МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Вычислительное устройство

Отчет по лабораторной работе №3 дисциплины

«Организация ЭВМ и систем»

Выполнил студент группы ИВТб-21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Киров 2019

1. Задание на лабораторную работу

Подсчитать число одинаковых цифр, расположенных в одноименных разрядах xi и yi 8-разрядных двоичных чисел X и Y.

1. Распределение ячеек ЗУ и регистров микропроцессора

Распределение ячеек ЗУ показано в таблице 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 00 | 0100 | КОП | Код операции |
| 01 |  | X | Первое число |
| 02 |  | Y | Второе число |
| 03 |  | P | Счетчик совпадений |

Таблица 1 – Распределение ячеек ЗУ

Распределение регистров микропроцессора приведено в таблице 2. В отличие от распределения регистров, использованного при исследовании ОУ, в МПР дополнительно выделяется регистр счётчика адреса ЗУ (R15).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | РЗУ (R0-R7) |  |  | РЗУ (R8-R15) |
| 0: | Регистр X |  | 8: | Регистр P |
| 1: | Регистр n-1 |  | 9: |  |
| 2: |  |  | 10: |  |
| 3: |  |  | 11: |  |
| 4: |  |  | 12: |  |
| 5: |  |  | 13: |  |
| 6: |  |  | 14: |  |
| 7: |  |  | 15: | Счетчик адреса ЗУ |
|  |  |  |  |  |
| RA: | Адрес ЗУ |  | RQ | Регистр Y |

Таблица 2 – Распределение регистров

1. Разработка микропрограммы для устройства без конвейерного выполнения микрокоманд

При разработке микропрограммы предполагается, что она состоит из двух частей (микропрограмм). В первой микропрограмме производится выборка операндов из ЗУ, дешифрация кода операции и запуск второй микропрограммы, а также запись результатов операции. Вторая микропрограмма обеспечивает выполнение операции. Эта микропрограмма работает с данными, уже находящимися в регистрах операционного устройства.

Разработка микропрограммы для устройства без конвейерного выполнения микрокоманд включает набор кода операции (подсчет одинаковых цифр в разрядах) и распределение ячеек преобразователя начального адреса, разработку граф – схем для первой и для второй частей микропрограммы, а также кодирование микропрограммы и составление таблицы прошивок блок памяти микропрограммы.

Для операции подсчета одинаковых цифр в разрядах принят код операции, равный 01. При этом вторая часть микропрограммы, выполняющая подсчет, размещена в блоке памяти микропрограмм начиная с ячейки, адрес которой равен 0A. Распределение ячеек преобразователя начального адреса, преобразующего код операции (01) в соответствующий начальный адрес микропрограммы операции (0A) представлено в таблице 3.

|  |  |  |
| --- | --- | --- |
| КОП (адрес) | Начальный адрес  микропрограммы | Комментарий |
| 00 | … | … |
| 01 | 00001010 | 0A – адрес микропрограммы подсчета |
| 02 | … | … |

Таблица 3 – Распределение ячеек преобразователя начального адреса

Граф – схема микропрограммы для устройства без конвейерного выполнения микрокоманд приведена на рисунке 1, где M[RA] – ячейка ЗУ, адрес которой указан в регистре адреса RA.

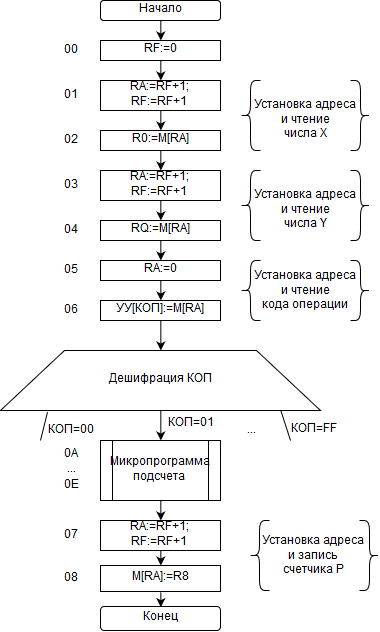


Рисунок 1 – Граф – схема микропрограммы выполнения операции в ВУ

В соответствии с принятым размещением данных в ЗУ (табл. 1), из ячейки памяти с адресом 01 в регистр операционного устройства R0 считывает число X (МК 01, 02), далее в ЗУ в регистр RQ считывается число Y (МК 03, 04). Затем в регистр адреса ЗУ записывается нуль, и производится чтение из ЗУ кода операции (МК 05, 06). Считываемый код операции поступает в преобразователь начального адреса УУ и обеспечивает переход на первую микрокоманду микропрограммы операции подсчета одинаковых чисел в разрядах. После выполнения микропрограммы подсчета осуществляется запись результата в ЗУ: счетчика P (МК 07, 08).

Граф – схема микропрограммы подсчета приведена на рисунке 2. В соответствии с принятым алгоритмом выполнения операции, перед началом цикла деления обнуляется регистр счетчика R8 (МК 0А). Затем с помощью МК 0B формируется число с 1 там, где расположены равные разряды. С помощью команд 0C, 0D убирается одна из 1 в регистре R0. В микрокоманде 0E происходит увеличение регистра R8 на 1 единицу, и происходит переход к команде начала цикла подсчета 0С.

Размещение кода микропрограммы в блоке памяти микропрограмм для устройства без конвейерного выполнения микрокоманд показано в таблице 4.

Микрокоманда подсчета одинаковых цифр размещается в памяти, начиная с ячейки 0A и заканчивая ячейкой 0E. Микрокоманды этой микропрограммы содержат операционные части соответствующих микрокоманд микропрограммы для ОУ, а управляющие части – микрокоманд микропрограммы для УУ.

Основная микропрограмма, в которой происходит выпорка операндов из ЗУ, дешифрация кода операции и запуск микропрограммы деления, а также запись результатов операции, занимает в блоке памяти микропрограмм диапазон ячеек с 00 по 08.

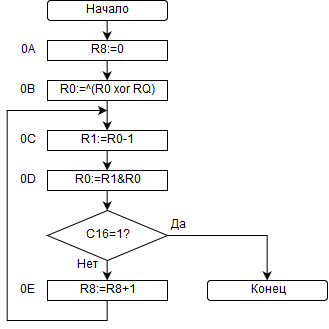


Рисунок 2 – Граф – схема подсчета одинаковых цифр двух чисел

(без конвейерного выполнения микрокоманд)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | МИ | РЗУ | | Упр. АЛУ | | | Упр. ОЗУ | | | Шина | МИ | Упр. усл. | | | Упр. УУ | | |
| N | I8-0 | A | B | C0 |  | SC |  |  |  | D11-0 | I3-0 | A | U |  | C0 |  |  |
| 00 | 343 | 0 | F | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=0 | | | | | | | | | | | | | | | | | |
| 01 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 02 | 337 | 0 | 0 | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R0:=X | | | | | | | | | | | | | | | | | |
| 03 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 04 | 037 | 0 | 0 | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RQ:=Y | | | | | | | | | | | | | | | | | |
| 05 | 143 | 0 | E | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RA:=0 | | | | | | | | | | | | | | | | | |
| 06 | 237 | 0 | E | 0 | 1 | 00 | 0 | 1 | 1 | 000 | 2 | 00 | 0 | 0 | 1 | 1 | 0 |
| Переход по КОП | | | | | | | | | | | | | | | | | |
| 07 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 08 | 133 | 0 | 8 | 0 | 0 | 00 | 0 | 0 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| Запись P | | | | | | | | | | | | | | | | | |
| 09 | … | . | . | . | . | .. | . | . | . | … | . | .. | . | . | . | . | . |
| 0A | 343 | 0 | 8 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R8:=0 | | | | | | | | | | | | | | | | | |
| 0B | 370 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R0:=^(R0 xor RQ) | | | | | | | | | | | | | | | | | |
| 0C | 314 | 0 | 1 | 0 | 1 | 00 | 1 | 1 | 1 | 007 | 3 | 11 | 1 | 0 | 1 | 1 | 0 |
| R1:=R0-1 | | | | | | | | | | | | | | | | | |
| 0D | 341 | 1 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R0:=R0&R1 | | | | | | | | | | | | | | | | | |
| 0E | 303 | 0 | 8 | 1 | 1 | 00 | 1 | 1 | 1 | 00C | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| R8:=R8+1 | | | | | | | | | | | | | | | | | |
| 0F | … | . | . | . | . | .. | . | . | . | … | . | .. | . | . | . | . | . |

Таблица 4 – Микропрограмма выполнения операции в ВУ без конвейерного выполнения микрокоманд

1. Разработка микропрограммы для устройства с конвейерным выполнением микрокоманд

Совмещение во времени выполнения микроопераций текущей микрокоманды и считывания следующей не позволяют использовать для формирования адреса следующей микрокоманды значения логических условий, определяемых в текущей микрокоманды. Для сохранения конвейерной обработки микрокоманд перехода необходимо сформировать и запомнить значения логических условий до выполнения микрокоманды, содержащей условный переход. Выполнение этого условия требует преобразования микропрограмм, составленных для ВУ без конвейерного выполнения микрокоманд.

Преобразование заключается в перенесении микроопераций формирования значений логических условий из микрокоманды условного перехода в предшествующую ей микрокоманду и включении в операционную часть микрокоманды перехода микроопераций, которые могут быть выполнены до перехода, но после формирования значений условий. Если такая перестановка микроопераций не допускается алгоритмом обработки, то микрокоманда перехода заменяется двумя микрокомандами: микрокомандой, формирующей значения логических условий, и “пустой” микрокомандой, которая обеспечивает переход по условию, но не выполняет преобразований, предписываемых алгоритмом обработки.

Размещение кода микропрограммы в блоке памяти микропрограмм для устройства с конвейерным выполнением микрокоманд показано в таблице 5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | МИ | РЗУ | | Упр. АЛУ | | | Упр. ОЗУ | | | Шина | МИ | Упр. усл. | | | Упр. УУ | | |
| N | I8-0 | A | B | C0 |  | SC |  |  |  | D11-0 | I3-0 | A | U |  | C0 |  |  |
| 00 | 343 | 0 | F | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=0 | | | | | | | | | | | | | | | | | |
| 01 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 02 | 337 | 0 | 0 | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R0:=X | | | | | | | | | | | | | | | | | |
| 03 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 04 | 037 | 0 | 0 | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RQ:=Y | | | | | | | | | | | | | | | | | |
| 05 | 143 | 0 | E | 0 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RA:=0 | | | | | | | | | | | | | | | | | |
| 06 | 237 | 0 | E | 0 | 1 | 00 | 0 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| Переход по КОП | | | | | | | | | | | | | | | | | |
| 07 | 133 | 0 | E | 0 | 1 | 00 | 1 | 1 | 1 | 000 | 2 | 00 | 0 | 0 | 1 | 1 | 0 |
| Переход по КОП | | | | | | | | | | | | | | | | | |
| 08 | 303 | 0 | F | 1 | 0 | 00 | 1 | 1 | 0 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| RF:=RF+1; RA:=RF+1 | | | | | | | | | | | | | | | | | |
| 09 | 133 | 0 | 8 | 0 | 0 | 00 | 0 | 0 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| Запись P | | | | | | | | | | | | | | | | | |
| 0A | 343 | 0 | 8 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R8:=0 | | | | | | | | | | | | | | | | | |
| 0B | 133 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| BRS:=F(R0) | | | | | | | | | | | | | | | | | |
| 0C | 370 | 0 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R0:=^(R0 xor RQ) | | | | | | | | | | | | | | | | | |
| 0D | 314 | 0 | 1 | 0 | 1 | 00 | 1 | 1 | 1 | 000 | E | 00 | 0 | 0 | 1 | 1 | 0 |
| R1:=R0-1 | | | | | | | | | | | | | | | | | |
| 0E | 341 | 1 | 0 | 0 | 1 | 00 | 1 | 1 | 1 | 008 | 3 | 11 | 1 | 0 | 1 | 1 | 0 |
| R0:=R0&R1 | | | | | | | | | | | | | | | | | |
| 0F | 303 | 0 | 8 | 1 | 1 | 00 | 1 | 1 | 1 | 00D | 3 | 00 | 0 | 1 | 1 | 1 | 0 |
| R8:=R8+1 | | | | | | | | | | | | | | | | | |
| 10 | … | . | . | . | . | .. | . | . | . | … | . | .. | . | . | . | . | . |

Таблица 5 – Микропрограмма выполнения операции в ВУ с конвейерным выполнением микрокоманд

1. Сравнение микропрограмм

Допустим, что в рассматриваемом примере среднее число циклов подсчета равных чисел равно 4, а вероятность возникновения единицы переноса – 0,01. Тогда среднее число тактов, необходимых для выполнения микропрограмм, равны: в МПР без конвейерного выполнения M1≈18, а в МПР с конвейерным выполнением микрокоманд M2≈20.

Для расчета времени выполнения микропрограмм примем τ1=τ, а τ1=kτ. Примем k-0,7, тогда T1=M1\*τ=18τ и T2=M2\*k\*τ=14τ.

Объём памяти, необходимой для размещения микропрограмм, равен: для МПР без конвейерного выполнения микрокоманд: N1=14, n1=47 и V1=14\*47=658 бит. Для МПР с конвейерным выполнением микрокоманд: N2=15, n2=47 и V2=15\*47=705 бит.

Таким образом, конвейерная обработка микрокоманд позволила уменьшить время вычислений и привела к незначительному увеличению объёма памяти, необходимого для размещения микропрограммы.