МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждения высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
_________________ О.Г. Локтионова
«__1__»_________________ 2018 г.

МОДЕЛИРОВАНИЕ РАБОТЫ ОПЕРАЦИОННОГО АВТОМАТА, ВЫПОЛНЯЮЩЕГО ОПЕРАЦИЮ УМНОЖЕНИЯ ЧИСЕЛ В ДОПОЛНИТЕЛЬНОМ И ОБРАТНОМ КОДЕ

Методические рекомендации лабораторной работы №2 для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 «Информационная безопасность»

Курск 2018

УДК 004

Составитель: С.С. Шевелев

Рецензент

Кандидат технических наук, доцент кафедры
«Информационная безопасность» А.Л. Марухленко

Моделирование работы операционного автомата, выполняющего операцию умножения чисел в дополнительном и обратном коде [Текст] : методические рекомендации для лабораторной работы №2 по дисциплине «Организация ЭВМ и вычислительных систем» / Юго-Зап. гос. ун-т; сост.: С.С. Шевелев. – Курск, 2018. – 23 с.: ил. 4. – Библиогр.: с. 23.

Содержат сведения по вопросам моделирования работы операционного автомата, выполняющего операцию умножения чисел в дополнительном и обратном коде. Указывается порядок выполнения практических и самостоятельных работ, правила оформления отчета.

Методические рекомендации соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальности.

Предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00 «Информационная безопасность».

Текст печатается в авторской редакции

Подписано в печать 1.02.18. Формат 60х84 1/16.
Усл.печ. л. 1,34. Уч.-изд. л. 1,21. Тираж 100 экз. Заказ. 230 Бесплатно
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

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 произведений. 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.

Пример

Таблица.1

Заданы: A=0.1011, B=0.1101 Получить произведение $A \cdot B$ путем умножения с младших разрядов множителя B.

```
 x 0.1011
   0.1101
   ------
     1011
    0000
   1011
  1011
 ----------
 0.10001111
```

Получить произведение путем умножения со старших разрядов множителя

```
   0.1011
   0.1101
   ------
   1011
 +  1011
     0000
      1011
   ----------
   0.10001111
```

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

$$C = AB = A\sum_{-n}^{-1} b_{-i} 2^{i} = Ab_1 2^{-1} + Ab_2 2^{-2} + \ldots + Ab_{n-1} 2^{-(n-1)} + Ab_n 2^{-n}, \qquad (1)$$

где С — искомое произведение; A — множимое; B — 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.

$$C=(((\ldots((0 + A + Ab_n)\, 2^{-1}+Ab_{n-1})\, 2^{-1} + \ldots + Ab_{n-i})\, 2^{-i}+ \ldots Ab_2)2^{-1}+Ab_1)2^{-1}, \qquad (2)$$

которое может быть сведено к n-кратному выполнению цикла

$$C_{i+1} = (C_i + Ab_{n-i}).\, 2^{-1} \qquad (3)$$

при начальных значениях $i=0$; $Co = 0$.

В каждом цикле множимое либо добавляется к сумме частичных произведений (если $b_i=1$), либо нет (если $b_i=0$), после чего сумма частичных произведений умножается на 2^{-1}, т. е. сдвигается на один разряд вправо. После окончания *n-го* цикла образуется искомое произведение, т. е. $C_n= C = AB$.

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

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

$$C = 2^{-n}(Ab_n + 2Ab_{n-1} + ... + 2^i Ab_{n-i} + ... + 2^{n-1} Ab_1). \quad (4)$$

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

$$C_{i+1} = C_i + A_i b_{n-i}, \quad (5)$$

где A_i=2A_{i-1}, при начальных значениях i=0; C_0=0; A_n=A.

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

Для реализации данного способа умножения требуется n-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

$$C = 2^{-(n+1)}((...(...((0 + Ab_1)2 + Ab_2)2 + ... + Ab_{n-1})2 + Ab_n)2, \quad (6)$$

то умножение двух чисел *A* и *B* сводится к n-кратному повторению цикла

$$C_{i+1} = (C_i + Ab_{i+1})2 \quad (7)$$

с начальными значениями i = 0; Co = 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 *n* 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

$$C = A \cdot 2^{-1} b_1 + A \cdot 2^{-2} b_2 + ... + A \cdot 2^{-i} b_i + ... + A \cdot 2^{-(n-1)} b_{n-1} + A \cdot 2^{-n} b_n, \quad (8)$$

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

$$A_{i+1} = A_i \cdot 2^{-1}; \quad C_{i+1} = C_i + A_{i+1} b_{i+1} \quad (9)$$

при начальных условиях i=0; $A_0 = A$; $C_o = 0$.

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

$$\Delta_{1\max} = 2^{-(n+k+1)} - (2^{-(n+k+1)} + 2^{-(n+k+2)}) - (2^{-(n+k+1)} + 2^{-(n+k+2)} + 2^{-(n+k+3)}) - ... - (2^{-(n+k+1)} + 2^{-(n+k+2)}$$
$$+ ... + 2^{-(n+k)}) = -2^{-(n+k)}[(1-2^{-1}) + (1-2^{-2}) + ... + (1-2^{-(n-k)})] = -2^{-(n+k)}[n-k-(1-2^{-(n-k)})] =$$
$$= -2^{-(n+k)}(n-k-1). \quad (10)$$

2. После окончания умножения отбрасываются все k младших разрядов

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

$$\Delta_{2\max} = 2^{-n}(1-2^{-k}) < 2^{-n} \quad (11)$$

Если потребовать, чтобы $\Delta_{1\max}$ было меньше 2^{-n} , то из неравенства

$$(n-1-k)\cdot 2^{-(n+k)} < 2^{-n} \qquad (12)$$

можно определить количество дополнительных разрядов /г:

$$k > \log_2(n-k-1). \qquad (13)$$

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 $(n+1)$-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 A отрицательно, то значащие цифры его дополнительного кода образуют величину $|A_д| = 2^n - |A|$, число 2^n 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. $A > 0$, $B > 0$. Случай тривиальный, получаем сразу истинное значение положительного произведения:

$$Z = (|A|_{пр} * |B|_{пр}) \qquad (14)$$

2. $A > 0$, $B < 0$. Получаем после непосредственного применения алгоритма умножения:

$$(|A|_{пр} * |B|_{пр}) = |A|_{дп} * (2^n - |B|_{дп}) = 2^n |A|_{пр} - (|A|_{пр} * |B|_{дп}) \qquad (15)$$

Истинный результат в дополнительном коде составит:

$$|A_{пр}| * |B_{дп}| = K_{ор} - (|A|_{пр} * |B|_{дп}), \qquad (16)$$

где $K_{ор}$ – коррекция произведения.
Следовательно, требуется коррекция псевдорезультата в старшей части произведения на величину $K_{ор} = 2^n|A|_{пр}$, т. е. на величину дополнения множителя до единицы.

3. $A < 0$, $B > 0$. Получаем после непосредственного применения алгоритма умножения:

$$(|A|_{пр} * |B|_{пр}) = (2^n - |A|_{дп}) * |B|_{пр} = 2^n |B|_{пр} - (|A|_{дп} * |B|_{пр}) \qquad (17)$$

Истинный результат в дополнительном коде составит

$$|A_{дп}| * |B_{пр}| = K_{ор} - (|A|_{дп} * |B|_{пр}), \qquad (18)$$

где $K_{ор}$ – коррекция произведения.
Следовательно, требуется коррекция псевдорезультата на величину $K_{ор} = 2^n - |B|_{пр}$, т. е. на величину дополнения множителя до единицы.

4. $A < 0$, $B < 0$. После непосредственного применения алгоритма умножения получаем:

$$|A_{пр}| * |B_{пр}| = (2^n - |A|_{дп}) * (2^n - |B|_{дп}) = 2^{2n} - (2^n|A|_{дп}) - (2^n|B|_{дп}) + (|A|_{дп} * |B|_{дп}) = (|A_{пр}| * |B_{пр}|) - 2^n(|A|_{дп} + |B|_{дп}) \qquad (19)$$

Псевдорезультат требует коррекции на величину $K_{ор} = 2^n(|A|_{дп} + |B|_{дп})$, 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 A отрицательно, то значащие цифры его обратного кода образуют величину $| A_{об} | = 2^n| A | - | A |$. Поэтому при перемножении модулей кодов операндов в зависимости от сочетания знаков сомножителей могут возникнуть 4 случая.

1. $A > 0$, $B > 0$. В этом случае получаем сразу истинное значение положительного произведения:

$$Z = (|A|_{пр} * |B|_{пр}) \qquad (20)$$

2. $A > 0$, $B < 0$. Получаем после непосредственного применения алгоритма умножения:

$$(|B|_{пр} = (2^n - |B|_{об}) – 1 \qquad (21)$$

$$Z = (|A|_{пр} * |B|_{пр}) = |A|_{дп} * [(2^n - |B|_{об}) - 1] = 2^n|A|_{пр} - (|A|_{пр} * |B|_{об}) - |A|_{пр} \qquad (22)$$

Истинный результат в обратном коде составит

$$|A_{пр}| * |B_{дп}| = K_{ор} – (|A|_{пр} * |B|_{об}), \qquad (23)$$

где $K_{ор}$ – коррекция произведения.

Коррекция псевдорезультата равна: $K_{ор} = 2^n|A|_{пр} - |A|_{пр}$.

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

1) $Z = [2^n|A|_{пр} - (|A|_{пр} * |B|_{об})] - |A|_{пр}$, (24)
2) $Z = [2^n|A|_{пр} - |A|_{пр}] - (|A|_{пр} * |B|_{об})$, (25)
3) $Z = 2^n|A|_{пр} – [(|A|_{пр} * |B|_{об}) + |A|_{пр}]$. (26)

3. $A < 0$, $B > 0$. Получаем после непосредственного применения алгоритма умножения:

$$(|A|_{пр} = (2^n - |A|_{об}) - 1 \qquad (27)$$

$Z = (|A|_{пр} * |B|_{пр}) = [(2^n - |A|_{об}) - 1] * |B|_{пр} = 2^n |B|_{пр} - (|A|_{об} * |B|_{пр}) - |B|_{дп}$ (28)

Истинный результат в дополнительном коде составит

$$|A_{дп}| * |B_{пр}| = К_{ор} - (|A|_{об} * |B|_{пр}), \qquad (29)$$

где $К_{ор}$ – коррекция произведения. Следовательно, требуется коррекция псевдорезультата на величину $К_{ор} = 2^n|B|_{пр} - |B|_{пр}$.

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

1) $Z = [2^n|B|_{пр} - (|A|_{об} * |B|_{пр})] - |B|_{пр}$, (30)
2) $Z = [2^n|B|_{пр} - |B|_{пр}] - (|A|_{об} * |B|_{пр})$, (31)
3) $Z = 2^n|B|_{пр} - [(|A|_{об} * |B|_{пр}) + |B|_{пр}]$. (32)

4. $A < 0$, $B < 0$. После непосредственного применения алгоритма умножения получаем:

$$(|A|_{пр} = (2^n - |A|_{об}) - 1; \qquad (|B|_{пр} = (2^n - |B|_{об}) - 1 \qquad (33)$$

$Z = |A_{пр}| * |B_{пр}| = [(2^n - |A|_{об}) - 1] * [(2^n - |B|_{об}) - 1] = 2^{2n} - (2^n|A|_{об}) - (2^n|B|_{об}) +$
$+ (|A|_{об} * |B|_{об}) + |A|_{об} + |B|_{об} + 1 = (|A_{об}| * |B_{об}|) - 2^n(|A|_{об} + |B|_{об}) +$
$+ (|A|_{об} + |B|_{об})$
(34)

Псевдорезультат требует коррекции на величину

$$К_{ор} = (|A|_{об} + |B|_{об}) - 2^n(|A|_{об} + |B|_{об}) \qquad (35)$$

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

1) $Z = [(|A_{об}| * |B_{об}|) - 2^n(|A|_{об} + |B|_{об})] + (|A|_{об} + |B|_{об})$, (36)

2) $Z = [(|A_{об}| * |B_{об}|) + (|A|_{об} + |B|_{об})] - 2^n(|A|_{об} + |B|_{об}),$ (37)

3) $Z = [(|A|_{об} + |B|_{об}) - 2^n(|A|_{об} + |B|_{об})] + (|A_{об}| * |B_{об}|).$ (38)

Алгоритм умножение чисел, представленных в дополнительном коде

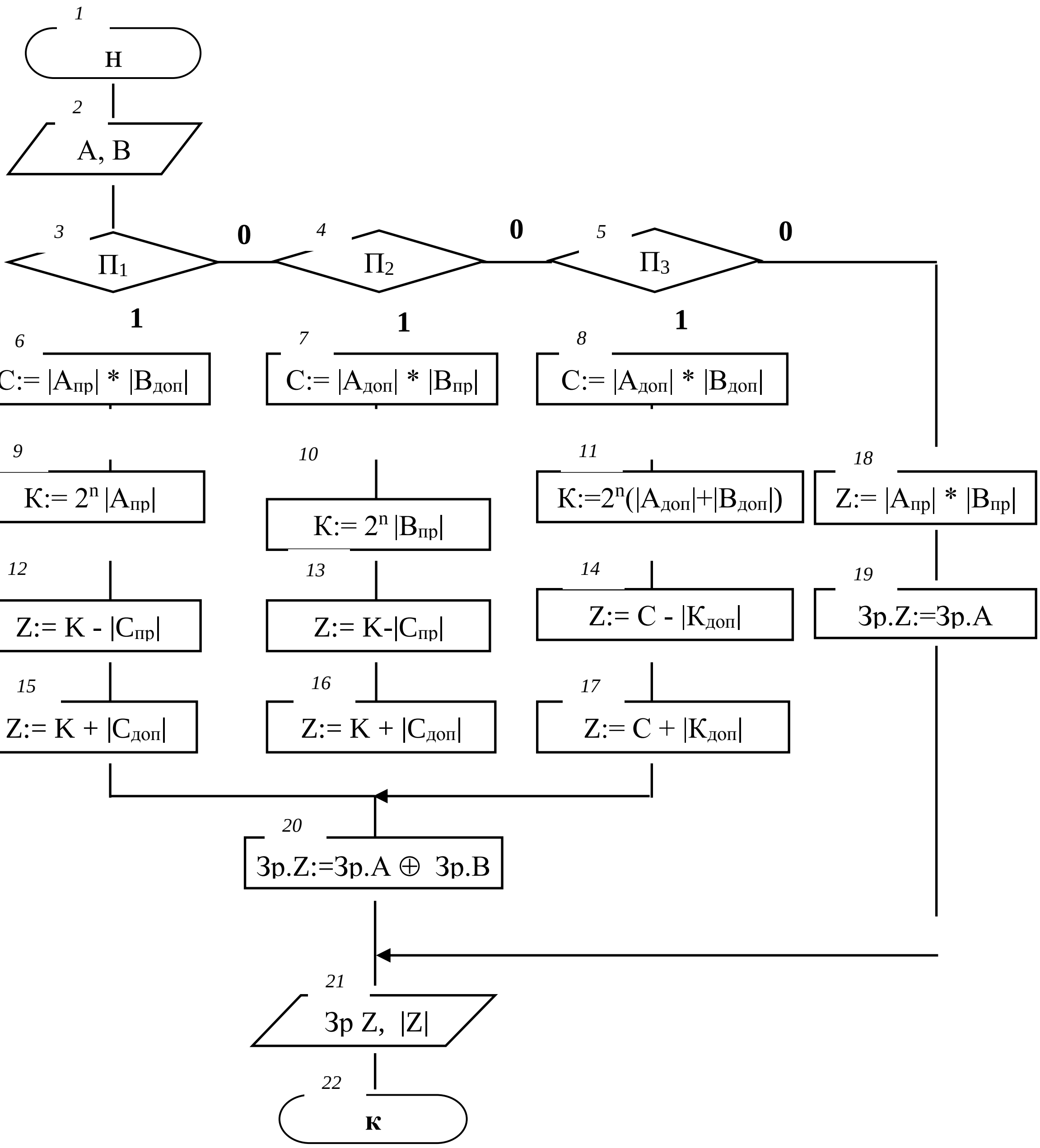

Рис.1

На рис.1 изображена блок-схема алгоритма сложения чисел, представленных в дополнительных кодах, где обозначено: A, B –

операнды; С – промежуточная переменная; З. р Z – знаковый разряд; К – коррекция; Z – окончательный результат; $X_{пр}$ – двоичное значение операнда Х; $X_{доп.}$ - значение операнда Х в дополнительном коде; $П_1$ – A>0, B<0; $П_2$ – A<0, B>0; $П_3$ – 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 $П_1$ – при котором операция умножения выполняется когда число В имеет единицу в знаковом разряде, т.е. число отрицательное, $A > 0$, $B < 0$.

В блоке 4 алгоритма анализируется признак $П_2$ – при котором операция умножения выполняется когда число А имеет единицу в знаковом разряде, т.е. число отрицательное $A < 0$, $B > 0$.

В блоке 5 алгоритма анализируется признак $П_3$ – при котором операция умножения выполняется когда числа А и В имеют единицы в знаковых разрядах, т.е. оба числа отрицательные, $A < 0$, $B < 0$.

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

$$C := |A_{пр}| * |B_{доп}|.$$

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

$$C := |A_{доп}| * |B_{пр}|.$$

В блоке 8 алгоритма вычисляется произведение чисел А и В, где числа А и В представлены в дополнительных кодах по команде: $C := |A_{доп}| * |B_{доп}|$.

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

$$К := 2^n |А_{пр}|.$$

В блоке 10 алгоритма вычисляется коррекция произведения - К по формуле: $К := 2^n |В_{пр}|$.

В блоке 11 алгоритма вычисляется коррекция произведения - К по формуле: $К := 2^n (|А_{доп}| + |В_{доп}|)$.

В блоках 12 и 13 алгоритма вычисляется результат Z в прямом коде по формуле: $Z := K - |С_{пр}|$.

В блоке 14 алгоритма вычисляется результат Z в прямом коде по форму-ле: $Z := С - |К_{доп}|$.

В блоках 15 и 16 алгоритма вычисляется результат Z в дополнительном коде по формуле: $Z := K + |С_{доп}|$.

В блоке 17 алгоритма результат Z вычисляется в дополнительном коде по формуле: $Z := С + |К_{доп}|$.

В блоках 18 и 19 алгоритма по формуле: $Z := |A_{пр}| * |B_{пр}|$ 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:

$$Зр.Z := Зр.А \oplus Зр.В.$$

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) A = 0.101 (5)

B= 1.111 (-7)

$Z = |A_{пр}| * |B_{доп}|) = |A|_{дп} * (2^n - |B|_{дп}) =$

$= 2^n |A|_{пр} - (|A|_{пр} * |B|_{дп})$

$$\begin{array}{rrl} \times & 101 & A_{пр} \\ & \underline{001} & B_{дп} \\ & 000\ 101 & C_{дп} \\ & \downarrow дп & \\ + & 111\ 011 & C_{пр} \\ & \underline{101\ 000} & 2^n A_{пр} \\ & 100\ 011 & |Z| = \end{array}$$

$(35)_{10}$

1 не учитывается

Знаковый разряд произведения получается с помощью операции - суммы по модулю два $ЗрZ = ЗрA \oplus ЗрB = 0 \oplus 1 = 1$

Окончательный результат $Z_2 = 1.\ 100\ 011$

$Z_{10} = -35$

2) $A = 1.110$ (-6)
$B = 0.100$ (4)
$Z = (|A|_{пр} * |B|_{пр}) = (2^n - |A|_{дп}) * |B|_{пр} =$
$= 2^n |B|_{пр} - (|A|_{дп} * |B|_{пр})$

$$
\begin{array}{rl}
\times\ 010 & A_{дп} \\
\underline{100} & B_{пр} \\
001\,000 & C_{дп} \\
\downarrow дп & \\
+\ 111\,000 & C_{пр} \\
\underline{100\,000} & B_{пр} \\
011\,000 & |Z|=
\end{array}
$$

1 не учитывается

Знаковый разряд произведения получается с помощью операции - суммы по модулю два $ЗрZ = ЗрA \oplus ЗрB = 1 \oplus 0 = 1$
Окончательный результат $Z_2 = 1.\,011\,000$
$Z_{10} = -24$

3) $A = 1.111$ (-7)
$B = 1.110$ (-6)
$Z = |A_{пр}| * |B_{пр}| = (2^n - |A|_{дп}) * (2^n - |B|_{дп}) =$
$= 2^{2n} - (2^n|A|_{дп}) - (2^n|B|_{дп}) + (|A|_{дп} * |B|_{дп}) =$
$= (|A_{пр}| * |B_{пр}|) - 2^n(|A|_{дп} + |B|_{дп})$

$$
\begin{array}{rl}
\times\ 001 & A_{дп} \\
\underline{010} & B_{дп} \\
000\,010 & C_{дп}
\end{array}
$$

$$
\begin{array}{rl}
+\ 001 & A_{дп} \\
\underline{010} & B_{дп} \\
011 & K_{(ор)дп} \\
101 & (K_{ор})_{пр}
\end{array}
$$

$$
\begin{array}{rl}
+\ 000\,010 & C_{дп} \\
\underline{101\,000} & K_{(ор)пр}\ в \\
 & \\
101\,010 & |Z| = (42)_{10}
\end{array}
$$

Знаковый разряд произведения получается с помощью операции - суммы по модулю два $ЗрZ = ЗрA \oplus ЗрB = 1 \oplus 1 = 0$ Окончательный результат $Z_2 = 0.\,101\,010$

Алгоритм умножение чисел, представленных в обратном коде

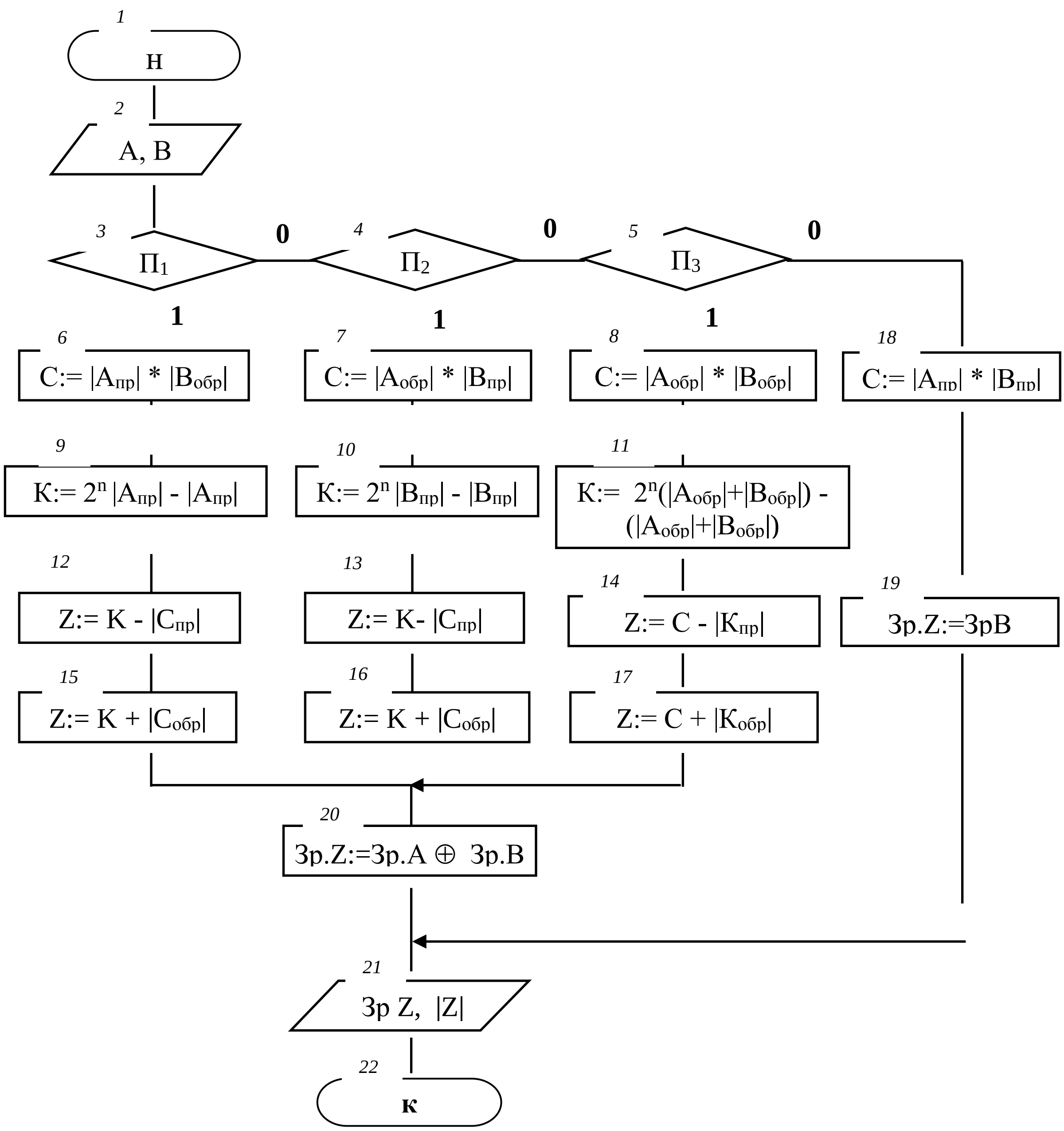

Рис. 2

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; $X_{пр}$ – двоичное значение операнда X; $X_{доп.}$ - значение операнда X в обратном коде;
$П_1$ – A>0, B<0; $П_2$ – A<0, B>0; $П_3$ – 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 $П_1$ – при котором операция умножения выполняется когда число В имеет единицу в знаковом разряде, т.е. число отрицательное, $A > 0$, $B < 0$.

В блоке 4 алгоритма анализируется признак $П_2$ – при котором операция умножения выполняется когда число А имеет единицу в знаковом разряде, т.е. число отрицательное $A< 0$, $B > 0$.

В блоке 5 алгоритма анализируется признак $П_3$ – при котором операция умножения выполняется когда числа А и В имеют единицы в знаковых разрядах, т.е. оба числа отрицательные, $A< 0$, $B < 0$.

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

$$C : = |A_{пр}| * |B_{обр}|.$$

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

$$C : = |A_{обр}| * |B_{пр}|.$$

В блоке 8 алгоритма вычисляется произведение чисел А и В, где числа А и В представлены в обратных кодах по команде: $C : = |A_{обр}| * |B_{обр}|$.

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

$$K : = 2^n |A_{пр}| - |A_{пр}|.$$

В блоке 10 алгоритма вычисляется коррекция произведения - К по формуле: $K : = 2^n |B_{пр}| - |B_{пр}|$.

В блоке 11 алгоритма вычисляется коррекция произведения - К по формуле: $K : = 2^n (|A_{обр}| + |B_{обр}|) - (|A_{обр}| + |B_{обр}|)$.

В блоках 12 и 13 алгоритма вычисляется результат Z в прямом коде по формуле: $Z := K - |C_{пр}|$.

В блоке 14 алгоритма вычисляется результат Z в прямом коде по форму-ле: $Z := C - |K_{пр}|$.

В блоках 15 и 16 алгоритма вычисляется результат Z в обратном коде по формуле: $Z := K + |C_{обр}|$.

В блоке 17 алгоритма результат Z вычисляется в обратном коде по формуле: $Z := C + |K_{обр}|$.

В блоках 18 и 19 алгоритма по формуле: $Z := |A_{пр}| * |B_{пр}|$ вычисляется произведение операндов, числа A и B в этом случае положительные. Результата Z будет представлен в прямом коде. Знаковому разряду произведения Зр.Z присваивается знак одного из чисел по команде:

$$Зр.Z := Зр.A.$$

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

$$Зр.Z := Зр.A \oplus Зр.B.$$

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) $A = 0.0111_{пр}$ (7) $A = 0.0111_{об}$

$B = 1.1001_{пр}$ (-9) $B = 1.0110_{об}$

$Z = |A_{пр}| * |B_{пр}|) = |A|_{дп} * [(2^n - |B|_{об}) - 1] =$

$= 2^n |A|_{пр} - [(|A|_{пр} * |B|_{об}) + |A|_{дп}]$

x	0111	$A_{пр}$
	0110	$B_{об}$
+	0010 1010	$C_{пр}$
	0111	$A_{пр}$
	0011 0001	$C_{пр}$
	↓ об	
+	1100 1110	$C_{об}$
	0111 0000	$2^n A_{пр}$
	1 0011 1110	
+	1	перенос
	0011 1111	Z_{10}

Знаковый разряд произведения получается с помощью операции - суммы по модулю два $ЗрZ = ЗрA \oplus ЗрB = 0 \oplus 1 = 1$ Окончательный результат $Z_2 = 1.0011\ 1111$

$Z_{10} = -63$

2) $A = 1.101_{пр}$ (-5) $A = 1.010_{об}$

$B = 0.011_{пр}$ (3) $B = 0.011_{об}$

$Z = |A_{пр}| * |B_{пр}|) = [(2^n - |A|_{об}) - 1] * |B|_{дп} =$

x	010	$A_{об}$
	011	$B_{пр}$
	000 110	$C_{пр}$

$= [2^n |B|_{пр} – (|A|_{об} * |B|_{пр})] - |B|_{дп}$

Знаковый разряд произведения получается с помощью операции - суммы по модулю два ЗрZ = ЗрA ⊕ ЗрB = 1 ⊕ 0 = 1 Окончательный результат Z_2 = 1. 001 111

$Z_{10} = -15$

↓ об

111 001 $C_{об}$

011 000 $2^n B_{пр}$

1 010 001

\+ 1 перенос

\+ 010 010 $C_{пр}$

111 100 $B_{об}$

1 001 110

\+ 1 перенос

001 111 $Z_{пр}$ (15)

3) $A = 1.1000_{пр}$ (-8) $A = 1.0111_{об}$

$B = 1.0110_{пр}$ (-6) $B = 1.1001_{об}$

$Z = |A_{пр}| * |B_{пр}|) = [(2^n - |A|_{об}) –1] * [(2^n - |B|_{об}) - 1] =$

$[(|A|_{об} * |B|_{об}) - 2^n(|A|_{об} + |B|_{об})] + (|A|_{об} + |B|_{об})$

x 0111 $A_{об}$

1001 $B_{об}$

0011 1111 $C_{об}$

\+ 0111 $A_{об}$

1001 $B_{об}$

\+ 1 0000

1 перенос

Знаковый разряд произведения получается с помощью операции - суммы по модулю два ЗрZ = ЗрA ⊕ ЗрB = 1 ⊕ 1 = 0 Окончательный результат Z_2 = 0. 0011 0000

0001 $K_{об}$

0001 0000 $2^n K_{об}$

↓ об

1110 1111 $2^n K_{пр}$

\+ 0011 1111 $C_{об}$

1110 1111 $2^n K_{пр}$

\+ 1 0010 1110

1

0010 1111 $C_{об} + 2_n K_{пр}$

0010 1111 $C_{об} + 2_n K_{пр}$

\+ 0000 0001 $K_{об}$

0011 0000 $Z_{пр}$ (48)

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

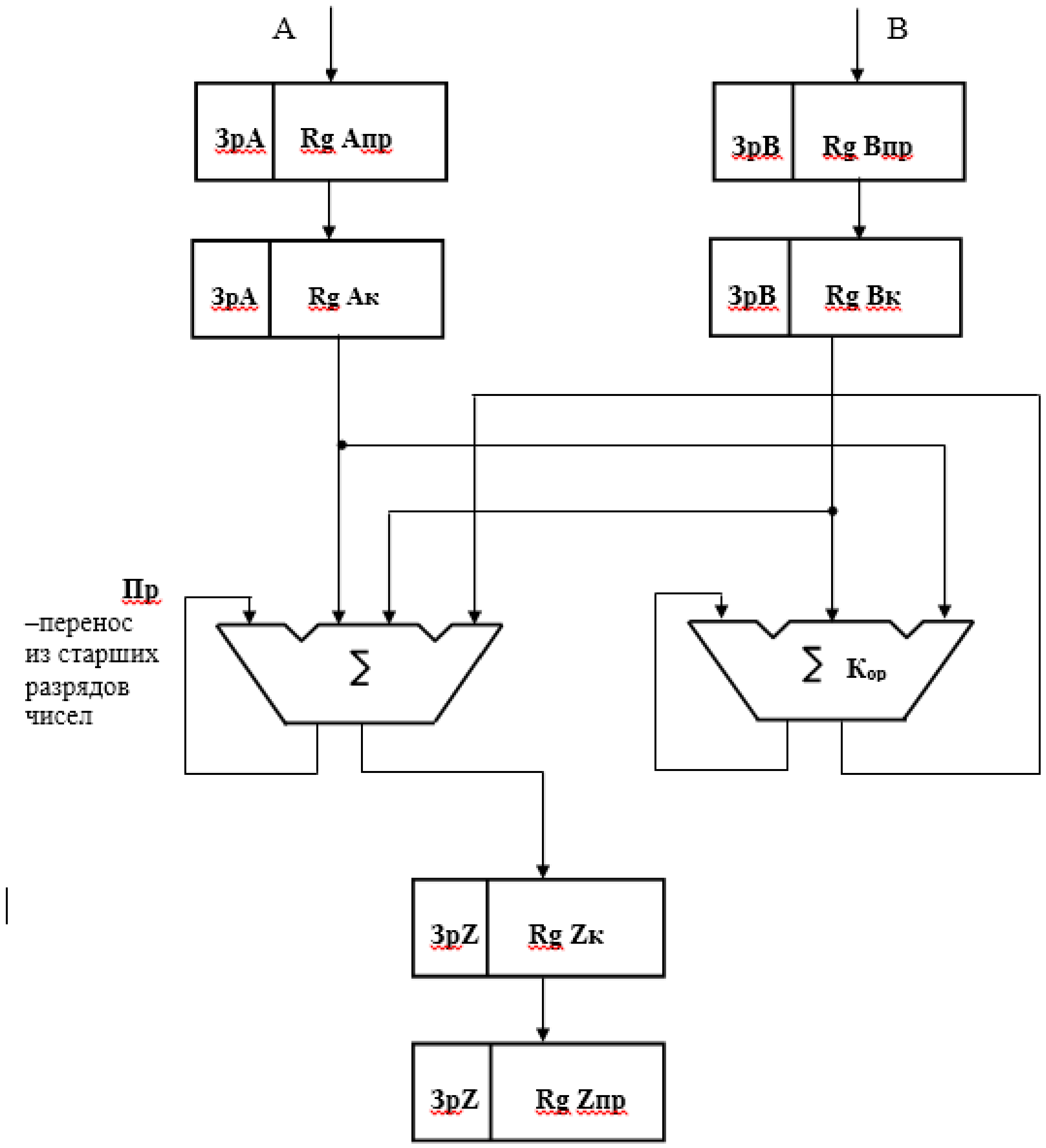

Рис.3

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, RgАпр – регистр для хранения модуля числа А в прямом коде, RgАк – 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, $\sum$ - сумматор, $К_{ор}$- 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.

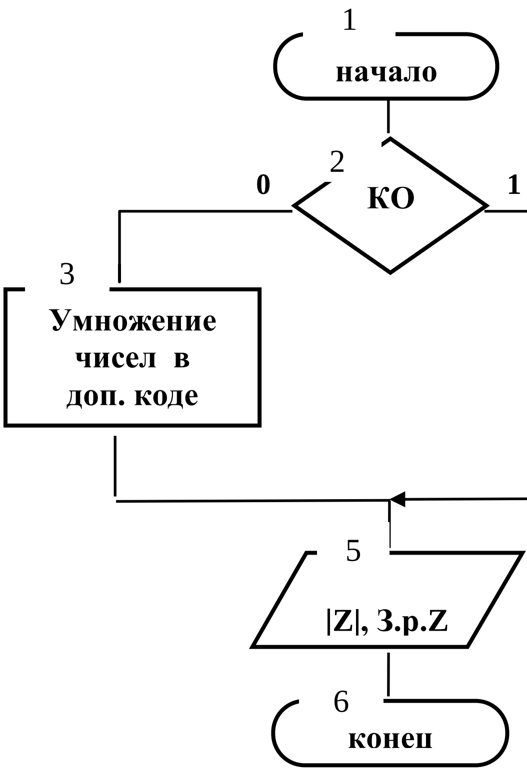

Рис.4

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

3. 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.