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 №5

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

Определить архитектуру, разработать и отладить микропрограмму командного цикла ЭВМ, составить и выполнить программу вычисления суммы сдвинутых в сторону старших разрядов чисел S

1. Система команд

Система команд приведена в таблице 1. Граф схема микропрограммы командного цикла представлена на рисунке 1. Граф схема алгоритма суммирования представлена на рисунке 2.



Рисунок 1 – граф схема микропрограммы командного цикла



Рисунок 2 – Блок-схема программы суммирования

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Мнемоника | Описание | Признак PZ |
| Суммирование | ADD r r\* | r := r + r\*; PC := PC + 1 | + |
| Вычитание с | SB r C | r := r - C; PC := PC + 1 | + |
| Вычитание | SUB r r\* | r := r – r\*; PC := PC + 1 | + |
| Чтение в регистр | LD r A | r := M[A]; PC := PC + 1 | + |
| Запись регистра | MV r A | M[A] := r; PC := PC + 1 | - |
| Переход | JMP A | PC := A | - |
| Переход если нуль | JZ A | Если PZ = 1, то PC := A  Иначе PC := PC + 1 | - |
| Чтение в регистр с индексацией | LDI r (r\*)+ | r := M[r\*]; r\* := r\* + 1;  PC := PC + 1 | + |
| Запись в стек | PUSH r (rSP)- | M[rSP] := r; rSP := rSP – 1;  PC := PC + 1 | - |
| Чтение из стека | POP r (rSP)+ | rSP := rSP + 1; r := M[rSP]  PC := PC + 1 | - |
| Вызов подпрограммы | CALL (rSP)- A | M[rSP] := PC; rSP := rSP – 1;  PC := A | - |
| Сдвиг влево r | SHL r | r := r << 1; PC := PC + 1 | + |
| Останов | HLT A | PC := A; Останов | - |

1. Исходный код программы логического сдвига

Исходный код программы представлен на рисунке 3. Исходный код подпрограммы сдвига числа на n разрядов влево представлен на рисунке 4

|  |  |  |
| --- | --- | --- |
|  | LD R8 AASP | Загрузка регистра указателя стека |
|  | LD R5 AAM | Загружаем адрес массива |
|  | LD R4 AN | Загружаем количество элементов |
|  | SUB R3 R3 | Очищаем регистр R3 |
|  | SUB R2 R2 | Очищаем регистр R2 |
| m2 | LDI R1 (R5)++ | Загружаем i-е число |
|  | LDI R0 (R5)++ | Загружаем i-e количество сдвигов |
|  | JZ m1 | Если количество сдвигов равно нулю, то переход на метрку m1 |
|  | CALL 20 | Вызываем подпрограмму сдвига числа |
| m1 | ADD R3 R1 | Суммируем числа |
|  | SB R4 1 | Вычитаем 1 из счетчика элементов массива |
|  | JZ m3 | Если прошли весь массив, то выходим из цикла (переход на метку m3) |
|  | JMP m2 | Иначе продолжаем (переход на метку m2) |
| m3 | MV R3 AS | Записываем результат суммирования в память |
|  | HLT | Завершаем выполнение программы |

Рисунок 3 – Исходный код программы

|  |  |  |
| --- | --- | --- |
| m2 | SHL R1 | Сдвиг числа |
|  | JZ m1 | Если число равно 0, то переход на метку m1 |
|  | SB R0 1 | Из счетчика сдвигов вычитаем 1 |
|  | JZ m1 | Если счетчик равен 0, то переходим на метку m1 |
|  | JMP m2 | Иначе на m1 |
| m1 | POP PC | Достаем из стека значение программного счетчика |

Рисунок 3 – Программа логического сдвига

Таблица 2. Форматы команд

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

1. Кодирование программы и распределение памяти

Кодирование программы приведено в таблице 3.

Таблица 3. Кодирование программы

|  |  |  |
| --- | --- | --- |
| Наименование | Мнемоника | Код операции |
| СУММИРОВАНИЕ | ADD | 02 |
| ВЫЧИТАНИЕ | SUB | 01 |
| ВЫЧИТАНИЕ КОНСТАНТЫ | SB | C |
| ЧТЕНИЕ В РЕГИСТР | LD | A |
| ЗАПИСЬ РЕГИСТРА | MV | B |
| ЧТЕНИЕ В РЕГИСТР С ИНДЕКСАЦИЕЙ | LDI | 03 |
| ЗАПИСЬ В СТЕК | PUSH | 04 |
| ЧТЕНИЕ ИЗ СТЕКА | POP | 05 |
| ПЕРЕХОД | BR | 10 |
| ПЕРЕХОД, ЕСЛИ НУЛЬ | BEQ | 06 |
| ОБРАЩЕНИЕ К ПОДПРОГРАММЕ | CALL | D |
| СДВИГ В СТОРОНУ СТАРШИХ РАЗРЯДОВ | SHL | 11 |
| ОСТАНОВ | HLT | 00 |

Распределение памяти программ и данных приведено в таблице 4

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 00 | 0006 | SA | Начальный адрес программы |
| 01 | 00FF | ASP | Адрес начала стека |
| 02 | 0016 | AM | Адрес начала массива |
| 03 | 0003 | N | Размерность массива |
| 04 | 0000 | S | Результат суммирования |
| 05 |  |  |  |
| 06 | A801 | LD R8 AASP | Загрузка в регистр R8 адреса стека |
| 07 | A502 | LD R5 AAM | Загрузка в регистр R5 адреса массива |
| 08 | A403 | LD R4 AN | Загрузка в регистр R4 размерность массива |
| 09 | 0133 | SUB R3 R3 | Обнуление регистра R3 |
| 0A | 0122 | SUB R2 R2 | Обнуление регистра R2 |
| 0B | 0315 | LDI R1 | Загрузка i-го числа в регистр R1 |
| 0C | 0305 | LDI R0 | Загрузка i-го числа сдвигов в R0 |
| 0D | 060F | JZ 0F | Проверка количества сдвигов на 0 |
| 0E | D920 | CALL 20 | Вызов подпрограммы сдвига |
| 0F | 0231 | ADD R3 R1 | S = S + R1 |
| 10 | C401 | SB R4 1 | AN = AN – 1 |
| 11 | 0613 | JZ 13 | Проверка окончания массива |
| 12 | 100B | JMP 0B | переход к следующему элементу |
| 13 | B304 | MV S r3 | Запись результата суммы в память |
| 14 | 0007 | HLT | Останов |
| 15 |  |  |  |
| 16 | 0001 | Number[1] |  |
| 17 | 0002 | N[1] |  |
| 18 | 0002 | Number[2] |  |
| 19 | 0002 | N[2] |  |
| 1A | 0001 | Number[3] |  |
| 1B | 0001 | N[3] |  |
| 1C |  |  |  |
| 1D |  |  |  |
| 1E |  |  |  |
| 1F |  |  |  |
| 20 | 1110 | SHL R1 | Сдвиг числа влево на 1 разряд |
| 21 | 0625 | JZ 25 | Проверка его равенства 0 |
| 22 | C001 | SB R0 1 | Уменьшение счетчика сдвигов на 1 |
| 23 | 0625 | JZ 25 | Проверка его на 0 |
| 24 | 1020 | JMP 20 | если не равен 0, то продолжаем сдвиг |
| 25 | 0560 | POP PC | Выход из подпрограммы |

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

Таблица 4

|  |  |  |  |
| --- | --- | --- | --- |
|  | РЗУ |  | РЗУ |
| 00: | N | 08: |  |
| 01: | Number | 09: |  |
| 02: | r2 | 10: |  |
| 03: | S | 11: |  |
| 04: | AN | 12: |  |
| 05: | AM | 13: | Буферный регистр команд |
| 06: | PC | 14: | Регистр константы |
| 07: | RP | 15: | Счетчик адреса ЗУ |
| RA: | Адрес ЗУ | RQ: |  |

Блок схемы операций вызова подпрограммы и выхода из подпрограммы представлены на рисунке 4



Рисунок 4 – Блок схемы вызова подпрограммы и выхода из нее

1. Текст отлаженной микропрограммы

Текст отлаженной микропрограммы конвейерного выполнения микрокоманд представлен в таблице 5.

Таблица 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | 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 |
| 0 | 571 | E | E | 0 | 0 | 0 | 1 | 1 | 1 | 6 | C | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 533 | 0 | E | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 9 | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 | 143 | 0 | 6 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 3 | 337 | 0 | 6 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 203 | 6 | 6 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 5 | 337 | 0 | C | 0 | 1 | 0 | 0 | 1 | 1 | 7 | 3 | 1 | 1 | 0 | 1 | 1 | 0 |
| 6 | 345 | E | F | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 345 | E | F | 0 | 1 | 0 | 1 | 1 | 1 | 3 | C | 0 | 0 | 0 | 1 | 1 | 0 |
| 8 | 533 | 0 | C | 0 | 0 | 0 | 1 | 1 | 1 | 8 | 9 | 0 | 0 | 0 | 1 | 1 | 0 |
| 9 | 131 | C | C | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0A | 343 | 0 | 7 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 0B | 133 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 00E | 3 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0C | 133 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0D | 503 | 0 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0E | 303 | 0 | 7 | 1 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0F | ... | . | . | . | . | .. | . | . | . | ... | . | ... | . | . | . | . | . |
| 10 | 133 | 0 | F | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 11 | 337 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 00A | 3 | 0 | 1 | 1 | 1 | 1 | 0 |
| 12 | 133 | 0 | F | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 13 | 133 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 00A | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 14 | 311 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 00A | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 15 | 301 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 00A | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 16 | 311 | F | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 00A | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 17 | 104 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 18 | 337 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 4 | C | 0 | 0 | 1 | 1 | 1 | 0 |
| 19 | 733 | C | C | 0 | 0 | 0 | 1 | 1 | 1 | 19 | 9 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1A | 303 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | E | 0 | 0 | 0 | 1 | 1 | 0 |
| 1B | 133 | 0 | F | 0 | 0 | 0 | 1 | 1 | 1 | 00A | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1C | 213 | 8 | 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 1 | 1 | 1 | 0 |
| 1D | 133 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1E | 303 | 0 | 8 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 1 | 1 | 1 | 0 |
| 1F | 337 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 20 | 113 | 0 | 7 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 0 | 1 | 1 | 0 |
| 21 | 113 | 0 | 7 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 1 | 1 | 0 | 1 | 1 | 0 |
| 22 | 334 | F | 6 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 23 | 334 | F | 6 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 1 |
| 24 | 213 | 8 | 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | E | 0 | 0 | 1 | 1 | 1 | 0 |
| 25 | 133 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | E | 0 | 0 | 1 | 1 | 1 | 0 |
| 26 | 334 | F | 6 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 3 | 0 | 0 | 1 | 1 | 1 | 0 |
| 27 | 733 | F | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 00A | 3 | 0 | 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»