МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»** Факультет автоматики и вычислительной техники Кафедра электронных вычислительных машин

Отчет по лабораторной работе №5 по дисциплине

«Организация ЭВМ и систем» Вариант 7

Выполнил студент группы ИВТ-32 /Кудяшев Я.Ю./ Проверил преподаватель /Клюкин В.Л./

Киров 2021

1. Задание

Определить архитектуру, разработать и отладить микропрограмму командного цикла ЭВМ, составить и выполнить программу вычисления суммы частных 𝑆:

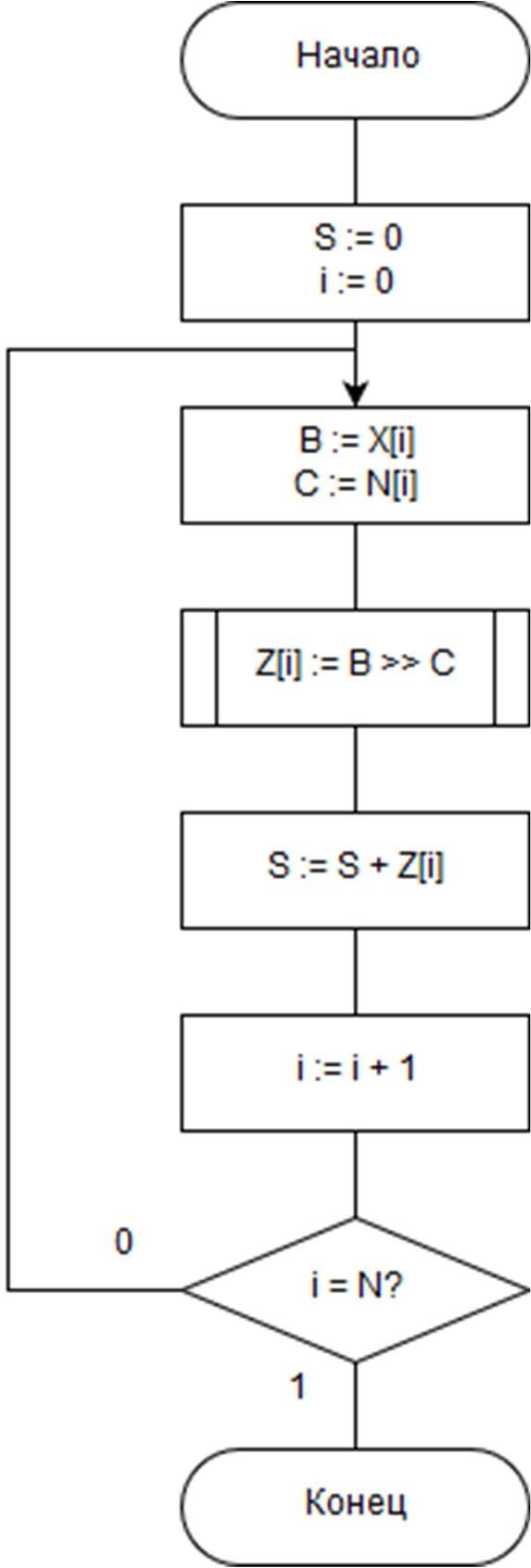
N

𝑆 = Σ 𝑥i

i=1

где 𝑥i – сумма единиц в одном из чисел. Для подсчёта единиц в числе использовать подпрограмму, составленную на основе предыдущей лабораторной работы. Обмен данными между программой и подпрограммой должен производиться через стек.

1. Определение архитектуры и программирование
   1. Схема алгоритма



* 1. Форматы данных

X изменяется в пределах 0...65536, поэтому любое число можно представить 16-разрядным двоичным кодом.

* 1. Программно-доступные регистры

ЭВМ имеет девять программно-доступных регистров: шесть регистров общего назначения (r0-r5), программный счетчик – IP (r6), регистр признаков – FLAGS (r7), содержащий разряд признака нуля (Z), а также регистр указателя стека – SP (r8).

* 1. Система команд

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Мнемоника | Описание | Изменение признака Z |
| Суммирование | ADD r r\* | r := r + r\*, IP := IP + 1 | + |
| Вычитание | SUB r r\* | r := r – r\*, IP := IP + 1 | + |
| Добавление C | AD r C | r := r + C, IP := IP + 1 | + |
| Вычитание C | SB r C | r := r – C, IP := IP + 1 | + |
| Чтение в регистр | LD r A | r := M[A], IP := IP + 1 | - |
| Запись регистра | MV r A | M[A] := r, IP := IP + 1 | - |
| Чтение в регистр с индексацией | LDI r, r\* | r := M[r\*], IP := IP + 1 | - |
| Запись в стек | PUSH r (SP) | M[SP] := r, SP := SP – 1, IP := IP + 1 | - |
| Чтение из стека | POP r (SP) | SP := SP + 1, r := M[SP], IP := IP + 1 | - |
| Переход | JMP A | IP := A | - |
| Переход, если нуль | JZ A | Если Z = 1, то IP := A, иначе IP := IP + 1 | - |
| Обращение к подпрограмме | CALL (SP) A | M[SP] := IP, SP := SP – 1, IP := IP + 1 | - |
| Возврат из подпрограммы | RET (SP) | SP := SP + 1, IP := M[SP] | - |
| Сдвиг вправо логический | SHR r r\* | r := r\* / 2, IP := IP + 1 | + |
| Останов | HLT A | IP := A, останов | - |

В описании системы команд приняты следующие обозначения:

* r, r\*∈ {r0, r1, … r8} – программно-доступные регистры: регистр r\* является источником данных, а регистр r – приемником результата, но может также служить источником второго операнда
* М[А] – ячейка памяти с адресом А
* Знак "+" в описании признаков означает, что устанавливается новое значение признака по результату выполнения команды, а знак "-" свидетельствует о сохранении старого значения признака
  1. Программа

В программе приняты следующие обозначения:

* BP – адрес ячейки памяти, в которой находится адрес начала накопителя стека
* ARRAY – адрес ячейки памяти, в которой находится начальный адрес массива исходных данных (𝑥0, 𝑛0, 𝑥1, 𝑛1, … , 𝑥n𝑛n)
* PROC – начальный адрес подпрограммы деления чисел нацело
* SUM – адрес ячейки, в которую записывается сумма S
* IPS – начальный адрес программы суммирования частных

LD SP BP Загрузка регистра указателя стека SP

LD R5 ARRAY Загрузка адреса массива ARRAY в регистр r5 LD R4 N Загрузка числа повторений цикла N в r4

SUB R3 R3 Очистка регистра r3 для суммы S m1: LDI R1 [R5] Чтение Xi в r1

AD R5 01 Увеличение r5 на 1

CALL PROC Обращение к подпрограмме по адресу PROC ADD R3 R1 Суммирование

SB R4 01 Вычитание единицы из числа повторений цикла JZ m2 Если N=0, то переход на метку m2

JMP m1 Переход на метку m1

m2: MV R3 SUM Запись суммы S адресу SUM HLT IP Загрузка IP и останов

* 1. Распределение программно-доступных регистров ЭВМ

|  |  |  |
| --- | --- | --- |
|  | Регистры ЭВМ |  |
| r0 |  | Исходное число |
| r1 | Xi | Число Xi (результат Zi) |
| r2 | Ni | Кол-во сдвигов Ni |
| r3 | S | Сумма S |
| r4 | N | Число повторений цикла N |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Регистры ЭВМ | |  |
| r5 | ARRAY | | Адрес массива |
| r6 | IP | | Программный счетчик |
| r7 | FLAGS | Z | Регистр признаков |
| r8 | SP | | Регистр указателя стека |

Для выполнения логических сдвигов чисел влево основная программы обращается к подпрограмме логического сдвига чисел. Перед обращением число Xi помещается в регистр r1, количество сдвигов Ni – в регистр r2. Подпрограмма возвращает в основную программу результат Zi с помощью регистра r1.

|  |  |  |
| --- | --- | --- |
| m1: | AD R2 00 | Проверка Ni на равенство нулю |
| JZ m3  JC m2 | Если Ni = 0, то переход на метку m3  Если F7=1, то переход на метку m2 |
|  | SHL R1 R1 | Логический сдвиг r1 на один разряд вправо |
|  | JMP m1 | Переход на метку m1 |
| m2: | INC R2 | Увеличение результат на 1 |
|  | JMP m1 | Переход на метку m1 |
| m3: | RET | Возврат из подпрограммы |

1. Кодирование программы и распределение памяти программ и данных
   1. Форматы команд

Команды ЭВМ имеют четыре формата и в зависимости от признака формата (Ф) и кода операции (К) делятся на четыре группы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ф |  | |  | |  |
| 15 | 14..12 | 11..8 | 7..4 | 3..0 |  |
| 0 | К1 | | r | r\* | ADD, SUB, LDI, PUSH, POP, SHR, RET |
| 0 | К2 | | А | | JMP, JZ, HLT |
| 1 | К3 | r | C | | AD, SB |
| 1 | К4 | r | A | | LD, MV, CALL, |

* 1. Коды операций

|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Суммирование | ADD | 0x01 |
| Вычитание | SUB | 0x02 |
| Добавление C | AD | 0x9 |
| Вычитание C | SB | 0xA |
| Чтение в регистр | LD | 0xB |

|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Запись регистра | MV | 0xC |
| Чтение в регистр с индексацией | LDI | 0x0E |
| Запись в стек | PUSH | 0x07 |
| Чтение из стека | POP | 0x06 |
| Переход | JMP | 0x03 |
| Переход, если нуль | JZ | 0x04 |
| Обращение к подпрограмме | CALL | 0xD |
| Возврат из подпрограммы | RET | 0x05 |
| Сдвиг вправо логический | SHR | 0x08 |
| Останов | HLT | 0x00 |

* 1. Распределение памяти

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарии |

Основная программа и данные

|  |  |  |  |
| --- | --- | --- | --- |
| 00 | 0010 | IPS | Адрес начала программы |
| 01 | 0040 | BP | Начальный адрес стека |
| 02 | 0006 | ARRAY | Адрес начала массива |
| 03 | 0005 | N | Кол-во пар элементов в массиве |
| 04 | 0000 | SUM | Результат – сумма сдвигов |
| 05 |  |  | Пустая ячейка |

Массив исходных чисел

|  |  |  |  |
| --- | --- | --- | --- |
| 06 |  | X0 |  |
| 07 |  | X1 |  |
| 08 |  | X2 |  |
| 09 |  | X3 |  |
| 0A |  | X5 |  |

Основная программа

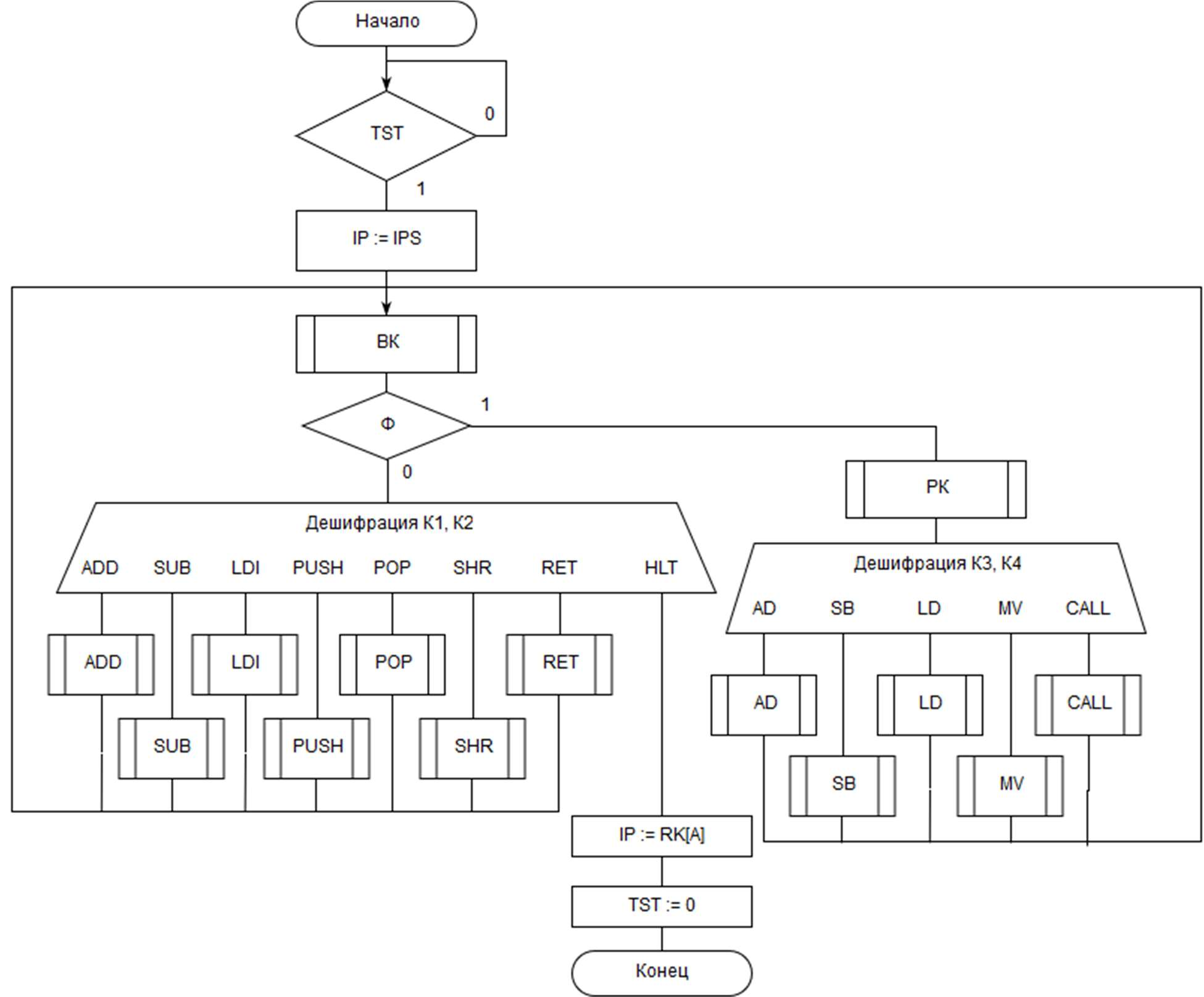
|  |  |  |  |
| --- | --- | --- | --- |
| 10 | 0101 | LD SP BP | Загрузка регистра указателя стека SP |
| 11 | 0202 | LS R5 ARRAY | Загрузка адреса массива ARRAY в r5 |
| 12 | 0303 | LD R4 N | Загрузка числа повторений цикла N |
| 13 | 0405 | LD 10 AD | Загрузка адреса подпрограммы |

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарии |
| 14 | 0515 | LDA A | Загрузка числа |
| 15 | 0600 | JZ A | Проверка на равенство 0 |
| 16 | 0700 | JC A | Проверка разряда |
| 17 | 0800 | INC R1 | Прибавление 1 к результату |
| 18 | 0900 | SHL | Сдвиг |
| 19 | 0A00 | JMP AD | Переход в начало подпрограммы |
| 1A | 0B00 | SUB N | Уменьшение счётчика на 1 |
| 1B | 0C00 | JZ M2 | Если N = 0, то переход на метку m2 |
| 1C | 0D04 | MOV A | Запись результата в 04 |
| 1D | 0010 | HLT | Завершение |

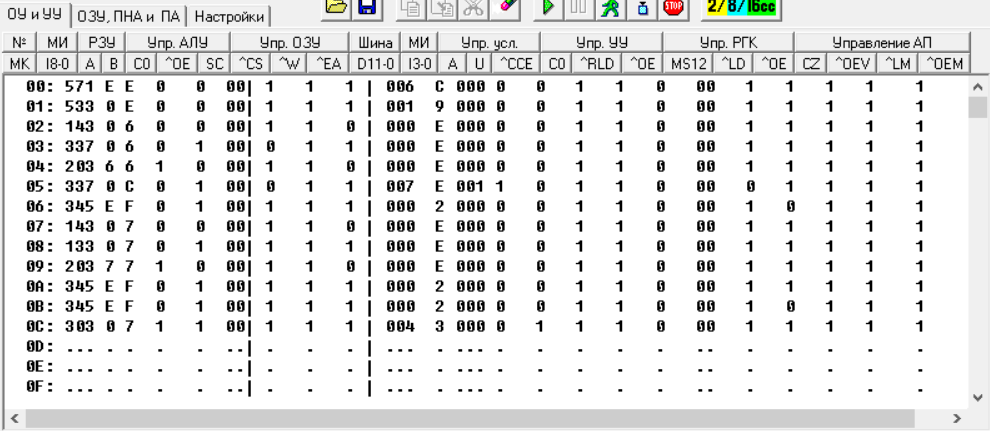
1. Разработка алгоритма работы и микропрограммная реализация ЭВМ
   1. Распределение регистров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | РЗУ (R0 – R7) | | |  | РЗУ (R8 – R15) |
| 0 | r0 | | | 8 | r8 |
| 1 | r1 | | | 9 |  |
| 2 | r2 | | | 10 |  |
| 3 | r3 | | | 11 |  |
| 4 | r4 | | | 12 |  |
| 5 | r5 | | | 13 | Буферный регистр команд |
| 6 | r6 (IP) | | | 14 | Регистр константы |
| 7 | r7 (FLAGS) | | Z | 15 | Счетчик адреса ЗУ RK[A] |
| RA | Адрес ЗУ |  | | RQ |  |

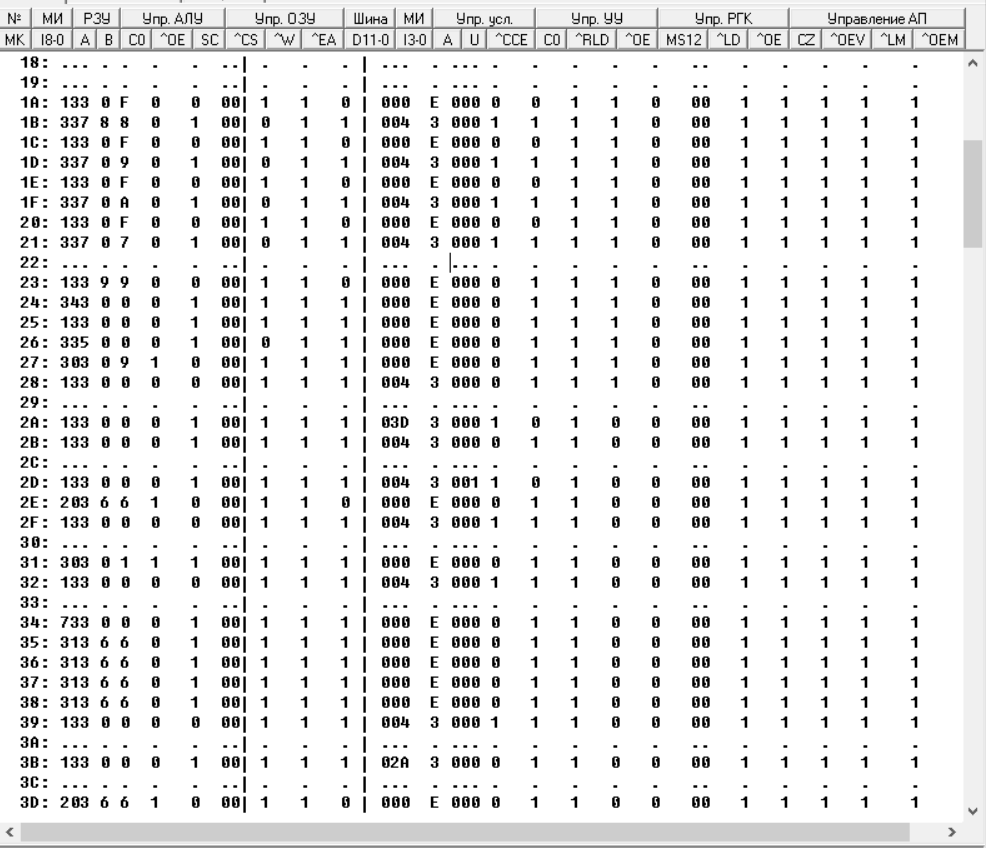
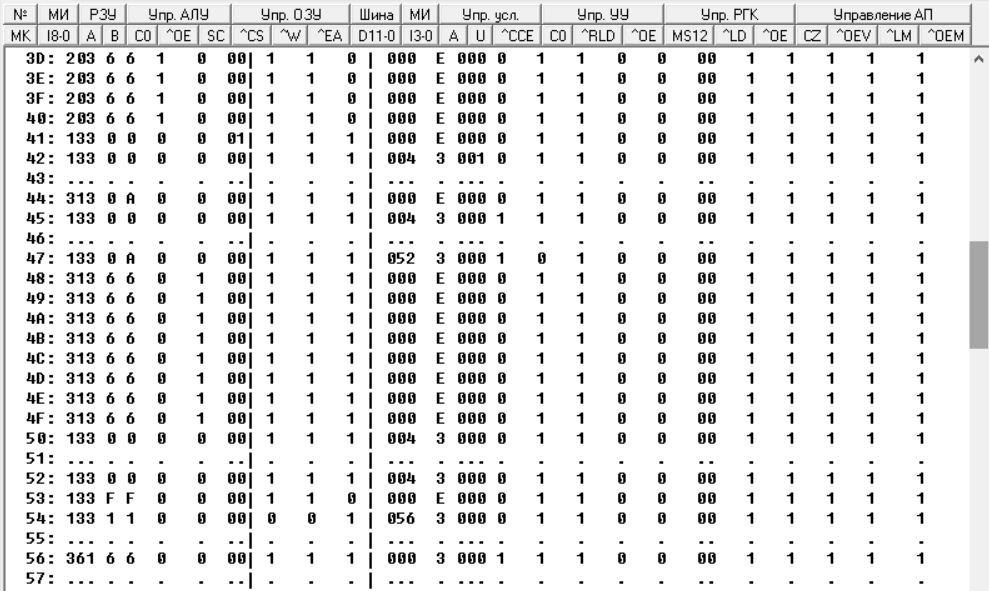
* 1. Граф-схема микропрограммы командного цикла ЭВМ



* 1. Микропрограмма командного цикла (выборка команды и установка признаков)



* 1. Микропрограмма командного цикла (выполнение операций)



1. Вывод

В ходе лабораторной работы была разработана и изучена учебная ЭВМ, разработана и реализована система команд, написана программа решения задачи, которая была помещена в ОЗУ. По сравнению с предыдущей лабораторной работой, система команд была расширена. В дополнение к прямой были добавлены следующие виды адресации; регистровая, неявная регистровая преинкрементная и постдекрементная, непосредственная. Введение различных видов адресации усложнило командный цикл, однако сделало написание программы удобнее и понятнее для программиста.