Лабораторная работа №2

**Моделирование работы операционного автомата, выполняющего операцию умножение чисел в дополнительном и обратном коде.**

**Цель работы:** изучить структуру операционного автомата, на основе данной структуры создать модель операционного автомата, выполняющего операцию умножение, выполняемое методом накопления частичных произведений чисел, представленных в дополнительном и обратном коде.

**Задача:** По представленным блок-схемам алгоритмов написать программу, протестировать на языке высокого уровня.

1. **Теоретическая часть**

1.1 Умножение, выполняемое методом накопления частичных произ-ведений. Операция умножения в современных ЭВМ чаще всего выполняется суммированием сдвинутых на один или несколько разрядов частичных произведении, каждое из которых является результатом умножения множимого на соответствующий разряд (разряды) множителя.

При точном умножении двух чисел количество значащих цифр произведения может в пределе достичь двойного количества значащих цифр сомножителей. Еще сложнее возникает ситуация при умножении нескольких чисел. Поэтому в произведении только в отдельных слу­чаях используют двойное количество разрядов, обычно же ограничи­ваются количеством разрядов, которое имели сомножители. Здесь учитывается то обстоятельство, что правила приближенных вычислений рекомендуют оставлять в произведении столько же значащих цифр, сколько их содержится в наименее точном из сомножителей. Младшие разряды результата при этом отбрасываются, а старшие обычно округляются по известным правилам с тем, чтобы ошибка произведения стала знакопеременной и ее математическое ожидание было равно 0 с учетом равновероятности любых значений отброшенных младших разрядов.

Наиболее просто операция умножения в ЭВМ выполняется в пря­мом коде. При этом на первом этапе определяется знак произведения путем сложения знаковых цифр сомножителей по модулю 2.

Произведение вычисляется как сумма частичных произведений, из которых каждое получается последовательными сдвигами и умножением множимого на соответствующий разряд множителя. Произведение двух n- разрядных чисел является 2n- разрядным числом.

Перемножение модулей сомножителей производится по пра­вилам арифметики согласно двоичной таблице умножения. Результа­ту присваивается полученный знак.

Так как умножение производится в двоичной системе счисления, частные произведения либо равны 0 (при умножении на 0), либо са­мому сомножителю (при умножении на 1), сдвинутому на соответству­ющее количество разрядов.

ПримерТаблица.1

|  |  |
| --- | --- |
| х 0.1011  0.1101  1011  0000  1011  1011  0.10001111 | Заданы: А=0.1011, В=0.1101 Получить произведение *А • В* путем умножения с младших разрядов множителя *В.* |
| 0.1011  0.1101  1011  + 1011  0000  1011  0.10001111 | Получить произведение путем умножения со старших разрядов множителя |

Процессом накопления суммы частных произведений можно уп­равлять с помощью цифр множителя в соответствии с выражением

(1)

где С — искомое произведение; *А —* множимое; *В —* множитель.

Управление процессом умножения может начинаться как с младших разрядов множителя, так и со старших.

При этом полную сумму (произведение) можно получить двумя путями:

сдвигом множимого на требуемое количество разрядов и добавлением полученного очередного частного произведения ранее накопленной сумме;

сдвигом суммы ранее полученных частных произведений на каждом шаге на один разряд и последующим добавлением к сдвинутой сумме неподвижного множимого либо 0.

Основываясь на выше изложенном можно создать четыре варианта схем машинного умножения:

1) умножение младшими разрядами множителя со сдвигом накапливаемой суммы вправо;

2) умножение младшими разрядами множителя со сдвигом множимого влево;

3) умножение старшими разрядами множителя со сдвигом суммы частичных произведений влево;

4) умножение старшими разрядами множителя со сдвигом множимого вправо.

Рассмотрим более детально каждую из четырех схем умножения.

1. Умножение младшими разрядами множителя со сдвигом суммы частичных произведений вправо.

Выражение (1) можно представить в виде схемы Горнера для вычисления полиномов.

С=(((... ((0 + А +Abn) 2-1+Abn-1) 2-1 + ... + Abn-i) 2-i+ ... Ab2)2-1+Ab1)2-1, (2)

которое может быть сведено к n-кратному выполнению цикла

Сi+1 = (Сi + Abn-i). 2-1 (3)

при начальных значениях i=0; Со = 0.

В каждом цикле множимое либо добавляется к сумме частичных произведений (если bi=1), либо нет (если *bi=0*), после чего сумма частичных произведений умножается на 2-1, т. е. сдвигается на один разряд вправо. После окончания *п-го* цикла образуется искомое произведение, т. е. *Сn= С = AB.*

Очередную цифру множителя, управляющую суммированием час­тичных произведений, удобнее всего снимать с младшего разряда регистра множителя, в котором в каждом цикле производится сдвиг содержимого на один разряд вправо.

Реализация данного способа требует п-разрядного сдвигающего регистра множителя, *п* схем И, пропускающих множимое на вход сумматора при bi=1 и запрещающих его передачу при bi=0, n-разрядного регистра множимого 2n-разрядного сумматора, в котором накапливается сумма и имеются цепи для ее сдвига вправо на один разряд.

1. Умножение младшими разрядами множителя со сдвигом множимого влево. Представим выражение (1) в виде

 (4)

Вычисление выражения (2) сводится к п-кратному выполнению цикла:

 (5)

где *Аi=*2Ai-1, при начальных значениях i=0; C0=0; An=A.

В каждом цикле умножения множимое сдвигается на один разряд влево и либо передается в СМ (при bi=1), либо нет (при bi= 0).

Для реализации данного способа умножения требуется п-разрядный сдвигающий регистр множителя, 2n-разрядный сдвигающий регистр множимого и 2n-разрядный сумматор.

3. Умножение старшими разрядами множителя со сдвигом суммы частичных произведений влево. Если преобразовать выражение (1) к виду

 (6)

то умножение двух чисел *А* и *В* сводится к п-кратному повторению цикла

 (7)

с начальными значениями i *=* 0; Со = 0. Тогда управление умножением будет производиться цифрами множителя, начиная со старших разрядов. Сумма частичных произведений в каждом цикле будет сдвигаться на один разряд влево.

Таким образом, для реализации данной схемы необходимы: сдвигающий влево регистр множителя с *п* разрядами, *n*-разрядный регистр множимого и 2n-разрядный сумматор со сдвигом влево.

4. Умножение старшими разрядами множителя со сдвигом множимого вправо. Если представить (1) в виде

 (8)

то вычисление произведения может быть сведено к n-кратному выполнению цикла:

 (9)

при начальных условиях i=0; *А0 = А; Со* = 0.

Таким образом, в каждом цикле множимое сдвигается на один разряд вправо и в зависимости от значения управляющего (старшего) разряда множителя либо передается в СМ, либо нет.

С учетом этого для реализации данной схемы умножения необходим n-разрядный регистр множителя со сдвигом влево, 2n-разрядный регистр множимого со сдвигом вправо и 2n-разрядный сумматор.

Если потребовать, чтобы ошибки произведения не превышали единицы младшего разряда, то можно значительно сократить количество дополнительных разрядов регистра множимого и сумматора.

При k дополнительных разрядах погрешность произведения обусловлена двумя причинами:

1. В каждом цикле передачи множимого в СМ после k-го будут пропадать его младшие разряды. Тогда при условии, что все n-k младших разрядов множимого равны 1, максимальная погрешность произведения составит



1. После окончания умножения отбрасываются все k младших разрядов

СМ. Если все они равны 1, погрешность от этого составит

 (11)

Если потребовать, чтобы  было меньше 2-n , то из неравенства

 (12)

можно определить количество дополнительных разрядов /г:

 (13)

Обе составляющие ошибки имеют отрицательный знак. Округление первой составляющей ошибки можно выполнять путем прибавле­ния 1 к (n+k+1)-му разряду СМ каждый раз, начиная после (k+1)-й передачи множимого в СМ, или всякий раз прибавлять 1 к (k+n+1)-му разряду СМ, когда она выходит за пределы разряд­ной сетки регистра множимого при сдвиге содержимого последнего вправо. Округление второй составляющей можно выполнить путем добавления 1 к *(п*+1)-му разряду СМ после окончания умножения.

1.2 Умножение чисел, заданных в дополнительном коде

Операцию умножения проще всего выполнять в прямых кодах чисел. Вместе с тем применение инверсных кодов позволяет существен­но упростить операцию алгебраического сложения. Поэтому числа желательно хранить в ЗУ и умножать также в инверсном коде. В этом случае сомножители заданы в прямом коде, если они положи­тельны, либо в инверсном коде, если они отрицательны. Необходимо получить произведение в прямом коде, если оно положительно, или в инверсном коде, если оно отрицательно. При этом, с целью устране­ния циклических переносов, рациональнее использовать дополни­тельный код. При умножении в дополнительном коде, так же, как и при алгебраическом сложении, требуется введение поправок в пред­варительный результат. Эти поправки вносятся исходя из следующих предпосылок.

Если число *А* отрицательно, то значащие цифры его дополнитель­ного кода образуют величину | Aд | = 2n — | *А* |, число 2n означает двойную разрядную сетку, которую необходимо отвести под произведение. Поэтому при перемно­жении модулей кодов операндов в зависимости от сочетания знаков сомножителей могут возникнуть 4 случая.

1. *А* > 0, *В* > 0. Случай тривиальный, получаем сразу истинное значение положительного произведения:

Z = (|A| пр \* |B| пр ) (14)

1. *А* > 0, *В* < 0. Получаем после непосредственного применения алгоритма умножения:

(|A| пр \* |B| пр ) = |A| дп \* (2n - |B|дп) = 2n |A|пр - ( |A| пр \* |B|дп) (15)

## Истинный результат в дополнительном коде составит:

|Aпр | \* |Bдп | = Кор – (|A| пр \* |B| дп), (16)

где Кор – коррекция произведения.

##### Следовательно, требуется коррекция псевдоре­зультата в старшей части произведения на величину Кор = 2n|А |пр, т. е. на величину дополнения множителя до единицы.

1. *А* < 0, *В* > 0. Получаем после непосредственного применения алгоритма умножения:

(|A| пр \* |B| пр ) = (2n -|A| дп) \* |B|пр = 2n |B|пр - ( |A| дп \* |B|пр) (17)

## Истинный результат в дополнительном коде составит

|Aдп | \* |Bпр | = Кор – (|A| дп \* |B| пр), (18)

где Кор – коррекция произведения.

##### Следовательно, требуется коррекция псевдоре­зультата на величину

Кор = 2n - |B|пр, т. е. на величину дополнения множителя до единицы.

1. *А* < 0, *В* < 0. После непосредственного применения алгорит­ма умножения получаем:

|Aпр| \* |Bпр| = (2n - |A|дп) \* (2n - |B| дп) = 22n – (2n|A| дп ) - (2n|B|дп )

+ (|A| дп \* |B| дп) = (|Aпр| \* |Bпр|) - 2n(|A| дп + |B| дп ) (19)

Псевдорезультат требует коррекции на величину Кор = 2n(|A| дп + |B| дп ), так как лишняя единица размещается в знаковом разряде и может быть просто заменена знаком произведения.

Как видим, сложность коррекции результата при умножении чи­сел в дополнительных кодах обусловлена тем, что в исправлении нуждается не только знак, но и цифровая часть произведения. Коррек­цию можно производить или в процессе формирования результата, или сразу по окончании этого процесса. Первый способ применяют чаще, так как он не требует дополнительных тактов работы АУ, сни­жающих скорость счета и усложняющих схему управления умноже­нием. Кроме того, при сдвигах, как правило, теряются один или оба сомножителя, а повторное обращение за ними в ЗУ нежелательно, так как приводит к снижению быстродействия ЭВМ.

1.3 Умножение чисел, заданных в обратном коде

Если число *А* отрицательно, то значащие цифры его обратного кода образуют величину | Aоб | = 2n| *А* | - | *А* |. Поэтому при перемно­жении модулей кодов операндов в зависимости от сочетания знаков сомножителей могут возникнуть 4 случая.

1. *А* > 0, *В* > 0. В этом случае получаем сразу истинное значение положительного произведения:

Z = (|A| пр \* |B| пр ) (20)

1. *А* > 0, *В* < 0. Получаем после непосредственного применения алгоритма умножения:

(|В| пр = (2n - |В| об) – 1 (21)

Z = (|A| пр \* |B| пр ) = |A| дп \* [(2n - |B|об) - 1] = 2n|A|пр - ( |A| пр \* |B|об) - |A| пр (22)

## Истинный результат в обратном коде составит

|Aпр | \* |Bдп | = Кор – (|A| пр \* |B| об), (23)

где Кор – коррекция произведения.

##### Коррекция псевдоре­зультата равна: Кор = 2n|А |пр - |А |пр.

На основании полученной формулы (22) возможны три варианта получения результата:

1) Z = [2n|A|пр - ( |A| пр \* |B|об)] - |A| пр , (24)

2) Z = [2n|A|пр - |A| пр] - ( |A| пр \* |B|об), (25)

3) Z = 2n|A|пр – [( |A| пр \* |B|об) + |A| пр]. (26)

1. *А* < 0, *В* > 0. Получаем после непосредственного применения алгоритма умножения:

(|A| пр = (2n - |A| об) – 1 (27)

Z = (|A| пр \* |B| пр ) = [(2n - |A| об) - 1] \* |B|пр = 2n |B|пр - ( |А| об \* |В|пр) - |В| дп (28)

## Истинный результат в дополнительном коде составит

|Aдп | \* |Bпр | = Кор – (|A|об \* |B| пр), (29)

где Кор – коррекция произведения. Следовательно, требуется коррекция псевдоре­зультата на величину Кор = 2n|B|пр - |B|пр.

На основании полученной формулы (28) возможны три варианта получения результата:

1) Z = [2n|В|пр - ( |A| об \* |В|пр)] - |В| пр , (30)

2) Z = [2n|В|пр - |В|пр] - ( |A|об \* |B|пр), (31)

3) Z = 2n|В|пр – [(|А|об \* |В| пр) + |В| пр]. (32)

1. *А* < 0, *В* < 0. После непосредственного применения алгорит­ма умножения получаем:

(|A| пр = (2n - |A| об) – 1; (|В| пр = (2n - |В| об) – 1 (33)

Z = |Aпр| \* |Bпр| = [(2n - |A|об) – 1] \* [(2n - |B| об) –1] = 22n – (2n|A| об ) - (2n|B|об ) +

+ (|A|об \* |B|об) + |A| об + |B|об + 1 = (|Aоб| \* |Bоб| ) - 2n(|A|об + |B|об) +

+ (|A|об + |B|об) (34)

# Псевдорезультат требует коррекции на величину

Кор = (|A|об + |B|об) - 2n(|A|об + |B|об ) (35)

На основании полученной формулы (34) - вычисления произведения возможны три варианта получения результата:

1. Z = [(|Aоб| \* |Bоб| ) - 2n(|A|об + |B|об)] + (|A|об + |B|об), (36)
2. Z = [(|Aоб| \* |Bоб| ) + (|A|об + |B|об)] - 2n(|A|об + |B|об), (37)
3. Z = [(|A|об + |B|об) - 2n(|A|об + |B|об)] + (|Aоб| \* |Bоб| ). (38)

Алгоритм умножение чисел, представленных в дополнительном коде

*1*

н

*2*

А, В

*3*

*4*

*5*

**0**

**0**

**0**

П1

П3

П2

**1**

**1**

**1**

*6*

*8*

*7*

С:= |Адоп| \* |Вдоп|

С:= |Адоп| \* |Впр|

С:= |Апр| \* |Вдоп|

*18*

*9*

*10*

*11*

Z:= |Апр| \* |Впр|

К:= 2n |Апр|

К:=2n(|Адоп|+|Вдоп|)

*19*

*14*

*13*

*12*

Z:= С - |Кдоп|

Зр.Z:=Зр.А

*15*

*16*

*17*

Z:= С + |Кдоп|

К:= 2n |Впр|

Z:= K-|Cпр|

Z:= K + |Cдоп|

Z:= K - |Cпр|

Z:= K + |Cдоп|

*20*

Зр.Z:=Зр.А ⊕ Зр.В

*21*

Зр Z, |Z|

*22*

**к**

Рис.1.

На рис.1 изображена блок-схема алгоритма сложения чисел, представленных в дополнительных кодах, где обозначено: А, В – операнды; С – промежуточная переменная; З. р Z – знаковый разряд; К – коррекция; Z – окончательный результат; Хпр – двоичное значение операнда Х; Хдоп.  - значение операнда Х в дополнительном коде; П1 – А>0, B<0; П2 – А<0, B>0; П3 – A<0, B<0.

**Работа алгоритма умножения чисел в дополнительном коде.**

Блок 1 алгоритма является начальным (рис.1).

В блоке 2 алгоритма происходит ввод чисел А и В в десятичной системе счисления. В этом блоке числа переводятся в двоичную систему счисления соответствующими процедурами с учетом знаков операндов. Числа А и В будут представлены в формате прямых кодов.

В блоке 3 алгоритма анализируется признак П1 – при котором операция умножения выполняется когда число В имеет единицу в знаковом разряде, т.е. число отрицательное, А > 0, В < 0 .

В блоке 4 алгоритма анализируется признак П2 – при котором операция умножения выполняется когда число А имеет единицу в знаковом разряде, т.е. число отрицательное А< 0, В > 0 .

В блоке 5 алгоритма анализируется признак П3 – при котором операция умножения выполняется когда числа А и В имеют единицы в знаковых разрядах, т.е. оба числа отрицательные, А< 0, В < 0 .

В блоке 6 алгоритма вычисляется произведение чисел А и В, где число А представлена в прямом коде, а число В в дополнительном по команде:

С : = |Апр| \* |Вдоп|.

В блоке 7 алгоритма вычисляется произведение чисел А и В, где число А представлена в дополнительном коде, а число В в прямом по команде:

С : = |Адоп| \* |Впр|.

В блоке 8 алгоритма вычисляется произведение чисел А и В, где числа А и В представлены в дополнительных кодах по команде: С : = |Адоп| \* |Вдоп|.

В блоке 9 алгоритма вычисляется коррекция произведения - К по формуле:

К : = 2n |Апр|.

В блоке 10 алгоритма вычисляется коррекция произведения - К по формуле: К : = 2n |Впр|.

В блоке 11 алгоритма вычисляется коррекция произведения - К по формуле: К : = 2n (|Адоп| + |Вдоп|).

В блоках 12 и 13 алгоритма вычисляется результат Z в прямом коде по формуле: Z : = K - |Cпр|.

В блоке 14 алгоритма вычисляется результат Z в прямом коде по форму-ле: Z : = С - |Кдоп|.

В блоках 15 и 16 алгоритма вычисляется результат Z в дополнительном коде по формуле: Z : = K + |Cдоп|.

В блоке 17 алгоритма результат Z вычисляется в дополнительном коде по формуле: Z : = С + |Кдоп|.

В блоках 18 и 19 алгоритма по формуле: Z : = |Апр| \* |Впр| вычисляется произведение операндов, числа А и В в этом случае положительные. Результата Z будет представлен в прямом коде. Знаковому разряду произведения Зр.Z присваивается знак одного из чисел по команде:

Зр.Z : =Зр.А.

В блоке 20 алгоритма по операции суммы по модулю два определяется знаковый разряд результата Z по команде:

Зр.Z : =Зр.А ⊕ Зр.В.

В блоке 21 алгоритма выдается знаковый разряд произведения Зр.Z и результата по модулю |Z|.

Блок 22 алгоритма является конечным блоком.

Примеры: умножение чисел в дополнительном коде.

1. А = 0.101 (5) х 101 Апр

В= 1.111 (-7) 001 Вдп

Z = |Апр| \* |Вдоп| ) = |A| дп \* (2n - |B|дп) = 000 101 С дп

= 2n |A|пр - ( |A| пр \* |B|дп) ↓дп

+ 111 011 Спр 101 000 2nАпр

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 0 ⊕ 1 = 1

Окончательный результат Z 2 = 1. 100 011

Z10 = -35

100 011 |Z| = (35)10

1 не учитывается

1. А = 1.110 (-6) х 010 Адп

В = 0.100 (4) 100 Впр

Z = (|A| пр \* |B| пр ) = (2n -|A| дп) \* |B|пр = 001 000 Сдп

= 2n |B|пр - ( |A| дп \* |B|пр) ↓дп

+ 111 000 Спр

100 000 Впр

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 1 ⊕ 0 = 1

Окончательный результат Z 2 = 1. 011 000

Z10 = -24

011 000 |Z|= (24)10

1. не учитывается

3) А = 1.111 (-7) х 001 Адп

В = 1.110 (-6) 010 Вдп

Z = |Aпр| \* |Bпр| = (2n - |A|дп) \* (2n - |B| дп) = 000 010 Сдп

= 22n – (2n|A| дп ) - (2n|B|дп ) + (|A| дп \* |B| дп) =

= (|Aпр| \* |Bпр|) - 2n(|A| дп + |B| дп ) + 001 Адп

010 Вдп

011 К(ор)дп

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 1 ⊕ 1 = 0

Окончательный результат Z 2 = 0. 101 010

Z10 = 42

101 (Кор)пр

+ 000 010 Сдп

101 000 К(ор)пр в ст. ч

101 010 |Z| = (42)10

Алгоритм умножение чисел, представленных в обратном коде

*1*

н

*2*

А, В

*3*

*4*

*5*

**0**

**0**

**0**

П1

П3

П2

**1**

**1**

**1**

С:= |Апр| \* |Впр|

*18*

*6*

*8*

*7*

С:= |Аобр| \* |Вобр|

С:= |Аобр| \* |Впр|

С:= |Апр| \* |Вобр|

*9*

*10*

*11*

К:= 2n(|Аобр|+|Вобр|) -

(|Аобр|+|Вобр|)

К:= 2n |Впр| - |Впр|

К:= 2n |Апр| - |Апр|

*19*

*14*

*13*

*12*

Зр.Z:=ЗрВ

Z:= С - |Кпр|

*15*

*16*

*17*

Z:= С + |Кобр|

Z:= K- |Cпр|

Z:= K + |Cобр|

Z:= K - |Cпр|

Z:= K + |Cобр|

*20*

Зр.Z:=Зр.А ⊕ Зр.В

*21*

Зр Z, |Z|

*22*

**к**

Рис. 2

На рис. 2 изображена блок-схема алгоритма сложения чисел, представленных в обратных кодах, где обозначено: А, В – операнды; С – промежуточная переменная; З. р Z – знаковый разряд; К – коррекция; Z – результат; Хпр – двоичное значение операнда Х; Хдоп.  - значение операнда Х в обратном коде;

П1 – А>0, B<0; П2 – А<0, B>0; П3 – A<0, B<0.

**Работа алгоритма умножение чисел в обратном коде.**

Блок 1 алгоритма является начальным (рис.2).

В блоке 2 алгоритма происходит ввод чисел А и В в десятичной системе счисления. В этом блоке числа переводятся в двоичную систему счисления соответствующими процедурами с учетом знаков операндов. Числа А и В будут представлены в формате прямых кодов.

В блоке 3 алгоритма анализируется признак П1 – при котором операция умножения выполняется когда число В имеет единицу в знаковом разряде, т.е. число отрицательное, А > 0, В < 0 .

В блоке 4 алгоритма анализируется признак П2 – при котором операция умножения выполняется когда число А имеет единицу в знаковом разряде, т.е. число отрицательное А< 0, В > 0 .

В блоке 5 алгоритма анализируется признак П3 – при котором операция умножения выполняется когда числа А и В имеют единицы в знаковых разрядах, т.е. оба числа отрицательные, А< 0, В < 0 .

В блоке 6 алгоритма вычисляется произведение чисел А и В, где число А представлена в прямом коде, а число В в обратном по команде:

С : = |Апр| \* |Вобр|.

В блоке 7 алгоритма вычисляется произведение чисел А и В, где число А представлена в обратном коде, а число В в прямом по команде:

С : = |Аобр| \* |Впр|.

В блоке 8 алгоритма вычисляется произведение чисел А и В, где числа А и В представлены в обратных кодах по команде: С : = |Аобр| \* |Вобр|.

В блоке 9 алгоритма вычисляется коррекция произведения - К по формуле:

К : = 2n |Апр| - |Апр|.

В блоке 10 алгоритма вычисляется коррекция произведения - К по формуле: К : = 2n |Впр| - |Впр|.

В блоке 11 алгоритма вычисляется коррекция произведения - К по формуле: К : = 2n (|Аобр| + |Вобр|) - (|Аобр| + |Вобр|).

В блоках 12 и 13 алгоритма вычисляется результат Z в прямом коде по формуле: Z : = K - |Cпр|.

В блоке 14 алгоритма вычисляется результат Z в прямом коде по форму-ле: Z : = С - |Кпр|.

В блоках 15 и 16 алгоритма вычисляется результат Z в обратном коде по формуле: Z : = K + |Cобр|.

В блоке 17 алгоритма результат Z вычисляется в обратном коде по формуле: Z : = С + |Кобр|.

В блоках 18 и 19 алгоритма по формуле: Z : = |Апр| \* |Впр| вычисляется произведение операндов, числа А и В в этом случае положительные. Результата Z будет представлен в прямом коде. Знаковому разряду произведения Зр.Z присваивается знак одного из чисел по команде:

Зр.Z : =Зр.А.

В блоке 20 алгоритма по операции суммы по модулю два определяется знаковый разряд результата Z по команде:

Зр.Z : =Зр.А ⊕ Зр.В.

В блоке 21 алгоритма выдается знаковый разряд произведения Зр.Z и результата по модулю |Z|.

Блок 22 алгоритма является конечным блоком.

Примеры: умножение чисел в обратном коде.

1. А = 0.0111пр (7) А = 0.0111об х 0111 Апр

В = 1.1001пр (-9) В = 1.0110об 0110 Воб

Z = |Апр| \* |Впр| ) = |A| дп \* [(2n - |B|об) –1] = + 0010 1010 Спр

= 2n |A|пр – [( |A| пр \* |B|об) + |A| дп ] 0111 Апр

0011 0001 Спр

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 0 ⊕ 1 = 1

Окончательный результат Z 2 = 1. 0011 1111

Z10 = -63

↓ об

+ 1100 1110 Соб

0111 0000 2nАпр

1 0011 1110

+ 1 перенос 0011 1111 Z10 (63)

1. А = 1.101пр (-5) А = 1.010об х 010 Аоб

В = 0.011пр (3) В = 0.011об 011 Впр

Z = |Апр| \* |Впр| ) = [(2n - |А|об) –1] \* |В|дп = 000 110 Спр

= [2n |В|пр – ( |A| об \* |B|пр)] - |В| дп ↓ об

111 001 Соб

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 1 ⊕ 0 = 1

Окончательный результат Z 2 = 1. 001 111

Z10 = -15

011 000 2nВпр

1 010 001

+ 1 перенос

+ 010 010 Спр

111 100 Воб

1 001 110

+ 1 перенос

001 111 Zпр (15)

1. А = 1.1000пр  (-8) А = 1.0111об х 0111 Аоб

В = 1.0110пр (-6) В = 1.1001об 1001 Воб

Z = |Апр| \* |Впр| ) = [(2n - |А|об) –1] \* [(2n - |В|об) - 1] = 0011 1111 Соб

[( |A| об \* |B|об) - 2n( |А|об + |B|об)] + (|А|об + |B|об)

+ 0111 Аоб

Знаковый разряд произведения получается с

помощью операции - суммы по модулю два

ЗрZ = ЗрА ⊕ ЗрВ = 1 ⊕ 1 = 0

Окончательный результат Z 2 = 0. 0011 0000

Z10 = 48

1001 Воб

+1 0000

1 перенос

0001 Коб

0001 0000 2n Коб + 0011 1111 Соб 0010 1111 Соб + 2nКпр

↓ об 1110 1111 2n Кпр + 0000 0001 Коб

1110 1111 2n Кпр + 1 0010 1110 0011 0000 Zпр (48)

1

0010 1111 Соб + 2nКпр

Структурная схема операционного устройства выполняющего операцию умножение чисел в дополнительном и обратном коде.

**ЗрА Rg Aпр**

# **ЗрВ Rg Впр**

**ЗрВ Rg Вк**

**ЗрА Rg Aк**

А

В

**Пр** –перенос из старших разрядов чисел

**Кор**

**ЗрZ Rg Zпр**

**ЗрZ Rg Zк**

Рис.3

На рис.3 представлена структурная схема операционного автомата, выполня-ющего операцию умножение чисел в дополнительном и обратном коде, где представлено: ЗрА – знаковый разряд числа А, RgAпр – регистр для хранения модуля числа А в прямом коде, RgAк – регистр для хранения числа А в коде, ЗрВ – знаковый разряд числа В, RgВпр – регистр для хранения модуля числа В в прямом коде, RgВк – регистр для хранения числа В в коде, - сумматор, Кор- сумматор корректор, Зр Z –знаковый разряд числа Z, Rg Zк – регистр для хранения результата Z в коде, Rg Zпр – регистр для хранения результата Z в прямом коде.

Совмещенный алгоритм умножение чисел в дополнительном и обратном коде.

1

**начало**

**1**

**0**

2

**КО**

4

3

**Умножение чисел в**

**обр. коде**

**Умножение чисел в**

**доп. коде**

5

**|Z|, З.р.Z**

6

**конец**

#### Рис.4

На рис.4 обозначено: КО – код операции; если КО = 0, то происходит выполнение алгоритма сложения чисел в дополнительном коде, если КО = 1, то происходит выполнение алгоритма сложения чисел в обратном коде.

**2. Задание**

1. Составить программы на языке высокого уровня по представленным блок-схемам алгоритмов:

* умножение чисел, заданных в дополнительных кодах (рис.1),
* умножение чисел, заданных в обратных кодах (рис.2),

1. Совмещенный алгоритм умножение чисел в дополнительном и обратном коде (рис.4).
2. Промоделировать (тестировать) программы на ПЭВМ.
3. Проанализировать результаты выполнения программ.
4. **Содержание отчета**

Отчет должен содержать:

* Титульный лист;
* Задание;
* Описание переменных, используемых в программе;
* Блок-схемы алгоритмов;
* Описание работы алгоритмов;
* Тексты программы;
* Результаты выполнения работы программ.

###### **Контрольные вопросы**

1. Назовите способы умножения чисел в прямых кодах.
2. Сколько существует вариантов построение комбинационных схем машинного умножения двоичных чисел.
3. По каким правилам формируется произведение чисел с фиксированной запятой.
4. Назовите основные принципы умножения двоичных чисел с фиксированной точкой в дополнительном коде.
5. Назовите основные принципы умножения двоичных чисел с фиксированной точкой в обратном коде.
6. По каким формулам вычисляются коррекции произведения в дополнительном коде.
7. Какие формулы применяются при вычислении коррекции произведения в обратном коде.
8. На чем основывается принцип построения и работы операционного блока для умножения двоичных чисел.
9. Поясните назначение узлов операционного блока для вычисления коррекции произведения при умножении двоичных чисел с фиксированной запятой (рис. 3).
10. Что такое признаки выполнения операции умножения в кодах, как они определяются.
11. Какие обозначения применяются в алгоритме умножения чисел с фиксированной запятой.
12. Какой длины отводится разрядная сетка под результат при выполнении операции умножения двоичных чисел.
13. Назовите основные отличия получения коррекции при умножении чисел в дополнительном и обратном коде.
14. Как определяется знаковый разряд результата при умножении чисел с фиксированной запятой.

**Библиографический список**

1. Самофалов К.Г., Романкевич А.М., Валуйский В.Н. Прикладная теория цифровых автоматов. – Киев: Высш. шк., 1987 – 374 с. : ил.
2. Карцев М.А. Арифметика цифровых машин. –М.: Наука. 1969. – 575 с.
3. Савельев А.Я. Прикладная теория цифровых автоматов. –М.: Высш. шк., 1987. – 271 с: ил.
4. Каган Б.М. Электронные вычислительные машины и системы. – М.: Энергия 1979. - 528 с.
5. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. – Л.: Машиностроение. 1974. – 431 с.
6. Поспелов Д.А. Арифметические основы вычислительных машин дискретного действия. – М.: Высш. шк. 1970. – 307 с.
7. Соловьев Г.Н. Арифметические устройства ЭВМ. – М.: Энергия. 1978. –177 с.
8. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники: Учеб. пособие для электротех. спец. вузов. – М.: Высш. шк., 1988. – 311 с.: ил.
9. Преснухин Л.Н., Нестеров П.В. Цифровые вычислительные машины. – М.: Высш. шк. 1981, - 511 с.