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 №6

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 программы для следующих исходных данных:

1. Основная программа производит вычисление суммы S элементов Ti массива размерности N:

Σ==NiiTS1,

где T0 и S<32768. ≥

2. В ЭВМ одновременно может поступать до восьми запросов на прерывание программ: Z7, Z8, … Z0. При этом запрос с меньшим номером имеет более высокий приоритет. Запросы на прерывание могут быть замаскированы с помощью 8-разрядной маски M, записываемой в регистр маски RM. Запрос маскируется единичным значением одноименного разряда маски.

3. Каждая программа имеет маску, с помощью которой она может быть защищена от прерываний по тем или иным запросам. Основная программа (P) имеет маску, разрешающую все прерывания. Служебная программа (PS), выполняющая необходимые начальные установки, характеризуется маской запрещающей все прерывания.

4. Программа P0 обработки прерывания нулевого уровня, вызываемая по запросу Z0, сдвигает заданное число X на N разрядов в сторону старших. Остальные программы обработки прерываний (P1-P2) являются

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



Рисунок 3 – Блок-схема программы прерывания P0

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 | - |
| Загрузка маски | LM A | RM := M[A]; PC := PC + 1 | - |
| Возврат из прерывающей программы | IRET | rSP := rSP + 1;  RM := M[rSP];  rSP := rSP + 1; PC := M[rSP] | - |

1. Исходный код программы логического сдвига

Служебная программа представлена на рисунке 4. Исходный код программы представлен на рисунке 5. Исходный код подпрограммы сдвига числа на n разрядов влево представлен на рисунке 6

|  |  |  |
| --- | --- | --- |
|  | LM AMS | Загрузка маски служебной программы |
|  | LD rSP ASP | Загрузка указателя стека SP |
|  | LD rATI ATI | Загрузка адреса таблицы прерывания ATI |
|  | LD PC AP | Загрузка начального адреса программы Р |

Рисунок 4 – Служебная программа начальных установок

|  |  |  |
| --- | --- | --- |
|  | LM AMP | Загрузка маски программы Р |
|  | LD r5 AAM | Загрузка адреса массива АМ в регистр r5 |
|  | LD r4 AN | Загрузка числа повторений цикла N в регистр r4 |
|  | SUB r3 r3 | Очистка регистра r3 для суммы S |
| m2 | LDI r1 (r5)+ | Чтение числа Т[i] в регистр r1 |
|  | ADD r3 r1 | Суммирование |
|  | SB r4 1 | Вычитание единицы из числа повторений цикла |
|  | JZ m1 | Если PZ=1 (r4=0), то переход на метку m1 |
|  | JMP m2 | Переход на метку m2 |
| m1 | MV r3 AS | Запись в память суммы S по адресу AS |
|  | HTL SA | Загрузка PC и останов |

Рисунок 3 – Исходный код программы

|  |  |  |
| --- | --- | --- |
|  | LM AM0 | Загрузка маски программы P0 |
|  | PUSH RP | Сохранение в стеке регистра признаков |
|  | PUSH r0 | Сохранение в стеке регистра r0 |
|  | PUSH r1 | Сохранение в стеке регистра r1 |
|  | MV r0 N | Запись в регистр r0 количества сдвигов |
|  | MV r1 Number | Запись в регистр r1 сдвигаемое число |
| m1 | SHL r1 | Сдвиг в право регистр r1 |
|  | JZ end | Если r1 = 0, то переход на метку end |
|  | SB r0 1 | Вычитаем 1 из счетчика сдвигов |
|  | JZ end | Если счетчик равен 0, то переходим на метку end |
|  | JMP m1 | Иначе на метку m1 |
| end | MV r1 Number | Записываем в память результат сдвига числа |
|  | POP r1 | Загружаем из стека r1 |
|  | POP r0 | Загружаем из стека r0 |
|  | POP RP | Загружаем из стека регистр признаков RP |
|  | IRET | Выходим из прерывающей программы |

Рисунок 3 – Программа логического сдвига

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 |
| ЗАГРУЗКА МАСКИ | LM | 14 |
| ВЫХОД ИЗ ПРЕРЫВАЮЩЕЙ ПРОГРАММЫ | IRET | 15 |
| ОСТАНОВ | HLT | 00 |

Распределение памяти программ и данных приведено в таблице 4

Таблица 4

┌──┬──────────────────┬───────────────────────────────────┐

│ │ │ │

├──┼──────────────────┼───────────────────────────────────┤

│00│ 0005 │SA.................................│

│01│ 00FF │MS.................................│

│02│ 00FF │ASP................................│

│03│ 000A │ATI................................│

│04│ 0023 │AP.................................│

│05│ 1201 │LM AMS.............................│

│06│ A802 │LD RSP ASP.........................│

│07│ A903 │LD RATI ATI........................│

│08│ A604 │LD PC AP...........................│

│09│ 0000 │...................................│

│0A│ 0042 │AP0................................│

│0B│ 0053 │AP1................................│

│0C│ 0056 │AP2................................│

│0D│ 0059 │AP3................................│

│0E│ 005C │AP4................................│

│0F│ 005F │AP5................................│

│10│ 0062 │AP6................................│

│11│ 0065 │AP7................................│

│12│ 0000 │...................................│

│13│ 0000 │M..................................│

│14│ 007F │M0.................................│

│15│ 00BF │M1.................................│

│16│ 00DF │M2.................................│

│17│ 0008 │M3.................................│

│18│ 0010 │M4.................................│

│19│ 0000 │M5.................................│

│1A│ 0000 │M6.................................│

│1B│ 0000 │M7.................................│

│1C│ 0000 │...................................│

│1F│ 0000 │...................................│

│20│ 002F │AM.................................│

│21│ 0003 │N..................................│

│22│ 0006 │S..................................│

│23│ 1213 │LM AMP.............................│

│24│ A520 │LD R5 AAM..........................│

│25│ A421 │LD R4 AN...........................│

│26│ 0133 │SUB R3 R3..........................│

│27│ 0315 │LDI R1 (R5)+.......................│

│28│ 0231 │ADD R3 R1..........................│

│29│ C401 │SB R4 1............................│

│2A│ 062C │JZ 2C..............................│

│2B│ 1027 │JMP 27.............................│

│2C│ B322 │MV R3 AS...........................│

│2D│ 0005 │HLT 5..............................│

│2E│ 0000 │...................................│

│2F│ 0001 │T1.................................│

│30│ 0002 │T2.................................│

│31│ 0003 │T3.................................│

│32│ 0000 │...................................│

│3F│ 0000 │...................................│

│40│ 0004 │NUMBER.............................│

│41│ 0001 │N..................................│

│42│ 1214 │LM AM0.............................│

│43│ 0470 │PUSH RP............................│

│44│ 0400 │PUSH R0............................│

│45│ 0410 │PUSH R1............................│

│46│ A041 │MV R0 N............................│

│47│ A140 │MV R1 NUMBER.......................│

│48│ 1110 │SHL R1.............................│

│49│ 064D │JZ END.............................│

│4A│ C001 │R0 = R0 - 1........................│

│4B│ 064D │JZ END.............................│

│4C│ 1048 │JMP 48.............................│

│4D│ B140 │MV R1 NUMBER.......................│

│4E│ 0510 │POP R1.............................│

│4F│ 0500 │POP R0.............................│

│50│ 0570 │POP RP.............................│

│51│ 1300 │IRET...............................│

│52│ 0000 │...................................│

│53│ 1215 │LM AM1.............................│

│54│ 1300 │IRET...............................│

│55│ 0000 │...................................│

│56│ 1216 │LM AM2.............................│

│57│ 1300 │IRET...............................│

│58│ 0000 │...................................│

│59│ 1217 │LM AM3.............................│

│5A│ 1300 │IRET...............................│

│5B│ 0000 │...................................│

│5C│ 1218 │LM AM4.............................│

│5D│ 1300 │IRET...............................│

│5E│ 0000 │...................................│

│5F│ 1219 │LM AM5.............................│

│60│ 1300 │IRET...............................│

│61│ 0000 │...................................│

│62│ 121A │LM AM6.............................│

│63│ 1300 │IRET...............................│

│64│ 0000 │...................................│

│65│ 121B │LM AM7.............................│

│66│ 1300 │IRET...............................│

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 5



Рисунок 5 – Граф-схемы перехода на прерывающую программу, операции возврата из прерывающей программы и операции загрузки маски

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»