МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Череповецкий государственный университет»

**Лабораторная работа № 1**

**«Разработка алгоритма и микропрограммы**

**арифметической операции»**

**Выполнил:**

**студент гр.** 1ИВТпб-01-31оп

Климов А.Г.  
**Проверил: преподаватель**

Виноградова Л.Н.  
**Отметка о зачете:**

Череповец

2017 год

**Формулировка варианта задания**

Эти задания не требуют реализации циклов в микропрограммах, достаточно использовать лишь переходы, в зависимости от принятого способа кодирования операндов и сочетания их знаков. В табл. 10.1 приведены варианты заданий. Их особенность состоит в том, что часто операнды и результат представляются в различных кодах. Конечно, в реальных АЛУ такого разнобоя в кодировке быть не может (это значительно усложняет алгоритм при отсутствии преимуществ реализации). Однако в учебных заданиях такой подход можно считать приемлемым, поскольку это, во-первых, увеличивает разнообразие вариантов и, во-вторых, позволяет читателю в рамках одного задания вспомнить особенности выполнения операции в различных кодах. В табл. 10.1 приняты следующие обозначения:

ПК — прямой код;

ОК — обратный код;

ДК — дополнительный код.

В задании указываются операция, код выполнения операции, коды входных операндов и код результата. Если код выполнения операции прямой, необходимо в процессе ее выполнения перевести оба операнда в прямой код и осуществить заданную операцию с использованием микроопераций суммирования или вычитания двоичных векторов. Если код выполнения операции дополнительный (обратный), необходимо в процессе выполнения операции перевести оба операнда в дополнительный (обратный) код. В случае выполнения вычитания далее необходимо определить дополнительный (обратный) код противоположного по знаку числа, осуществить операцию с использованием микрооперации суммирования двоичных векторов и, возможно, провести коррекцию результата. После выполнения операции результат необходимо перевести в код, указанный в задании как код результата. Таблица 1. Вариант задания

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Тип операции** | **Код 1 операнда** | **Код 2 операнда** | **Код выполнения** | **Код результата** |
| 5 | Вычитание | ДК | ПК | ДК | ПК |

Таблица 1. Вариант задания

**Разработанный алгоритм заданной операции в форме ГСА**

Разработанный алгоритм заданной операции в форме ГСА изображён на рис. 1.

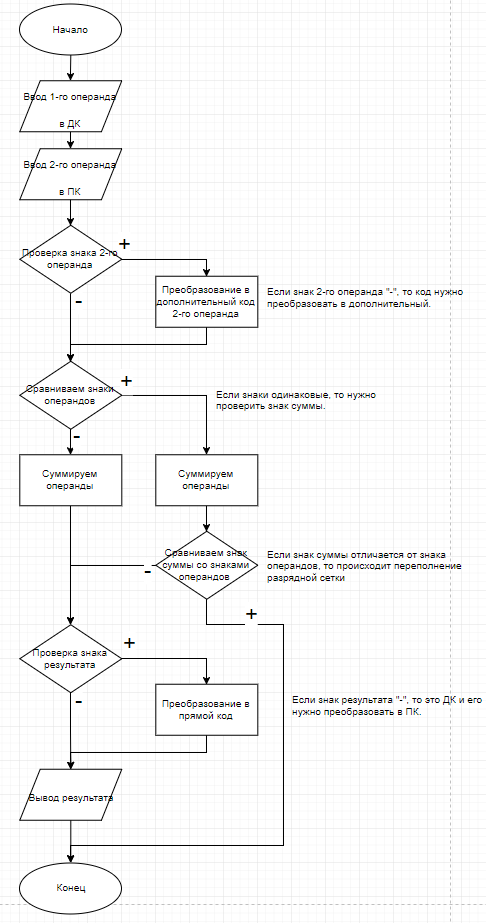


Рис. 1. Разработанный алгоритм заданной операции в форме ГСА

Микропрограмма, соответствующая разработанному алгоритму и структуре операционного автомата, на котором предполагается ее реализовать (ALU-1) представлена в табличной форме табл. 2.

Таблица 2. Микропрограмма

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Метка** | **Микрокоманда** | **Комментарий** |
| 1 |  | y0, y2 | Ввод операнда A |
| 2 |  | y1, y3 | Ввод операнда B |
| 3 |  | y6, y13, y18, y25 | Формируем вектор неравнозначности Q = A  ⊕ B и сдвигаем влево. Т.к. микрокоманда подключения на вход R отсутствует, то подаются все нули, при этом значение неравнозначности знаковых разрядов попадает в DL, а значит x2 = a7 ⊕ b7 |
| 4 |  | [if x2 then L2] | Проверяем знак операнда, переход на метку L2, если знак операнда “-” |
| 5 | L1: | y14, y28 | Сброс флага p0 и установка триггера DL в 0 |
| 6 |  | y4, y6, y13, y18 | Формируем вектор неравнозначности Q = A  ⊕ B и сдвигаем влево, при этом значение  неравнозначности знаковых разрядов попадает в DL, а значит x2 = a7 ⊕ b7 |
| 7 |  | [if x2 then L3] | Проверяем равенство знаков операндов, переход на метку L3, если знаки операндов разные |
| 8 |  | y4, y6, y8, y16, y24 | Сложение, сумма попадает в C |
| 9 |  | y26, y2 | Записывает результат суммы в регистр A |
| 10 |  | y4, y6, y13, y18 | Формируем вектор неравнозначности Q = A  ⊕ B и сдвигаем влево, при этом значение  неравнозначности знаковых разрядов суммы и операнда попадает в DL, а значит x2 = a7 ⊕ b7 |
| 11 |  | [if x2 then END] | Если знак суммы и операндов совпадают, то x2=0 и программа выполняется дальше, иначе происходит переполнение. |
| 12 |  | [go to L5] | Переход на метку L5 |
| 13 | L3: | y4, y6, y8, y16, y24 | Сложение, сумма попадает в C |
| 14 |  | y26, y2 | Записывает результат суммы в регистр A |
| 15 | L5: | y4, y13, y18 | Формируем вектор неравнозначности Q = A  ⊕ B и сдвигаем влево. Т.к. микрокоманда подключения на вход S отсутствует, то подаются все нули, при этом значение неравнозначности знаковых разрядов попадает в DL, а значит x2 = a7 ⊕ b7 |
| 16 |  | [if x2 then L4] | Проверяем знак операнда, переход на метку L4, если знак операнда “-” |
| 17 |  | [go to END] | Завершение операции |
| 18 | L2: | y27, y3 | Возвращение результата сдвига влево в регистр B |
| 19 |  | y15 | Установка триггера p0 в 1 |
| 20 |  | y7, y8, y17, y25 | Перевод кода из регистра B в ДК и запись в D |
| 21 |  | y27,y3 | Запись в регистр B значений из регистра D |
| 22 |  | [go to L1] | Переход на метку L1 |
| 23 | L4: | y15 | Установка триггера p0 в 1 |
| 24 |  | y5, y8, y18, y24 | Инвертируем число и сдвигаем влево |
| 25 |  | y29 | Установка DL в 1 |
| 26 |  | y26, y2 | Запись сдвинутого кода в регистр A |
| 27 |  | y4, y8, y17, y24 | Сдвиг преобразованного кода вправо и запись итогового результата в регистр C |
| 28 | END: | y21 | Завершение операции |

Результат тестирования микропрограммы на нескольких примерах. В табл. 3. происходит сложение двух положительных чисел A и B. A = 310 = 000000112 и B = 710 = 000001112.

Таблица 3. Результат тестирования, когда А и В положительные

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ МК** | **Микрокоманда** | **Ax (ДК)** | **Bx (ПК)** | **A** | **B** | **C** | **D** | **DL**  **(x2)** |
|  |  | 00000011 | 00000111 |  |  |  |  |  |
| 1 | y0, y2 |  |  | 00000011 |  |  |  |  |
| 2 | y1, y3 |  |  |  | 00000111 |  |  |  |
| 3 | y6, y13, y18, y25 |  |  |  |  |  | 00001110 | 0 |
| 4 | Анализируем x2 |  |  |  |  |  |  |  |
| 5 | y14, y28 |  |  |  |  |  |  | 0 |
| 6 | y4, y6, y13, y18 |  |  |  |  |  |  | 0 |
| 7 | Анализируем x2 |  |  |  |  |  |  |  |
| 8 | y4, y6, y8, y16, y24 |  |  |  |  | 00001010 |  |  |
| 9 | y26, y2 |  |  | 00001010 |  |  |  |  |
| 10 | y4, y6, y13, y18 |  |  |  |  |  |  | 0 |
| 11 | Анализируем x2 |  |  |  |  |  |  |  |
| 12 | Переход на L5 |  |  |  |  |  |  |  |
| 15 | y4, y13, y18 |  |  |  |  | 00010100 |  | 0 |
| 16 | Анализируем x2 |  |  |  |  |  |  |  |
| 18 | Переход на END |  |  |  |  |  |  |  |
| 28 | y21 |  |  |  |  |  |  |  |

В табл. 4. происходит сложение, когда А положительное, а B отрицательное и по абсолютной величине больше, чем А.A = 310 = 000000112 и B = -1010 = 100010102.

Таблица 4. Результат тестирования, когда А положительное, B отрицательное и по абсолютной величине больше, чем А

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ МК** | **Микрокоманда** | **Ax (ДК)** | **Bx (ПК)** | **A** | **B** | **C** | **D** | **DL**  **(x2)** |
|  |  | 00000011 | 10001010 |  |  |  |  |  |
| 1 | y0, y2 |  |  | 00000011 |  |  |  |  |
| 2 | y1, y3 |  |  |  | 10001010 |  |  |  |
| 3 | y6, y13, y18, y25 |  |  |  |  |  | 00001010 | 1 |
| 4 | Анализируем x2 |  |  |  |  |  |  |  |
| 18 | y27, y3 |  |  |  | 00010100 |  |  |  |
| 19 | y15 |  |  |  |  |  |  |  |
| 20 | y7, y8, y17, y25 |  |  |  |  |  | 11110110 |  |
| 21 | y27,y3 |  |  |  | 11110110 |  |  |  |
| 22 | Переход к L1 |  |  |  |  |  |  |  |
| 5 | y14, y28 |  |  |  |  |  |  | 0 |
| 6 | y4, y6, y13, y18 |  |  |  |  |  |  | 1 |
| 7 | Анализируем x2 |  |  |  |  |  |  |  |
| 13 | y4, y6, y8, y16, y24 |  |  |  |  | 11111001 |  |  |
| 14 | y26, y2 |  |  | 11111001 |  |  |  |  |
| 15 | y4, y13, y18, y24 |  |  |  |  |  |  | 1 |
| 16 | Анализируем x2 |  |  |  |  |  |  |  |
| 23 | y15 |  |  |  |  |  |  |  |
| 24 | y5, y8, y18, y24 |  |  |  |  | 00001110 |  |  |
| 25 | y29 |  |  |  |  |  |  |  |
| 26 | y26, y2 |  |  | 00001110 |  |  |  |  |
| 27 | y4, y8, y17, y24 |  |  |  |  | 10000111 |  |  |
| 28 | y21 |  |  |  |  |  |  |  |

В табл. 5. А и В положительные, сумма А+В больше, либо равна 2n–1.A = 6510 = 010000012 и B = 9710 = 011000012.

Таблица 5. Результат тестирования, когда А и В положительные, сумма А+В больше, либо равна 2n–1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ МК** | **Микрокоманда** | **Ax (ДК)** | **Bx (ПК)** | **A** | **B** | **C** | **D** | **DL**  **(x2)** |
|  |  | 01000001 | 01100001 |  |  |  |  |  |
| 1 | y0, y2 |  |  | 01000001 |  |  |  |  |
| 2 | y1, y3 |  |  |  | 01100001 |  |  |  |
| 3 | y6, y13, y18, y25 |  |  |  |  |  | 00001110 | 0 |
| 4 | Анализируем x2 |  |  |  |  |  |  |  |
| 5 | y14, y28 |  |  |  |  |  |  | 0 |
| 6 | y4, y6, y13, y18 |  |  |  |  |  |  | 0 |
| 7 | Анализируем x2 |  |  |  |  |  |  |  |
| 8 | y4, y6, y8, y16, y24 |  |  |  |  | 10100010 |  |  |
| 9 | y26, y2 |  |  | 10100010 |  |  |  |  |
| 10 | y4, y6, y13, y18 |  |  |  |  |  |  | 1 |
| 11 | Анализируем x2 |  |  |  |  |  |  |  |
| 12 | Переход на END |  |  |  |  |  |  |  |
| 28 | y21 |  |  |  |  |  |  |  |

**Теория**

**Действия в ОА, необходимые для реализации алгоритма:**

* сравнение знаков операндов;
* сложение или вычитание (прямое или обратное) операндов;
* анализ знака результата;
* присваивание результату знака одного из операндов;
* установка признака переполнения.

**Алгоритм:**

* Ввод 1-го операнда;
* Ввод 2-го операнда;
* Преобразование 2-го операнда из ПК в ДК;
* Определение дополнительного кода противоположного по знаку числа;
* Сравнение знаков операндов;
* Дополнительное вычитание операндов;
* Присваивание результату знака одного из операндов;
* Установка признака переполнения;
* Перевод результата в ПК.

**Пояснения к алгоритму:**

Преобразование из ПК в ДК:

Проверяем знак (если знак “+”, то ДК совпадает с ПК, иначе знак “-”, следовательно, получаем обратный код и добавляем к младшему разряду единицу).

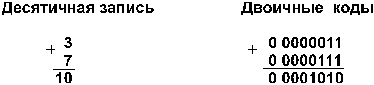
Дополнительный код положительного числа совпадает с прямым кодом. Для отрицательного числа дополнительный код образуется путем получения обратного кода и добавлением к младшему разряду единицы.

Неравнозначность получается поразрядным сложением по модулю 2, т.е. без учёта переноса в старший разряд.

**Сложение дополнительных кодов 6 случаев:**

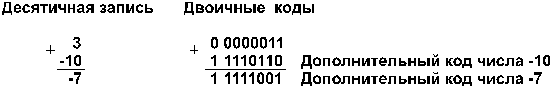
1. А и В положительные.

При суммировании складываются все разряды, включая разряд знака. Так как знаковые разряды положительных слагаемых равны нулю, разряд знака суммы тоже равен нулю. Например:



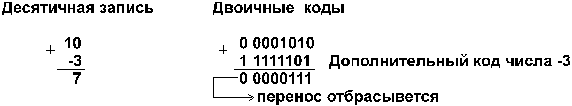
Верно!

2. А положительное, B отрицательное и по абсолютной величине больше, чем А.  Например:



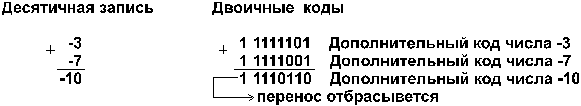
Верно!

3. А положительное, B отрицательное и по абсолютной величине меньше, чем А. Например:



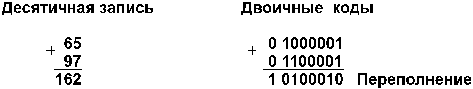
Верно!

4. А и В отрицательные. Например:



Верно!

5. А и В положительные, сумма А+В больше, либо равна 2n–1, где n — количество разрядов формата чисел (для однобайтового формата n=8, 2n–1= 27 = 128). Например:



Верно!

6. А и В отрицательные, сумма абсолютных величин А и В больше, либо равна 2n–1. Например:



Здесь знак суммы тоже не совпадает со знаками слагаемых, что свидетельствует о переполнении разрядной сетки.

Верно!