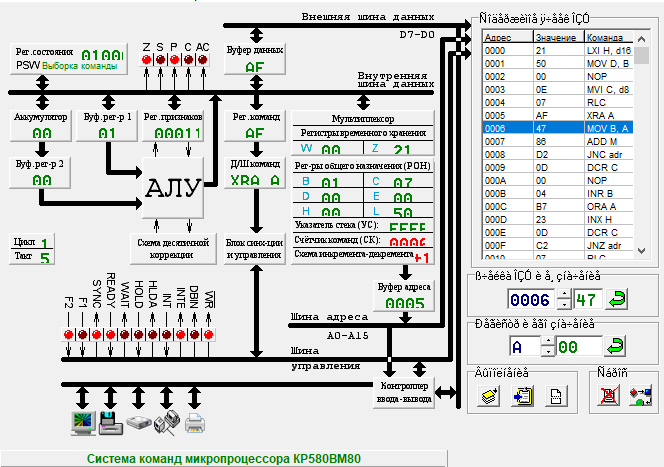


Рисунок 17 – Эмулятор до выполнения программы

На рисунке 18 продемонстрирован эмулятор после выполнения данной программы. Результат сохранен в ОЗУ по адресу 0020-0021.

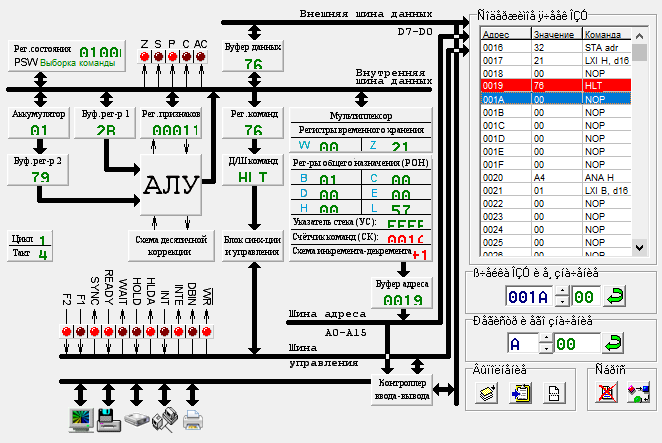


Рисунок 18 – Эмулятор после выполнения программы

# Лабораторная работа № 10. Сложение двухбайтовых десятичных чисел

Цель: 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.

Содержание: сложить два двухбайтовых числа 1434+3458 = 4892   
(059A + 0D82 = 131C)

В таблице 11 продемонстрирован программный код сложения двухбайтных чисел.

**Таблица 11 – Программный код сложения двухбайтных чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| № ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Младший байт первого слагаемого заносится в аккумулятор A (9A) |
| 0001 | 9A | SBB D |
| 0002 | 06 | MVI B, d8 | Младший байт второго слагаемого заносится в регистр B (82) |
| 0003 | 82 | ADD D |  |
| 0004 | 80 | ADD B | Прибавляем B к А |
| 0005 | 32 | STA adr | Заносим А в память (0012) |
| 0006 | 12 | STAX D |
| 0007 | 00 | NOP |
| 0008 | 3E | MVI A, d8 | Старший байт первого слагаемого заносится в аккумулятор А(05) |
| 0009 | 05 | DCR B |
| 000A | 06 | MVI B, d8 | Старший байт первого слагаемого заносится в регистр В (0D) |
| 000B | 0D | DCR C |
| 000C | 88 | ADC B | Прибавляем B к А |
| 000D | 32 | STA adr | Заносим А в память (0013) |
| 000E | 13 | INX D |
| 000F | 00 | NOP |
| 0010 | 76 | HLT | Конец |

На рисунке 19 продемонстрирован эмулятор до выполнения программы.

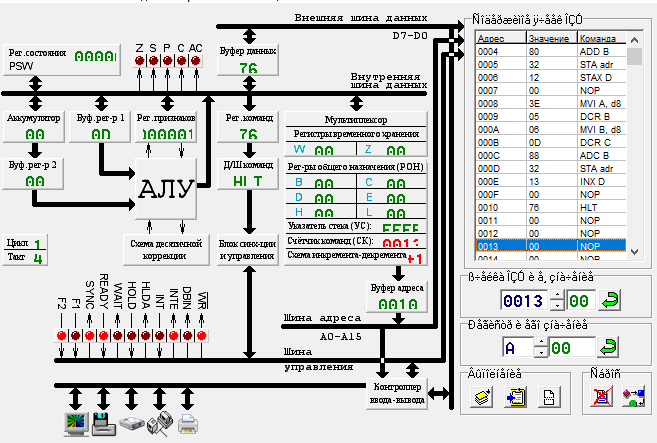


Рисунок 19 – Эмулятор до выполнения программы

На рисунке 20 продемонстрирован результат выполнения программного кода (результат в 0012 0013)

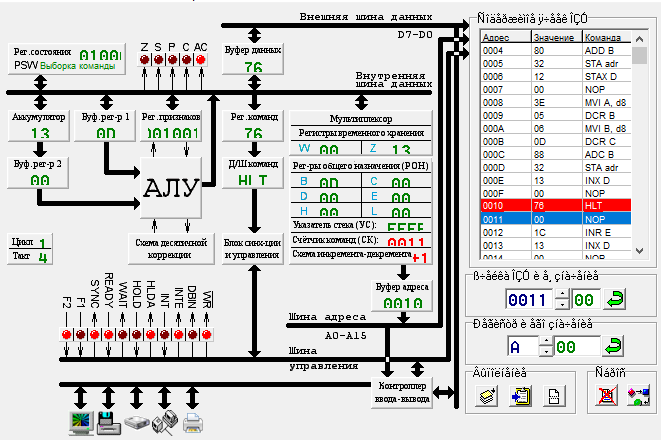


Рисунок 20 – Результат выполнения программы

# Лабораторная работа № 11. Вычитание двухбайтовых десятичных чисел

Цель: 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.

Содержание: вычесть два двухбайтовых числа 40267-34308= 5959   
(9D4B - 8604 = 1747)

В таблице 12 продемонстрирован программный код сложения двухбайтных чисел.

**Таблица 12 – Сложение двухбайтных чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| № ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Младший байт уменьшаемого заносится в аккумулятор A (9A) |
| 0001 | 4B | MOV C, E |  |
| 0002 | D6 | SUI d8 | Младший байт вычитаемого заносится в регистр B (82) |
| 0003 | 04 | INR B |  |
| 0004 | 32 | STA adr | Вычитаем из А к B |
| 0005 | 10 | - | Заносим А в память (0012) |
| 0006 | 00 | NOP |  |
| 0007 | 3E | MVI A, d8 | Старший байт уменьшаемого заносится в аккумулятор А(9D) |
| 0008 | 9D | SBB L |  |
| 0009 | DE | SBI d8 | Старший байт вычитаемого заносится в регистр В (0D) |
| 000A | 86 | ADD M |  |
| 000B | 32 | STA adr | Прибавляем B к А |
| 000C | 11 | LXI D, d16 | Заносим А в память (0013) |
| 000D | 00 | NOP |  |
| 000E | 76 | HLT | Конец |
| 000F | 00 | NOP |  |
| 0010 | 47 | MOV B, A |  |
| 0011 | 17 | RAL |  |

На рисунке 21 продемонстрирован эмулятор до запуска программного кода.

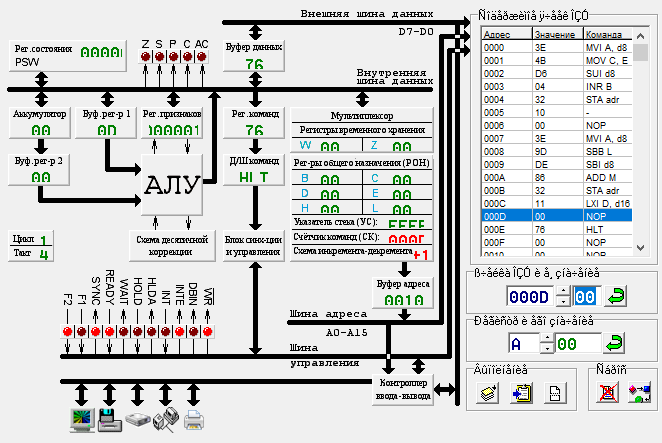


Рисунок 21 – Эмулятор до запуска программного кода

На рисунке 22 продемонстрирован результат работы программы.

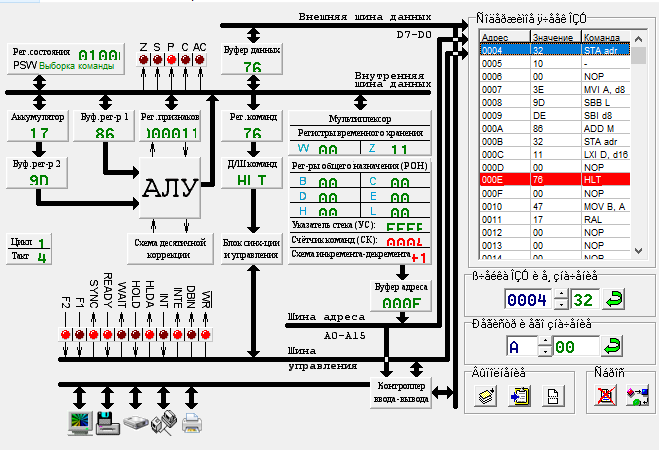


Рисунок 22 – Эмулятор после выполнения программного кода

# Лабораторная работа № 12. Алгоритмы ветвления

Цель: рассмотреть особенности выполнения алгоритмов ветвлений, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

Содержание: реализовать следующий алгоритм ветвления формулы 1.

(1)

В таблице 13 продемонстрирован программный код для данной функции.

**Таблица 13 – Программный код ветвления**

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3A | LDA adr | Перенос в аккумулятор по адресу 0020 |
| 0001 | 20 | - |
| 0002 | 00 | NOP |
| 0003 | FE | CPI d8 | Вычесть из аккумулятора 06 |
| 0004 | 06 | MVI B, d8 |
| 0005 | FA | JM adr | Если итог < 0 то переход к 0010 адресу |
| 0006 | 10 | - |
| 0007 | 00 | NOP |
| 0008 | 47 | MOV B, A | Перенос из аккумулятора в регистр В |
| 0009 | 80 | ADD B | Прибавить к аккумулятору содержимое регистра В |
| 000A | 80 | ADD B | Прибавить к аккумулятору содержимое регистра В |
| 000B | 32 | STA adr | Перенесли содержимое аккумулятора по адресу 0021 |
| 000C | 21 | LXI H, d16 |
| 000D | 00 | NOP |
| 000E | 76 | HLT | Конец |
| 000F | 00 | NOP |  |
| 0010 | 47 | MOV B, A | Перенос из аккумулятора в регистр В |
| 0011 | 80 | ADD B | Прибавить к аккумулятору содержимое регистра В |
| 0012 | 06 | MVI B, d8 | Перенос числа 02 в регистр В |
| 0013 | 02 | STAX B |
| 0014 | 80 | ADD B | Прибавить к аккумулятору содержимое регистра В |
| 0015 | 32 | STA adr | Перенесли содержимое аккумулятора по адресу 0021 |
| 0016 | 21 | LXI H, d16 |
| 0017 | 00 | NOP |
| 0018 | 76 | HLT | Конец |
| 0019 | 00 | NOP |  |

На рисунке 1 продемонстрирован эмулятор до выполнения программы. В данном случае проверяется ветвь x<6 (x = 510, y = 1510 (F16)).

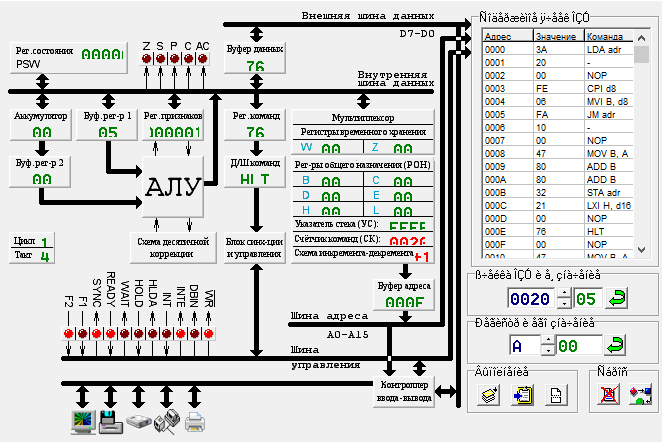


Рисунок – Проверка верхней ветви. Эмулятор до выполнения программы

Результат работы программы продемонстрирован на рисунке 2 (в области памяти 0021).

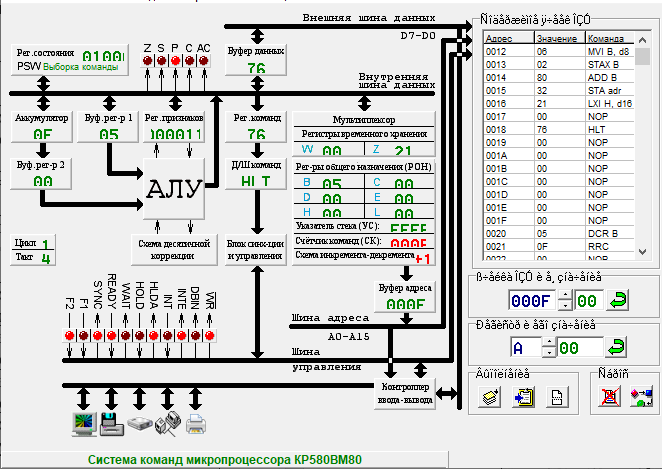


Рисунок – Результат работы программы при прохождении верхней ветви

На рисунке 3 продемонстрирован эмулятор перед запуском программы со следующими исходными данными (x=6, y должен стать равным 1410 (E16))

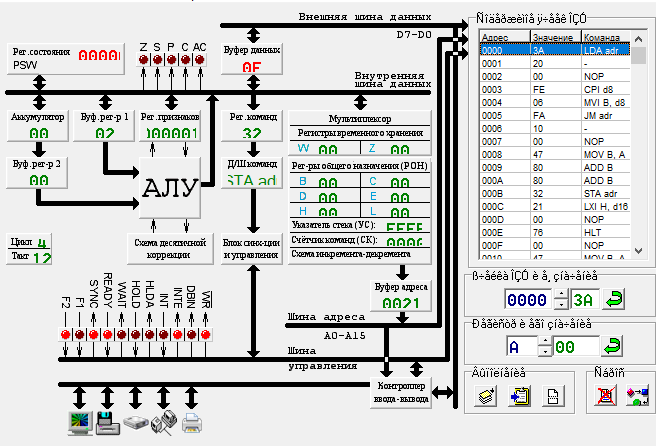


Рисунок 3 – Проверка ветви, когда вводимое число больше 6

Результат проверки побочной ветви продемонстрирован на рисунке 4 (в области памяти 0021).

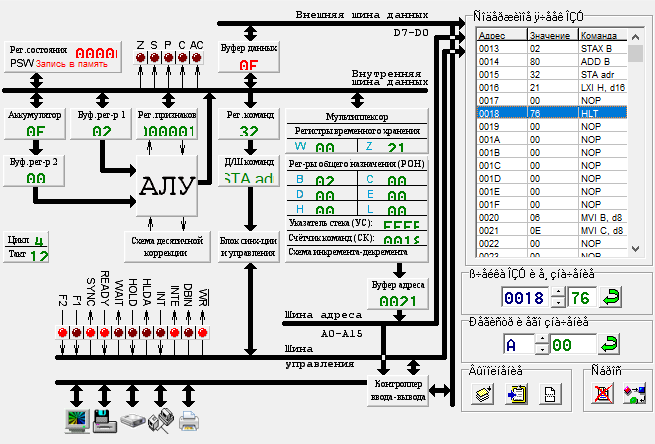


Рисунок 4 – Результат проверки

# Лабораторная работа № 13. Вывод информации на экран

Цель: изучить возможности вывода информации на экран

Содержание: вывести номер группы, фамилию и имя на экран разными цветами.

На рисунке 1 продемонстрирован вывод номер группы, фамилии и имени разными цветами.

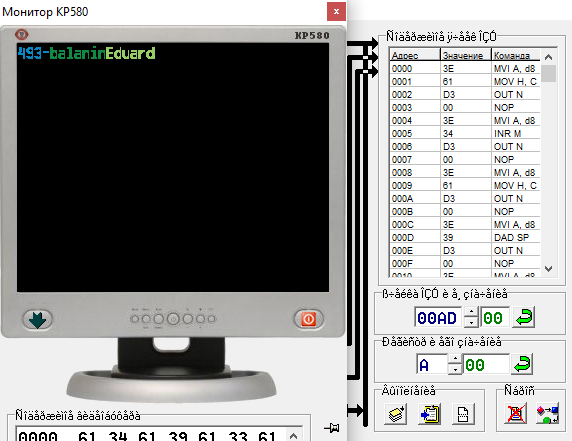


Рисунок 1 – Вывод информации на экран

Листинг программного кода продемонстрирован на таблице 14.

**Таблица 14 – Листинг программного кода задания**

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Бирюзовый цвет |
| 0001 | 61 | MOV H, C |
| 0002 | D3 | OUT N |
| 0003 | 00 | NOP |
| 0004 | 3E | MVI A, d8 | Цифра 4 |
| 0005 | 34 | INR M |
| 0006 | D3 | OUT N |
| 0007 | 00 | NOP |
| 0008 | 3E | MVI A, d8 | Бирюзовый цвет |
| 0009 | 61 | MOV H, C |
| 000A | D3 | OUT N |
| 000B | 00 | NOP |
| 000C | 3E | MVI A, d8 | Цифра 9 |
| 000D | 39 | DAD SP |
| 000E | D3 | OUT N |
| 000F | 00 | NOP |
| 0010 | 3E | MVI A, d8 | Бирюзовый цвет |
| 0011 | 61 | MOV H, C |
| 0012 | D3 | OUT N |
| 0013 | 00 | NOP |
| 0014 | 3E | MVI A, d8 | Цифра 3 |
| 0015 | 33 | INX SP |
| 0016 | D3 | OUT N |
| 0017 | 00 | NOP |
| 0018 | 3E | MVI A, d8 | Бирюзовый цвет |
| 0019 | 61 | MOV H, C |
| 001A | D3 | OUT N |
| 001B | 00 | NOP |
| 001C | 3E | MVI A, d8 | Символ «-» |
| 001D | 2D | DCR L |
| 001E | D3 | OUT N |
| 001F | 00 | NOP |
| 0020 | 3E | MVI A, d8 | Зеленый цвет |
| 0021 | 22 | SHLD adr |
| 0022 | D3 | OUT N |
| 0023 | 00 | NOP |
| 0024 | 3E | MVI A, d8 | Буква «b» |
| 0025 | 62 | MOV H, D |
| 0026 | D3 | OUT N |
| 0027 | 00 | NOP |
| 0028 | 00 | NOP | Зеленый цвет |
| 0029 | 3E | MVI A, d8 |
| 002A | 22 | SHLD adr |
| 002B | D3 | OUT N |
| 002C | 00 | NOP | Буква «a» |
| 002D | 3E | MVI A, d8 |
| 002E | 61 | MOV H, C |
| 002F | D3 | OUT N |
| 0030 | 00 | NOP | Зеленый цвет |
| 0031 | 3E | MVI A, d8 |
| 0032 | 22 | SHLD adr |
| 0033 | D3 | OUT N |
| 0034 | 00 | NOP | Буква «l» |
| 0035 | 3E | MVI A, d8 |
| 0036 | 6C | MOV L, H |
| 0037 | D3 | OUT N |
| 0038 | 00 | NOP | Зеленый цвет |
| 0039 | 3E | MVI A, d8 |
| 003A | 22 | SHLD adr |
| 003B | D3 | OUT N |
| 003C | 00 | NOP | Буква «a» |
| 003D | 3E | MVI A, d8 |
| 003E | 61 | MOV H, C |
| 003F | D3 | OUT N |
| 0040 | 00 | NOP | Зеленый цвет |
| 0041 | 3E | MVI A, d8 |
| 0042 | 22 | SHLD adr |
| 0043 | D3 | OUT N |
| 0044 | 00 | NOP | Буква «n» |
| 0045 | 3E | MVI A, d8 |
| 0046 | 6E | MOV L, M |
| 0047 | D3 | OUT N |
| 0048 | 00 | NOP | Зеленый цвет |
| 0049 | 3E | MVI A, d8 |
| 004A | 22 | SHLD adr |
| 004B | D3 | OUT N |
| 004C | 00 | NOP | Буква «i» |
| 004D | 3E | MVI A, d8 |
| 004E | 69 | MOV L, C |
| 004F | D3 | OUT N |
| 0050 | 00 | NOP | Зеленый цвет |
| 0051 | 3E | MVI A, d8 |
| 0052 | 22 | SHLD adr |
| 0053 | D3 | OUT N |
| 0054 | 00 | NOP | Буква «n» |
| 0055 | 3E | MVI A, d8 |
| 0056 | 6E | MOV L, M |
| 0057 | D3 | OUT N |
| 0058 | 00 | NOP | Желтый цвет |
| 0059 | 3E | MVI A, d8 |
| 005A | 33 | INX SP |
| 005B | D3 | OUT N |
| 005C | 00 | NOP | Буква «E» |
| 005D | 3E | MVI A, d8 |
| 005E | 45 | MOV B, L |
| 005F | D3 | OUT N |
| 0060 | 00 | NOP | Желтый цвет |
| 0061 | 3E | MVI A, d8 |
| 0062 | 33 | INX SP |
| 0063 | D3 | OUT N |
| 0064 | 00 | NOP | Буква «d» |
| 0065 | 3E | MVI A, d8 |
| 0066 | 64 | MOV H, H |
| 0067 | D3 | OUT N |
| 0068 | 00 | NOP | Желтый цвет |
| 0069 | 3E | MVI A, d8 |
| 006A | 33 | INX SP |
| 006B | D3 | OUT N |
| 006C | 00 | NOP | Буква «u» |
| 006D | 3E | MVI A, d8 |
| 006E | 75 | MOV M, L |
| 006F | D3 | OUT N |
| 0070 | 00 | NOP | Желтый цвет |
| 0071 | 3E | MVI A, d8 |
| 0072 | 33 | INX SP |
| 0073 | D3 | OUT N |
| 0074 | 00 | NOP | Буква «a» |
| 0075 | 3E | MVI A, d8 |
| 0076 | 61 | MOV H, C |
| 0077 | D3 | OUT N |
| 0078 | 00 | NOP | Желтый цвет |
| 0079 | 3E | MVI A, d8 |
| 007A | 33 | INX SP |
| 007B | D3 | OUT N |
| 007C | 00 | NOP | Буква «r» |
| 007D | 3E | MVI A, d8 |
| 007E | 72 | MOV M, D |
| 007F | D3 | OUT N |
| 0080 | 00 | NOP | Желтый цвет |
| 0081 | 3E | MVI A, d8 |
| 0082 | 33 | INX SP |
| 0083 | D3 | OUT N |
| 0084 | 00 | NOP | Буква «d» |
| 0085 | 3E | MVI A, d8 |
| 0086 | 64 | MOV H, H |
| 0087 | D3 | OUT N |
| 0088 | 00 | NOP |  |
| 0089 | 00 | NOP |  |

# Лабораторная работа № 14. Умножение целых чисел без знака

Цель: изучение возможностей МП для умножения целых чисел без знака

Содержание: написать программу выполняющую умножение двух чисел: 23 \* 16 = 368 (17 16 \* 1016 = 17016).

Программный код предоставлен в таблице 15.

**Таблица 15 – Программный код умножения двух чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код  команды | Мнемоника | Комментарий |
| 0000 | 00 |  | Не используется |
| 0001 | 1E | MVI E,10h | Записать в регистр E число 1716 (2310) |
| 0002 | 17 |
| 0003 | 0E | MVI C,1Ah | Записать в регистр C число 1016 (1610) |
| 0004 | 10 |
| 0005 | AF | XRA A | Очистить аккумулятор |
| 0006 | 67 | MOV H,A | Очистить регистр Н |
| 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 А = 0, то выход |
| 000B | AF | XRA A | Очистить аккумулятор |
| 000C | 81 | ADD C | A – множитель |
| 000D | C8 | RZ | Если А = 0, то выход |
| 000E | 1F | RAR | Арифметический сдвиг аккумулятора вправо |
| 000F | D2 | JNС С2 | Флаг переноса не установлен |
| 0010 | 13 |
| 0011 | 00 |
| 0012 | 19 | DAD D | Сдвиг множимого влево |
| 0013 | EB | XCHG | Обменять пару регистров DE с парой HL |
| 0014 | 29 | DAD H | Прибавить к паре регистров HL содержимое пары регистров H |
| 0015 | EB | XCHG | Обменять пару регистров DE с парой HL |
| 0016 | B7 | ORA A | Логическое сложение «ИЛИ» аккумулятора. В данном случае очистить флаг переноса |
| 0017 | C2 | JNZ С1 | Флаг нуля не установлен(JUMP IF NOT ZERO) |
| 0018 | 0E |
| 0019 | 00 |
| 001A | 76 | HLT | Конец |

На рисунке 1 продемонстрирован эмулятор до выполнения программного кода продемонстрированного в таблице 15.

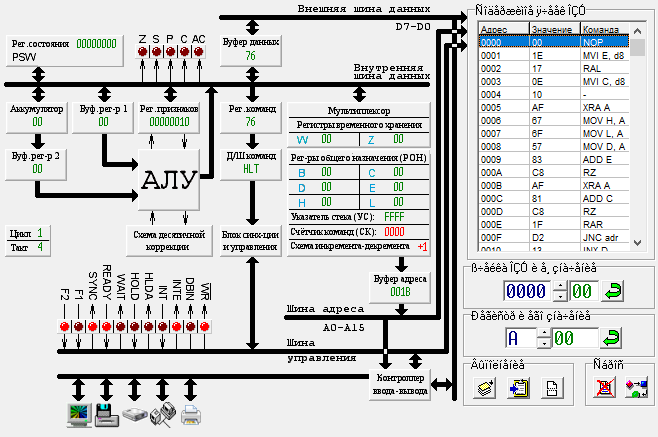


Рисунок 1 – Эмулятор до выполнения программы

На рисунке 2 продемонстрирован эмулятор после выполнения программного кода (результат умножения расположен в регистрах хранения мультипроцессора).

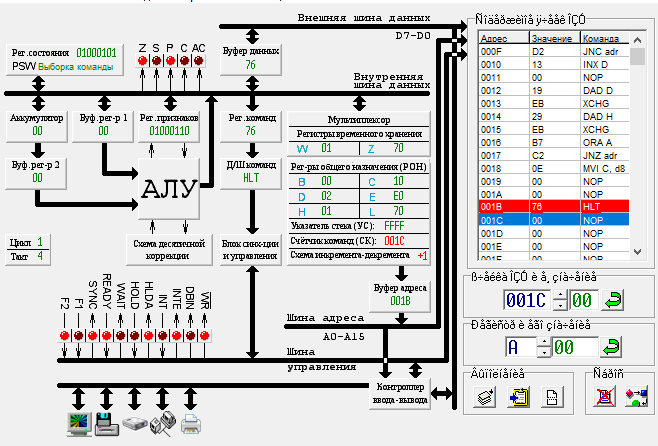


Рисунок 2 – Результат выполнения программы

# Лабораторная работа № 15. Деление однобайтных чисел

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

Содержание: выполнить деление чисел 180 / 20 = 9 (B416 / 1416 = 916)

Таблица 16 содержит программный код предназначенный для выполнения деления 180 на 20.

**Таблица 16 – Программный код деления**

|  |  |  |  |
| --- | --- | --- | --- |
| № ячейки | Код команды | Мнемоника | Комментарии |
| 0000 | 11 | LXI D, 14A0 | Загрузка делимого и делителя в пару DE |
| 0001 | А0 |  |
| 0002 | 14 |  |
| 0003 | 21 | LXI H, 0008 | Загрузка счетчика в регистр L и очистка регистра Н |
| 0004 | 08 |  |
| 0005 | 00 |  |
| 0006 | 0Е | MVI C, 0 | Очистка регистра промежуточного остатка |
| 0007 | 00 |  |
| 0008 | 7B | MOV A, E | Загрузка делимого в аккумулятор |
| 0009 | 17 | RAL | Смещение делимого влево на один бит |
| 000А | 5F | MOV E, A | Пересылка делимого в регистр Е |
| 000В | 79 | MOV A, C | Загрузка промежуточного остатка в А |
| 000С | 17 | RAL | Смещение промежуточного остатка влево |
| 000D | 92 | SUB D | Вычитание делителя от А |
| 000Е | D2 | JNC NOADD | Если С = 0 - переход на метку NOADD |
| 000F | 12 |  |
| 0010 | 00 |  |
| 0011 | 82 | ADD D | Восстановление остатка |
| 0012 | 4F | MOV C, A | Пересылка промежуточного остатка в регистр С |
| 0013 | 3F | CMC | Инверсия признака переполнения |
| 0014 | 7C | MOV A, H | Запись доли в А |
| 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 |  |
| 001А | 00 |  |
| 001В | 7C | MOV A, H | Пересылка младшего байта результата в А |
| 001С | 32 | STA 0050 | Сохранение содержимого А в памяти по адресу 0030 |
| 001D | 30 |  |
| 001E | 00 |  |
| 001F | 79 | MOV A, H | Пересылка старшего байта результата в А |
| 0020 | 32 | STA 0051 | Сохранение содержимого А в памяти по адресу 0031 |
| 0021 | 31 |  |
| 0022 | 00 |  |
| 0023 | CF | RST1 | Прекращение выполнения программы |

На рисунке 1 продемонстрирован программный код перед запуском программы продемонстрированной в таблице 16.

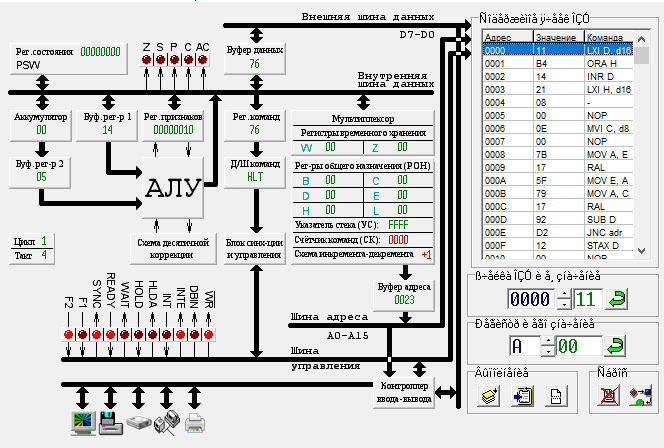


Рисунок 1 – Эмулятор до выполнения программы

На рисунке 2 продемонстрирован эмулятор после выполнения программы (результат деления помещен по адресу 0025 0026).

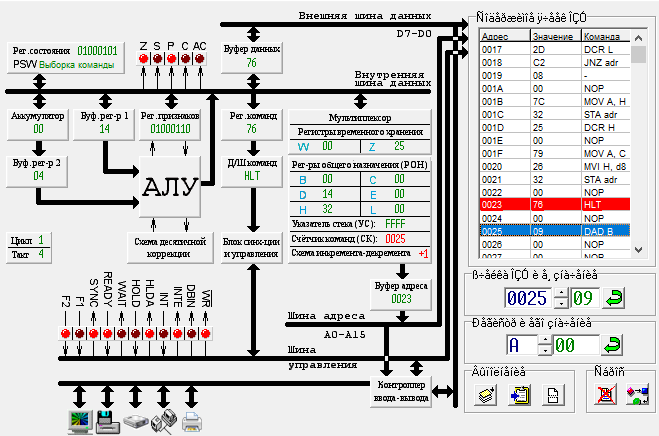


Рисунок – Результат выполнения программы

# Лабораторная работа № 16. Арифметические операции

Цель: 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.

Содержание: В таблице 17 продемонстрирован программный код логического умножения чисел 105&209 (6616&D116 = 4016).

**Таблица 17 – Программный код логического умножения**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A,d8 | Перенос 66 в аккумулятор |
| 0001 | 66 | MOV H, M |
| 0002 | 06 | MVI B, d8 | Перенос D1 в регистр B |
| 0003 | D1 | POP D |
| 0004 | A0 | ANA B | Логическое умножение регистра В на сумматор |
| 0005 | 76 | HLT | Конец |

На рисунке 1 продемонстрирован эмулятор до выполнения программы.

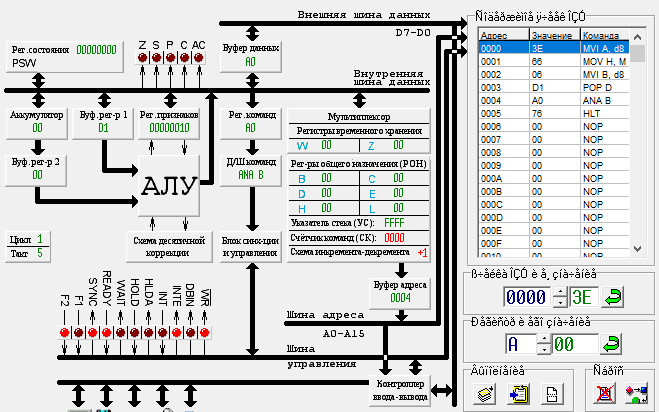


Рисунок 1 – Эмулятор до выполнения программы

На рисунке 2 продемонстрирован результат логического умножения чисел 105 и 209.

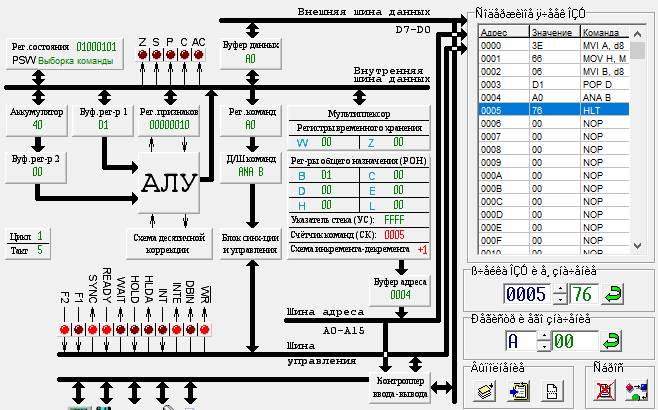


Рисунок 2 – Результат логического умножения

Задание: выполнить обнуление пятого бита в коде ASCII символа «a». Символу a соответствует код 9710=6116

Для обнуления пятого бита данного кода понадобится маска EF16

Программный код продемонстрирован в таблице 18.

**Таблица 18 – Программный код обнуления пятого бита**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 21 | LXI H, d16 | Загрузка в HL адреса 0040 |
| 0001 | 40 | MOV B, B |
| 0002 | 00 | NOP |
| 0003 | 3E | MOV A,d8 | Помещаем в аккумулятор 6116 |
| 0004 | 61 | MOV H,C |
| 0005 | 36 | MVI M,d8 | Помещаем в регистр В маску EF |
| 0006 | CF | RST 5 |
| 0007 | А6 | ANA M | Логическое умножение |
| 0008 | 76 | HLT | Конец |

На рисунке 3 продемонстрирован эмулятор до выполнения программы.

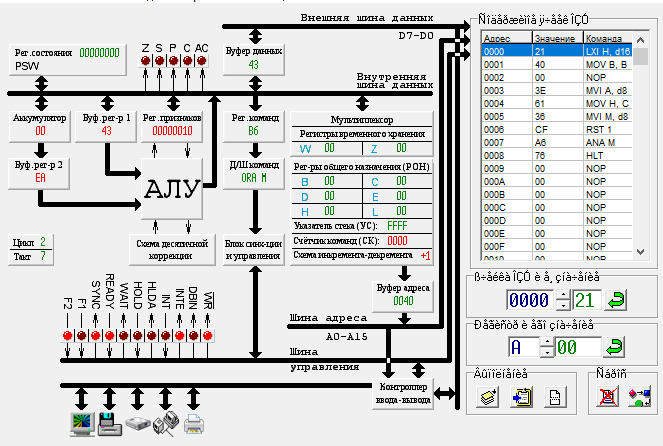


Рисунок 3 – Эмулятор до выполнения программы

На рисунке 4 продемонстрирован результат программного кода.

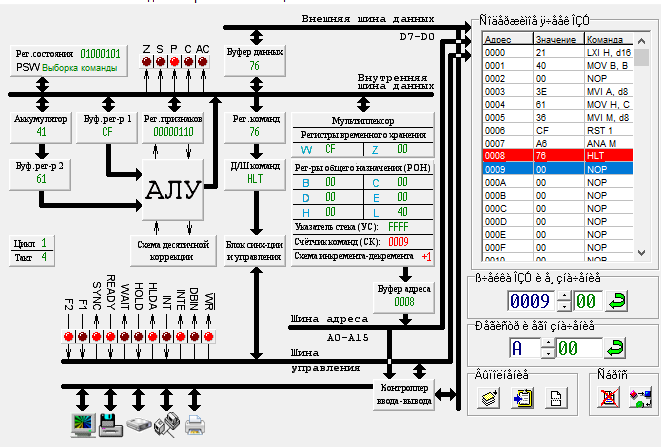


Рисунок 4 – Результат выполненного программного кода

Задание: выполнить логическое сложение чисел 234+67 (EA16 + 4316= EB16)

Программный код логического сложения чисел представлен в таблице 19.

**Таблица 19 – Логическое сложение чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 21 | LXI H, d16 | Загрузка в HL адреса 0040 |
| 0001 | 40 | MOV B, B |  |
| 0002 | 00 | NOP |  |
| 0003 | 3E | MOV A,d8 | Помещаем в аккумулятор EA16 |
| 0004 | EA | JPE adr |
| 0005 | 36 | MVI M,d8 | Помещаем в регистр В 4316 |
| 0006 | 43 | MOV B,E |
| 0007 | B6 | ORA M | Логическое сложение |
| 0008 | 76 | HLT | Конец |

На рисунке 5 продемонстрирован эмулятор перед выполнением программы.

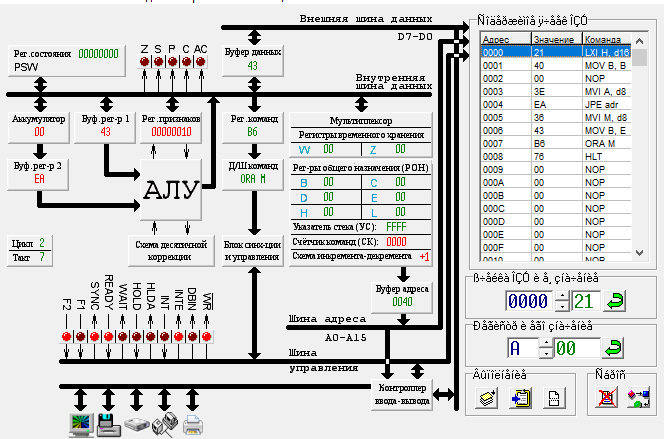


Рисунок 5 – Эмулятор до выполнения программы

На рисунке 6 продемонстрирован результат работы программы логического сложения.

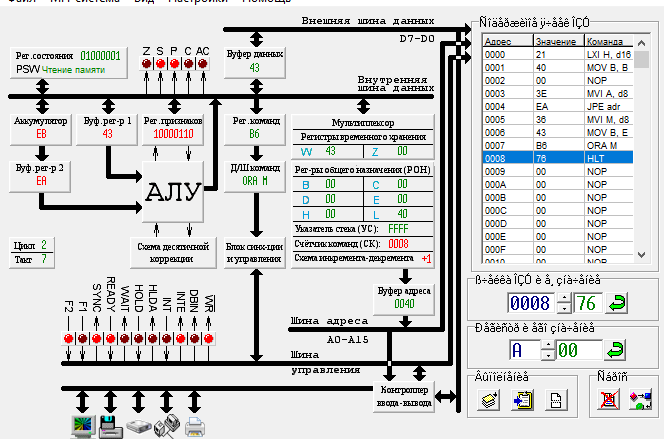


Рисунок 6 – Результат работы программы

Задание: восстановить пятый бит кода символа «Q» (8110=5116)

Программный код предоставлен в таблице 20.

**Таблице 20 – Программный код восстановления пятого символа**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A, d8 | Занести 51 в аккумулятор |
| 0001 | 51 | MOV D, C |  |
| 0002 | 0E | MVI C, d8 | Занести 20 в регистр С |
| 0003 | 20 |  |  |
| 0004 | B1 | ORA C | Логическое сложение |
| 0005 | 76 | HLT |  |

На рисунке 7 продемонстрирован эмулятор перед выполнением программы.

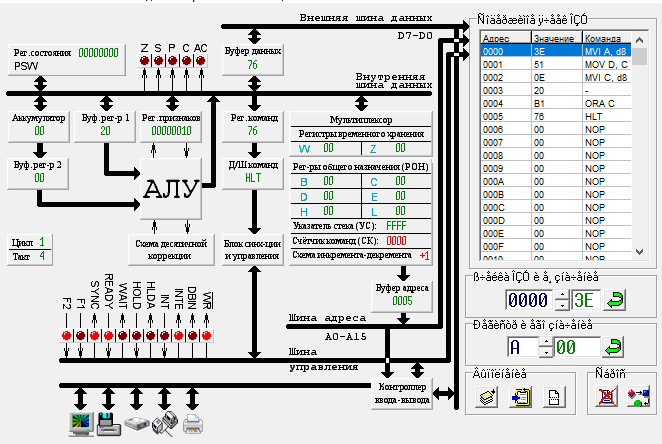


Рисунок 7 – Эмулятор до выполнения программы

На рисунке 8 продемонстрирован результат работы программы восстановления бита.

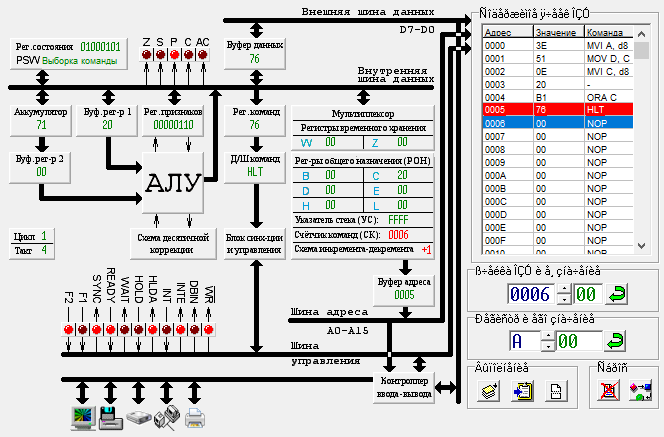


Рисунок 8 – Результат работы программы

Задание: выполнить инверсию битов числа 178 (b216)

Программный код продемонстрирован в таблице 21.

**Таблица 21 – Программный код инверсии битов числа**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MIV A,d8 | Перенос в аккумулятор числа B2 |
| 0001 | B2 | ORA D |
| 0002 | 2F | CMA | Инверсия битов |
| 0003 | 76 | HLT | Конец |

На рисунке 9 продемонстрирован эмулятор перед выполнением программы.

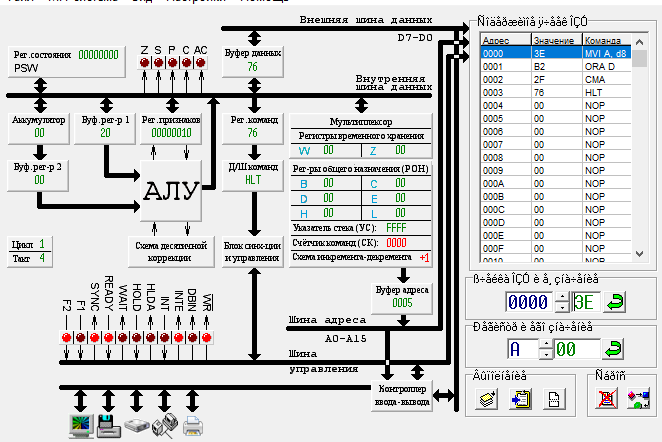


Рисунок 9 – Эмулятор до выполнения программы

На рисунке 10 продемонстрирован результат работы программы восстановления бита.

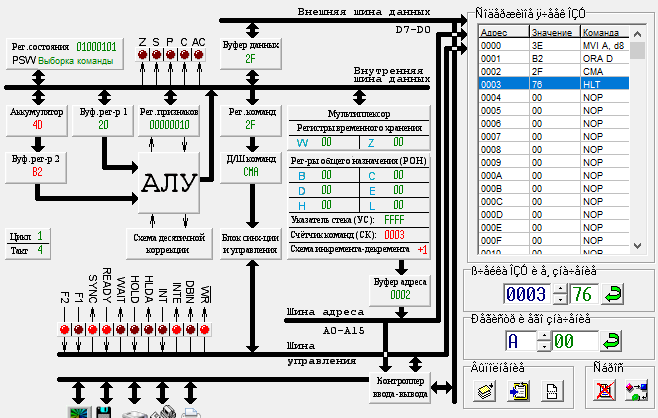


Рисунок 10 – Результат работы программы

Задание: инвертировать 2,4,5 биты числа 187 (5716)

Для инвертирования битов 2, 4, 5 понадобится маска 3416. Результатом инвертирования окажется число 6316. Программный код продемонстрирован в таблице 22.

**Таблица 22 – Программный код инверсии 2, 4, 5 битов**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A, d8 | Перенос 57 в аккумулятор |
| 0001 | 57 | MOV D,A |
| 0002 | 0E | MVI C, d8 | Перенос маски 34 в регистр С |
| 0003 | 34 | INR M |
| 0004 | A9 | XRA C | XOR над аккумулятором и регистром С |
| 0005 | 76 | HLT | Конец |

На рисунке 11 продемонстрирован эмулятор перед выполнением программы.

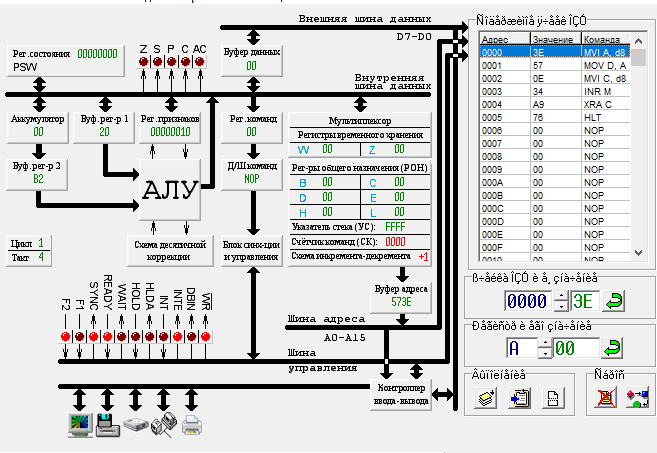


Рисунок 11 - Эмулятор до выполнения программы

На рисунке 12 продемонстрирован результат работы программы восстановления бита.

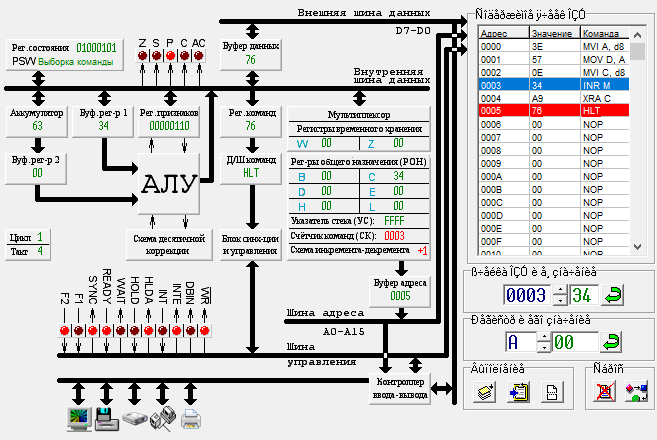


Рисунок 12 - Результат работы программы

Задание: выполнить сравнение двух чисел: 245 и 86 (F516 и 8616). Результат записать в ОЗУ 000A.

В таблице 23 продемонстрирован программный код сравнения двух чисел.

**Таблица 23 – Программный код сравнения двух чисел**

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A, d8 | Положить в сумматор F5 |
| 0001 | F5 | PUSH PSV |
| 0002 | EE | XRI, d8 | Xor между содержимым сумматора и вторым байтом |
| 0003 | 86 | ADD M |
| 0004 | 32 | STA adr | Положить из сумматора в ОЗУ 000A |
| 0005 | 0A | LDAX B |
|  | 00 |  |
| 0007 | 76 | HLT | Конец |

На рисунке 11 продемонстрирован эмулятор перед выполнением программы.

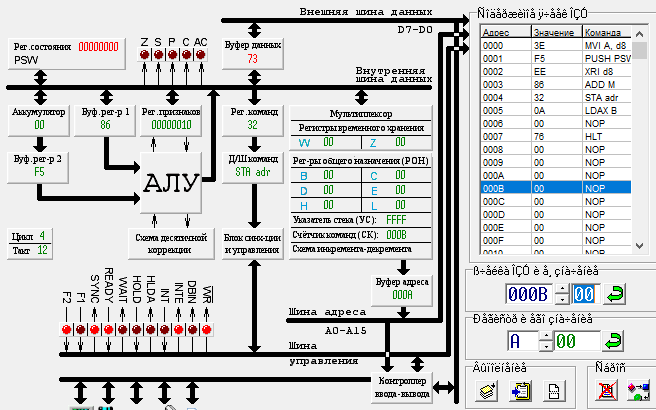


Рисунок 11 - Эмулятор до выполнения программы

На рисунке 12 продемонстрирован результат работы программы сравнения двух чисел (результат в 000A).

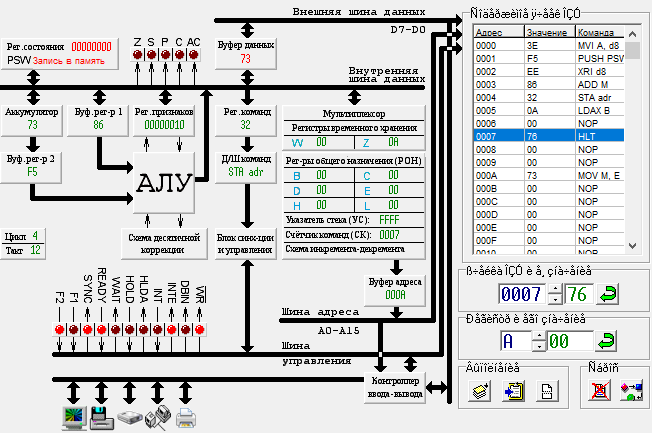


Рисунок 12 – Результат работы программы