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 №4 по дисциплине

«Организация ЭВМ и систем» Вариант 7

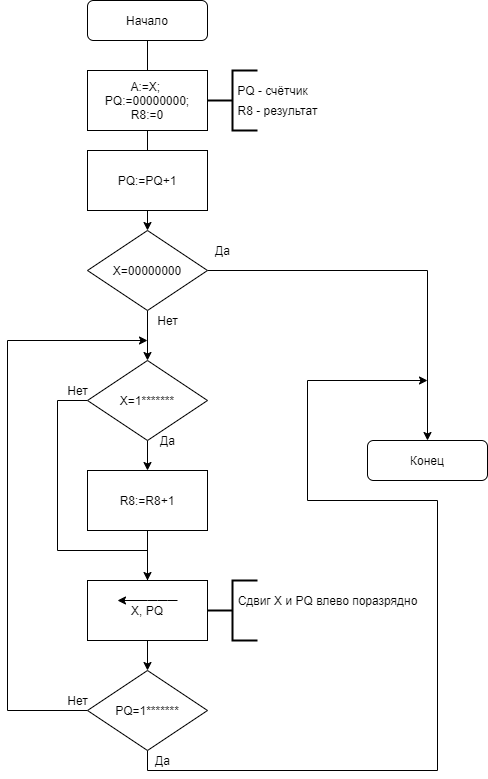
Выполнил студент группы ИВТ-32 /Кудяшев Я.Ю./ Проверил преподаватель /Клюкин В.Л./

Киров 2021

1. Задание

Определить архитектуру ЭВМ, система команд которой состоит из одноадресных команд, использующих прямую адресацию; разработать структурную схему и алгоритм работы ЭВМ; составить и отладить микропрограмму командного цикла ЭВМ; составить и выполнить программу подсчёта количества единиц в числе X, которое вводится пользователем.

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

Программно-доступными регистрами МЭ, система команд которой

состоит из одноадресных команд, можно считать: аккумулятор AX, счетчик CX, программный счетчик PC и регистр признаков RP, содержащий в простейшем случае разряд признака нуля (Z).

* 1. Система команд

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Мнемоника |  | Изменение признака Z |
| Загрузка AX | LDA A | AX := M[A]; PC := PC + 1 | + |
| Запись AX | MOV A | M[A] := AX; PC := PC + 1 | - |
| Сдвиг AX влево | SHL | AX := AX / 2; PC := PC + 1 | + |
| Переход, если ноль | JZ A | Если Z = 1, PC := A, иначе PC := PC + 1 | - |
| Останов | HLT | PC := 0 | - |
| Увеличение AZ на 1 | INC | AZ:=AZ+1 | - |
| Переход если старший разряд 0 | JC A | Если F7=1, PC:=A, иначе PC:=PC+1 | - |
| Безусловный переход | JMP | Переход в A | - |
| Очистка | CLM A | Обнулить AZ | - |

* 1. Программа

1. CLM AZ
2. LDA AX
3. JZ m1
4. SHL

m0: 5 JC m2

1. JMP m0

m2: 7 INC AZ

8 JMP m0

m1: 9 MOV AZ

10 HLT

В программе приняты следующие обозначения:

* + AX — адрес ячейки памяти (ЯП), в которой находится число
  + AZ — адрес ЯП, в которой находится результат

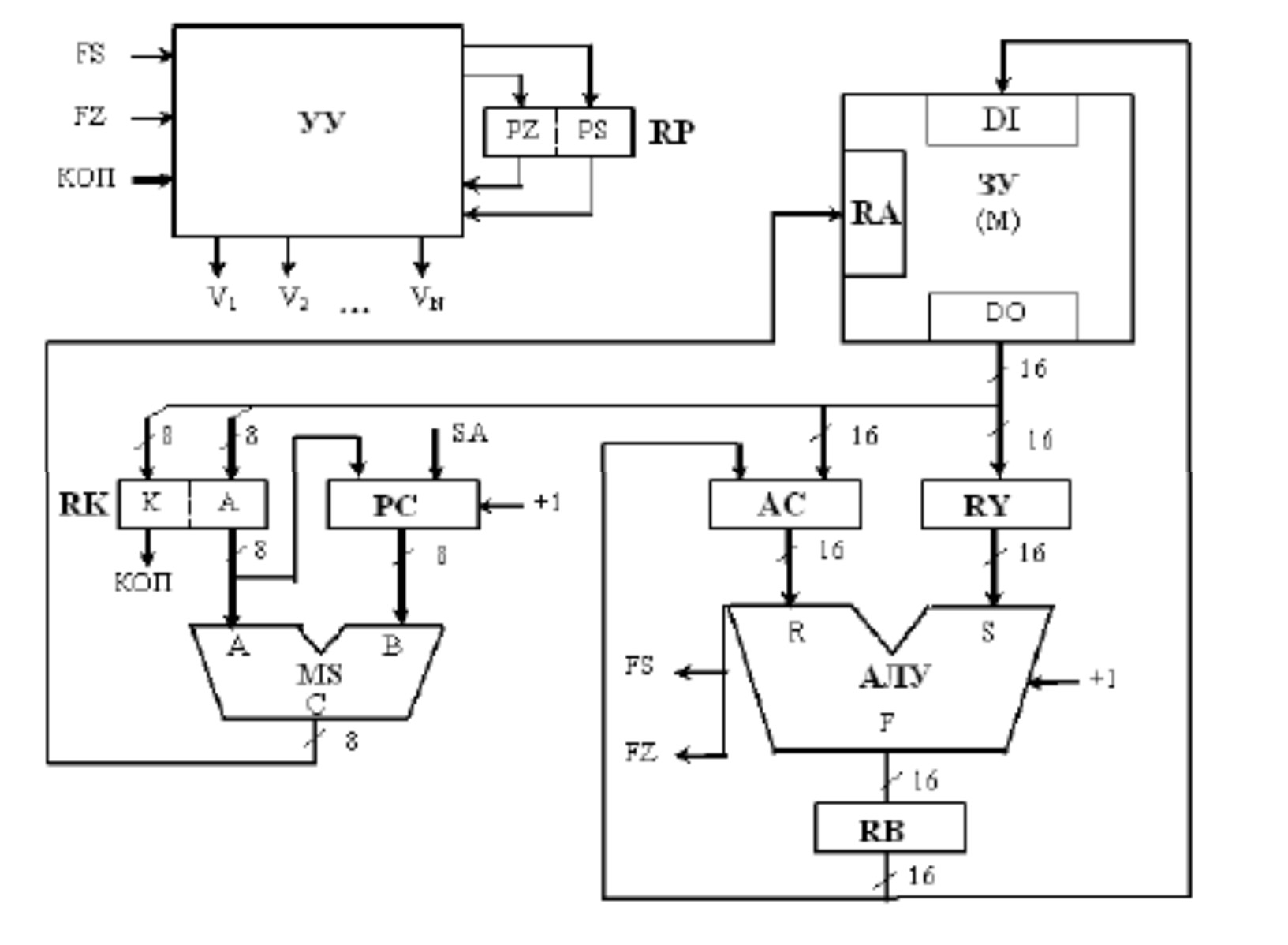
1. Кодирование программы и распределение памяти программ и данных
   1. Коды операций

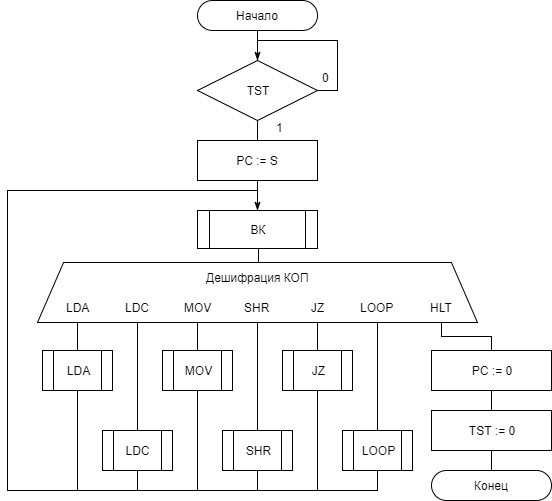
|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Загрузка AX | LDA A | 0x01 |

|  |  |  |
| --- | --- | --- |
| Название | Мнемоника | Код операции |
| Запись AX | MOV A | 0x02 |
| Сдвиг AX влево | SHL | 0x04 |
| Переход, если ноль | JZ A | 0x05 |
| Безусловный переход | JMP A | 0x03 |
| Очистка | CLM A | 0x06 |
| Увеличение на 1 | INC A | 0x07 |
| Останов | HLT | 0x00 |

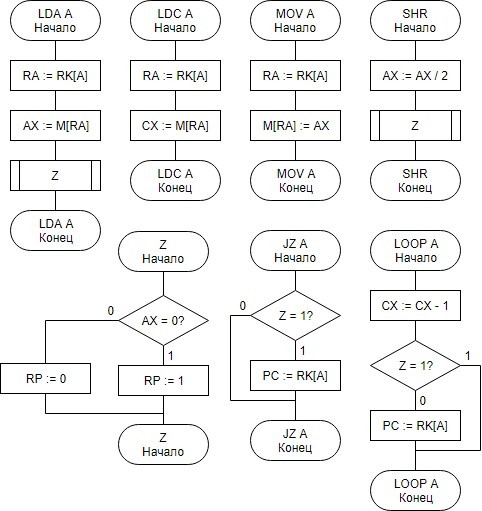
* 1. Распределение памяти программ и данных

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарии |
| 00 | 0006 | S | Адрес начала программы |
| 01 |  | X | Число |
| 02 |  | Z | Результат |
| 03 |  |  | Свободная ячейка |
| 04 |  |  | Свободная ячейка |
| 05 |  |  | Свободная ячейка |
| 06 | 0102 | CLM A | Очистка AZ |
| 07 | 0201 | LDA AX | Загрузка X в AX |
| 08 | 0301 | JZ m1 | Если X=0, то переход |
| 09 | 0402 | JC m2 | Если F7=1, то переход |
| 0A | 0502 | INC AZ | Увеличение AZ на 1 |
| 0B | 0601 | SHL | Сдвиг числа влево |
| 0C | 0704 | JMP 08 | Переход в 08 |
| 0D | 0802 | MOV AX | Записать AX в память |
| 0E | 0006 | HLT | Завершение |

1. Разработка структуры и алгоритма работы
   1. Структура учебной ЭВМ
   2. Алгоритм работы ЭВМ



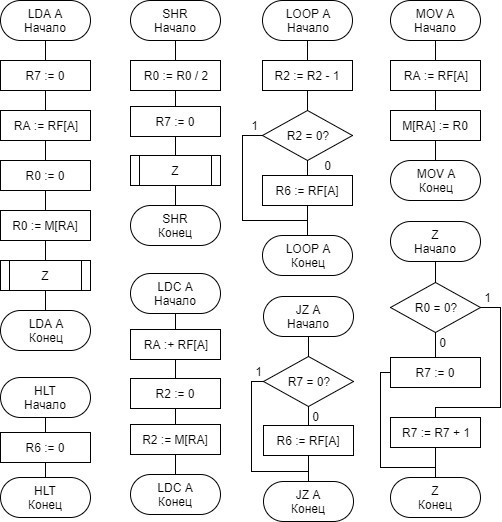
* 1. Алгоритм работы подмикропрограмм



1. Микропрограммная реализация ЭВМ
   1. Распределение регистров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | РЗУ (R0 – R7) | | |  | РЗУ (R8 – R15) |
| 0 | AX | | | 8 |  |
| 1 |  | | | 9 |  |
| 2 | CX | | | 10 |  |
| 3 |  | | | 11 |  |
| 4 |  | | | 12 |  |
| 5 |  | | | 13 | Регистр для данных из ЗУ |
| 6 | PC | | | 14 | Регистр константы |
| 7 | RP | | Z | 15 | Счетчик адреса ЗУ |
| RA | Адрес ЗУ |  | | RQ |  |

* 1. Граф-схемы подмикропрограмм операций



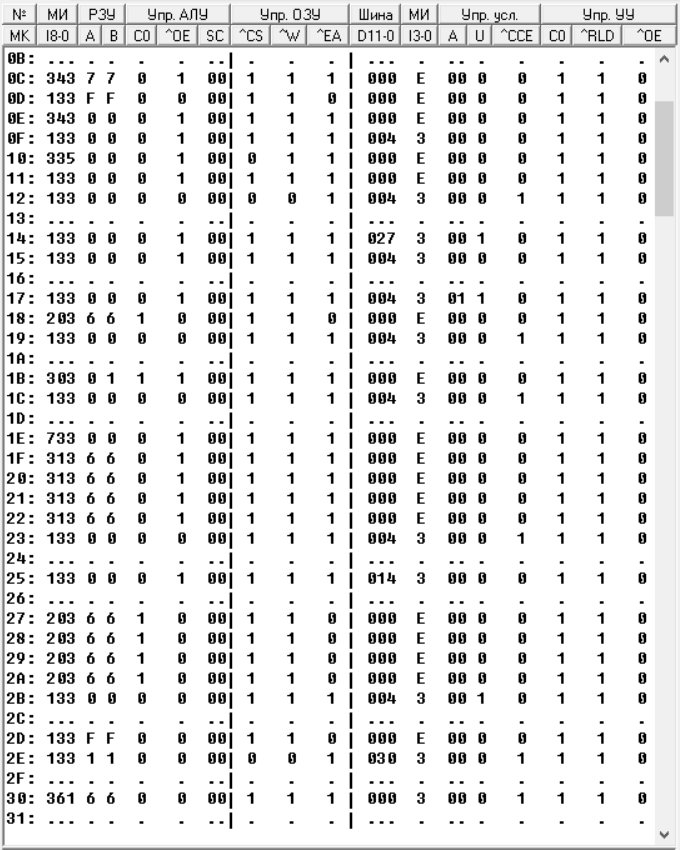
* 1. Коды операций и начальные адреса подмикропрограмм

|  |  |  |
| --- | --- | --- |
| Мнемоника | Код операции | Адрес первой микрокоманды |
| LDA A | 0x01 | 0x0F |
| MOV A | 0x02 | 0x1A |
| JMP A | 0x03 | 0x21 |
| SHL | 0x04 | 0x2A |
| JZ A | 0x05 | 0x2E |
| JC A | 0x07 | 0x32 |
| HLT | 0x00 | 0x1E |

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **МИ** | **РЗУ** | | **Упр. АЛУ** | | | **Упр. ОЗУ** | | | **Шина** | **МИ** | **Упр. усл.** | | | **Упр. УУ** | | |
| **I8-0** | **A** | **B** | **C0** | **^OE** | **SC** | **^CS** | **^W** | **^EA** | **D11-0** | **I3-I0** | **A** | **U** | **^CCE** | **C0** | **^RLD** | **^OE** |
| 00 | 571 | E | E | 0 | 0 | 00 | 1 | 1 | 1 | 006 | C | 00 | 0 | 0 | 1 | 1 | 0 |
| RE:=0111111111111111; РА/СЦ:=6 | | | | | | | | | | | | | | | | | |
| 01 | 533 | 0 | E | 0 | 0 | 00 | 1 | 1 | 1 | 001 | 9 | 00 | 0 | 0 | 1 | 1 | 0 |
| RE – сдвиг вправо; РА/СЦ:=РА/СЦ-1 | | | | | | | | | | | | | | | | | |
| 02 | 143 | 0 | 6 | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RA:=0 | | | | | | | | | | | | | | | | | |
| 03 | 337 | 0 | 6 | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R6:=SA (PC:=S) | | | | | | | | | | | | | | | | | |
| 04 | 203 | 6 | 6 | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RA:=R6; R6:=R6+1 (RA:=PC; PC:=PC+1) | | | | | | | | | | | | | | | | | |
| 05 | 245 | E | F | 0 | 1 | 00 | 0 | 1 | 1 | 000 | 2 | 00 | 0 | 0 | 1 | 1 | 0 |
| Переход по КОП, RF:=K[A] (RK[A]:=K[A]) | | | | | | | | | | | | | | | | | |
| 06 | 303 | 0 | 7 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| Переход к микрокоманде 04 | | | | | | | | | | | | | | | | | |

5.5Микропрограмма командного цикла (выполнение операций)



1. Расчет производительности и быстродействия

Допустим, что среднее число циклов в программе сдвига С = 8, вероятность обнуления операнда до окончания цикла p1 = 0.004, вероятность того, что N = 0, p2 = 0.004; кроме того, будем считать, что время обращения к ЗУ включено во время выполнения микрокоманд.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип  команды | Ср. число  команд, hi | Ср. число  микрокоманд, bi | hi\*bi | Вероятность  команды pi | bi\*pi |
| LDA | 2.992 | 7.5 | 22.44 | 0.097 | 0.73 |
| LDC | 0.996 | 3 | 2.988 | 0.032 | 0.1 |
| MOV | 0.996 | 2 | 1.992 | 0.032 | 0.06 |
| JZ | 8.984 | 1.5 | 13.476 | 0.290 | 0.44 |
| SHR | 7.984 | 5.5 | 43.912 | 0.258 | 1.42 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип  команды | Ср. число  команд, hi | Ср. число  микрокоманд, bi | hi\*bi | Вероятность  команды pi | bi\*pi |
| HLT | 1 | 1 | 1 | 0.032 | 0.032 |
|  | H = 30.936 |  | R = 97.784 |  | r = 3.17 |

Примем 𝜏 = 100 нс. Тогда среднее время выполнения команды составит 𝑇V = 𝑟𝜏 ≈ 317 нс, а быстродействие 𝑉 = 1/𝑇V =

3.15 млн. команд/сек. Аналогично среднее время решения задачи – 𝑇W =

𝑅𝜏 ≈ 9.8 мкс, производительность – 𝑊 = 1/𝑇W ≈ 102 тыс. задач/сек.

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. При выборке данных из ОЗУ старшие 8 бит указывали на код операции, который затем поступал в преобразователь начального адреса – так осуществлялся механизм вызова нужной микропрограммы, а младшие 8 бит указывали на адрес данных. Так была реализована прямая адресация.