МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №4

по дисциплине «Организация ЭВМ»

Выполнил студент группы ИВТ-32 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил старший преподаватель кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Гагарский К. Н./

Киров 2016

1. Задание на лабораторную работу

Необходимо реализовать микропрограмму, осуществляющую логический сдвиг 16 разрядного числа на N разрядов влево.

1. Система команд

Система команд приведена в таблице 1. Граф схема микропрограммы командного цикла представлена на рисунке 1.

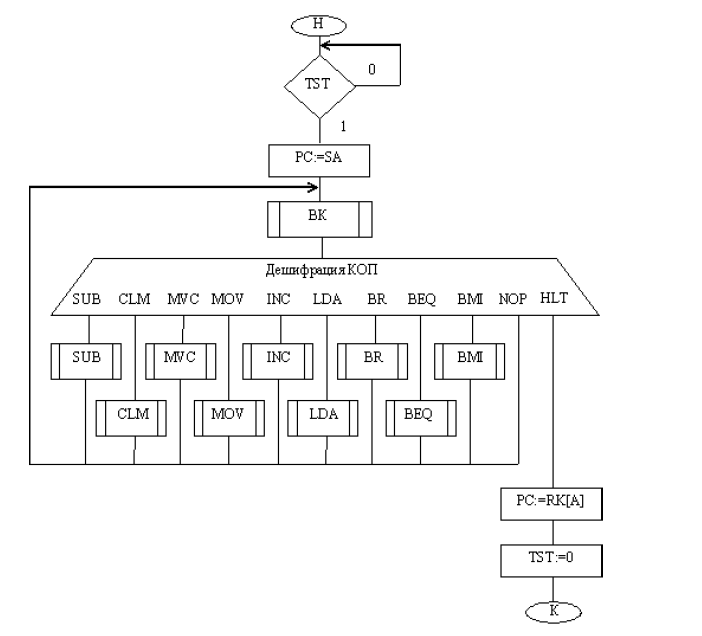


Рисунок 1 – граф схема микропрограммы командного цикла

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Мнемоника | Описание | Признак PZ |
| Очистка | CLM A | M[A] := 00..0; PC := PC + 1 | + |
| Запись AC | MOV A | M[A] := AC; PC := PC + 1 | - |
| Декремент | DEC A | M[A] := M[A] – 1; PC := PC + 1 | - |
| Загрузка AC | LDA A | AC := M[A]; PC := PC + 1 | + |
| Нет операции | NOP | PC := PC + 1 | - |
| Переход | JMP A | PC := A | - |
| Переход если нуль | JZ A | Если PZ = 1, то PC := A  Иначе PC := PC + 1 | - |
| Сдвиг влево AC | SHL AC | AC := AC << 1; PC := PC + 1 | + |
| Останов | HLT A | PC := A; Останов | - |

1. Исходный код программы логического сдвига

Исходный код программы представлен на рисунке 1

|  |  |  |
| --- | --- | --- |
|  | LDA N | Загрузка в AC счетчика сдвига |
|  | JZ m1 | Переход на метку конца, если счетчик N равен 0 |
|  | LDA Number | Загрузка в AC числа, которое необходимо сдвинуть |
|  | SHL AC | Сдвиг числа |
|  | JZ m2 | Если «сдвигаемое» число равно нулю, то переходим на метку 2 |
| m3 | MOV Number | Записываем промежуточные данные сдвига в ОЗУ |
|  | DEC N | Уменьшаем счетчик сдвига на 1 |
|  | LDA N | Загружаем счетчик в AC |
|  | JZ m1 | Если он равен нулю, то переходим к метке конца программы |
|  | JMP m3 | Иначе переходим на m3 |
| m2 | MOV Number | Записываем конечный результат в ОЗУ |
| m1 | HLT SA | Загрузка PC и останов |

Рисунок 3 – Программа логического сдвига

В программе приняты следующие обозначения:

N – количество сдвигов числа

Number – число, которое необходимо сдвинуть

SA – адрес начала программы

1. Кодирование программы и распределение памяти

Кодирование программы приведено в таблице 2.

Таблица 2

|  |  |  |
| --- | --- | --- |
| Наименование | Мнемоника | Код операции |
| Очистка | CLM | 01 |
| Загрузка AC | LDA | 02 |
| Переход, если нуль | JZ | 03 |
| Переход | JMP | 04 |
| Сдвиг влево AC | SHL | 05 |
| Декремент | DEC | 06 |
| Запись AC | MOV | 07 |
| Останов | HLT | 00 |

Распределение памяти программ и данных приведено в таблице 3

Таблица 3

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 00 | 06 | SA | Начальный адрес программы |
| 01 |  | Number | Число, которое необходимо сдвинуть |
| 02 |  | N | Количество сдвигов |
| 03 |  |  |  |
| 04 |  |  |  |
| 05 |  |  |  |
| 06 | 0202 | LDA N | Загрузка счетчика в AC |
| 07 | 0311 | JZ m1 | Переход на метку конца, если счетчик N равен 0 |
| 08 | 0201 | LDA Number | Загрузка в AC числа, которое необходимо сдвинуть |
| 09 | 0500 | SHL AC | Сдвиг числа |
| 0A | 0310 | JZ m2 | Если «сдвигаемое» число равно нулю, то переходим на метку 2 |
| 0B | 0701 | MOV Number | Записываем промежуточные данные сдвига в ОЗУ |
| 0C | 0602 | DEC N | Уменьшаем счетчик сдвига на 1 |
| 0D | 0202 | LDA N | Загружаем счетчик в AC |
| 0E | 0311 | JZ m1 | Если он равен нулю, то переходим к метке конца программы |
| 0F | 0408 | JMP m3 | Иначе переходим на m3 |
| 10 | 0701 | MOV Number | Записываем конечный результат в ОЗУ |
| 11 | 0000 | HLT SA | Загрузка PC и останов |

Распределение внутренних регистров операционного устройства, используемое при выполнении логического сдвига влево, приведено в таблице 4.

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
|  | РЗУ |  | РЗУ |
| 00: | AC | 08: |  |
| 01: |  | 09: |  |
| 02: |  | 10: |  |
| 03: |  | 11: |  |
| 04: |  | 12: |  |
| 05: |  | 13: | Регистр данных из ЗУ |
| 06: | PC | 14: | Регистр константы |
| 07: | RP | 15: | Счетчик адреса ЗУ |
| RA: | Адрес ЗУ | RQ: |  |

1. Граф-схемы подмикропрограмм

Граф схемы подмикропрограмм представлены на рисунке 3



Рисунок 3 – Граф схемы подмикропрограмм

1. Текст отлаженной микропрограммы

Текст отлаженной микропрограммы без конвейерного выполнения микрокоманд представлен в таблице 2.

Таблица 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | МИ | РЗУ | | Упр. АЛУ | | | Упр. ОЗУ | | | Шина | МИ | Упр. Усл. | | | Упр. УУ | | |
| N | I8-0 | A | B | C0 | ^OE | SC | ^CS | ^W | ^EA | D11-0 | I3-0 | A | U | ^CCE | C0 | ^RLD | ^OE |
| 00 | 571 | E | E | 0 | 0 | 00 | 1 | 1 | 1 | 006 | C | 00 | 0 | 0 | 1 | 1 | 0 |
| 01 | 533 | 0 | E | 0 | 0 | 00 | 1 | 1 | 1 | 001 | 9 | 00 | 0 | 0 | 1 | 1 | 0 |
| 02 | 143 | 0 | 6 | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 03 | 337 | 0 | 6 | 0 | 0 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 04 | 203 | 6 | 6 | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 05 | 245 | E | F | 0 | 1 | 00 | 0 | 1 | 1 | 000 | 2 | 00 | 0 | 0 | 1 | 1 | 0 |
| 06 | 133 | 0 | F | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 07 | 143 | 0 | F | 0 | 0 | 00 | 0 | 0 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 08 | 133 | 0 | F | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 1 | 0 | 1 | 1 | 0 |
| 09 | 337 | 0 | 0 | 0 | 1 | 00 | 0 | 1 | 1 | 00E | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 0A | 133 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 0B | 113 | 0 | 7 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 0 | 1 | 1 | 0 |
| 0С | 334 | F | 6 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 0D | 334 | F | 6 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 0E | 343 | 0 | 7 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 0F | 133 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 0 | 1 | 1 | 0 |
| 10 | 303 | 0 | 7 | 1 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 11 | 733 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 12 | 133 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 00E | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 13 | 133 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 004 | 3 | 00 | 0 | 0 | 1 | 1 | 0 |
| 14 | 133 | 0 | F | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 15 | 337 | 0 | D | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 16 | 313 | 0 | D | 0 | 0 | 00 | 0 | 0 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 17 | 133 | 0 | F | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| 18 | 134 | 0 | 0 | 0 | 0 | 00 | 0 | 0 | 1 | 004 | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| 19 | 334 | F | 6 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 1 | 1 | 1 | 0 |

1. Вывод

В ходе выполнения лабораторной работы была построена микропрограмма, которая выполняет логический сдвиг по направлению старших разрядов. Был закреплен материал по теме построения программ для управляющих и операционных устройств. Данный материал является фундаментальным и необходим для продолжения изучения дисциплины «Организация ЭВМ»