ОПЕРАЦИИ ДЕЛЕНИЯ

Операция деления в ЭВМ - это последовательность вычитаний и сдвига.

$$z = int\left(\frac{x}{y}\right), \quad s = x - zy, \qquad s < y$$

X-делимое, представляется целым числом удвоенной разрядности (2n) цифровых разрядов);
Y- делитель, представляемые словами, содержащими n- цифровых разрядов
Z-частное- представляемые словами, содержащими n- цифровых разрядов.

Операция выполняется за n итераций и может быть описана следующим образом:

$$s^{i} = 2S^{\mathrm{i}-1} - z_{n-\mathrm{i}}(2^{n} \cdot y), \text{ при } \quad S^{0} = \mathrm{Y},\ S^{n} = 2^{n}\,\mathrm{S}$$

$$z_{\mathrm{n}-i} = \begin{cases} 1, \text{ если } (2S^{(i-1)} - 2^{n}\ \mathrm{x}) \geq 0 \\ 0, \text{ если } (2S^{(i-1)} - 2^{n}\ \mathrm{x}) < 0 \end{cases}$$

После *n* итераций получается

$$S^{n} = 2^{n} \cdot S^{0} - Z\left(2^{n}\,Y\right) = 2^{n}[\mathrm{X} - (\mathrm{Z} * \mathrm{Y})] = 2^{n}Y$$

Частное от деления 2n-разрядного числа на n-разрядное может содержать более, чем *n* разрядов. В этом случае возникает переполнение, поэтому перед выполнением деления необходима проверка условия

$$\mathrm{Z} < (2^{n} - 1)\mathrm{Y} + \mathrm{Y} = 2^{n}\,\mathrm{Y}$$

Делению подвергаются целые числа в прямом коде. Z-слово, поэтому должно выполняться неравенство $|Z| < 2^{n-1}$. Это возможно при $(|X'| - |Y|) < 0$, где $|X'| = |X| \cdot 2^{-(n-1)}$

Для получения $(|X'| - |Y|)$ следует вычесть из ДМ делитель.

Если результат пробного вычитания >0, то $|Z| \geq 2^{n-1}$ и деление невозможно, если результат <0, то можно выполнить деление.

Пример ручного счета 38:5 =7 остаток 3
Z=X/Y

```
 00100110 | 0101
 0101     | 0111
-1101<0 пробное вычитание
  0100
- 0101
  1111<0
   1001
-  0101
    0100 >0
    1001
-    0101
     0100>0
     1000
   - 0101
     0011>0
```

X-делимое, представляется целым числом удвоенной разрядности (2n) цифровых разрядов;
Y- делитель, представляемые словами, содержащими n- цифровых разрядов
Z-частное- представляемые словами, содержащими n- цифровых разрядов.

ЭВМ

0100110 0101 *38/5*

0101

1111<0 **0**

1001

0101

0100>0 **1**

1001

0101

0100>0 **1**

1000

0101 >0 **1 частное 0111**

0011 -остаток

ЭВМ

Алгоритм деления целых чисел с восстановлением остатка:

1. Числа представлены в прямом коде. Прием делимого и делителя.
2. Получение модулей, анализ знака.

3. Проверка на корректность деления: для целых чисел частное должно укладываться в разрядную сетку одинарной длины. Для дробных чисел делимое должно быть меньше делителя.

4. Исходное значение частичного остатка (промежуточные суммы) полагается равным старшим разрядам делимого. Частичный остаток удваивается путем сдвига на 1 разряд влево. При этом в высвободившийся младший разряд частичного остатка заносится очередная цифра делимого.

5. Производится вычитание делителя (прибавление в дополнительном коде) и анализ полученного остатка. Если частичный остаток меньше 0, то цифра частного равна 0 и в этом же такте производится восстановление остатка прибавлением делителя. Если частичный остаток больше 0, то очередная цифра частного равна 1. Такая последовательность действий повторяется до получения всех цифр модуля частного.

6. Пункты 4-5 последовательно выполняются для получения всех цифр модуля частного.

7.Знак частного формируется отдельно; при реализации алгоритма выполняется проверка делимого на 0 и делителя на 0.

41/6=6(5)

0010 1001 :0110 *0110*=+6

0101 001* *1010*=-6

+1010

1111 001***0*** частичный остаток<0

+0110*такт восстановления*

0101 001***0***

1010 01***0****

+1010

0100 01***01*>0**

1000 1***01****

+1010

0010 1*01** >0

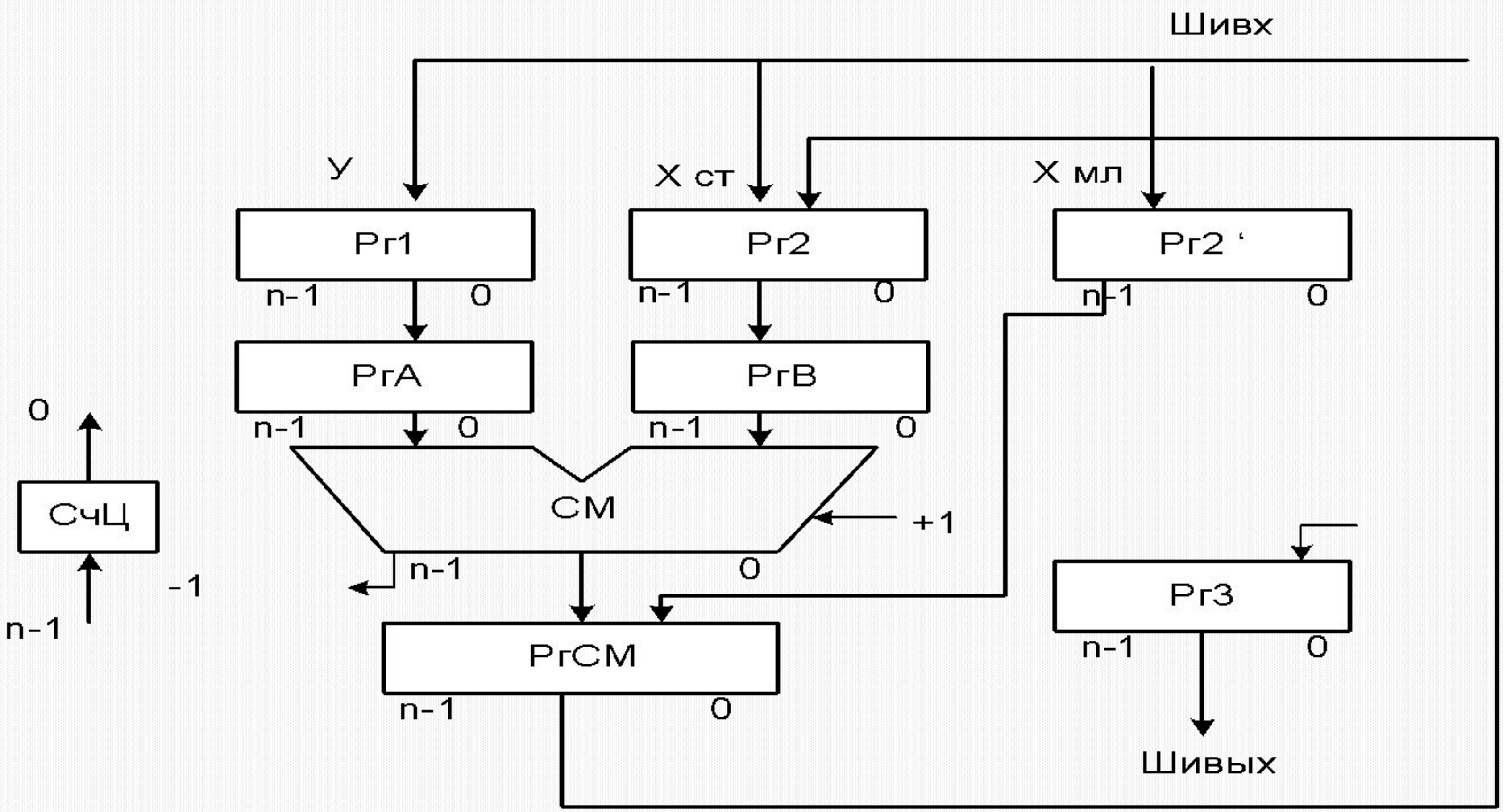
0101 *011**

+1010

1111 *0110*<0

+ 0110-*такт восстановления*

0101- остаток



Шивх
У
Х ст
Х мл
Рг1
Рг2
Рг2 '
n-1
0
РгА
РгВ
СМ
+1
СчЦ
-1
РгСМ
Рг3
Шивых

Приведем алгоритм деления без восстановления остатка

1. Исходное значение частичного остатка полагается равным старшим разрядам делимого.
2. Частичный остаток удваивается путем сдвига на один разряд влево. При этом в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра частного.
3. Из сдвинутого частичного остатка вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицательный.
4. На каждом шаге содержимое регистра делимого (РгДМ) и регистра частного (РгЧ) сдвигается на один разряд влево.
5. В зависимости от сочетания знаков частичного остатка и делителя определяется значение очередной цифры частного и требуемое действие: вычитание или прибавление делителя.
6. Вычитание делителя производится посредством прибавления дополнительного кода делителя. Преобразование в дополнительный код осуществляется за счет передачи делителя на вход сумматора обратным (инверсным) кодом с последующим добавлением единицы к младшему разряду сумматора.
7. Данная процедура повторяется для всех цифр делимого, о чем свидетельствует нулевое содержимое счетчика циклов(исходное значение СЧЦ равно разрядности делителя). Содержимое СЧЦ уменьшается на единицу после каждой итерации.
8. По окончании операции деления частное располагается в регистре частного, а в регистре делимого будет остаток от деления.
9. На заключительном этапе, если это необходимо, производится корректировка полученного результата, как это предусматривает алгоритм деления чисел со знаком

*На практике для накопления и хранения частного вместо отдельного регистр используют освобождающиеся в процессе сдвигов младшие разряды регистра делимого.

Особенности алгоритма деления с неподвижным делителем без восстановления остатка, для чисел представленных в прямом коде.

Отличие от предыдущего алгоритма:

-если знак частичного остатка больше 0, то выполняется вычитание делителя,

- если знак частичного остатка меньше 0, то производится прибавление делителя.

Представленная на рисунке структура АЛУ используется для деления чисел на сумматоре прямого и дополнительного кодов. При использовании прямого кода в структуру вводят триггеры для хранения знаков.

5:2=2(1)

000101 010=+2 ; 110=-2

← 00101_

+110

11101*0*

← **1101*0*_**

+010

0001*01*

← **001*01*_**

+110

111*010*

частное

Остаток в доп. Коде *111доп.код=001 в прямом коде*

Особенности деления чисел представленных дополнительным кодом

(алгоритм деления без восстановления остатка)

Так как делимое и делитель могут иметь разные знаки, то характер арифметического

д

Знак остатка	Знак ДТ	Арифметическое действие
+	+	вычитание
+	-	сложение
-	+	сложение
-	-	вычитание

2. Формирование цифр частного

Если знак остатка равен знаку ДТ, то цифра частного равна 1; в противном случае цифра частного равна 0.

3. Коррекция частного. Если:

$X>0$; $Y<0$; то к частному «+1» (ДМ>0, ДТ<0)

$X<0$; $Y>0$; то к частному «+1, остаток $\neq 0$; (ДМ<0, ДТ>0)

$X<0$; $Y<0$; то к частному «+1», остаток =0; (ДМ<0, ДТ<0)

4. Коррекция остатка.

Если знак остатка равен знаку ДМ, то выполняется коррекция (арифметическое действие определяется по таблице)

Знак остатка	Знак ДТ	Арифметическое действие
+	+	вычитание
+	-	сложение
-	+	сложение
-	-	вычитание

-5:-2=+2(1) 010=+2: 110=-2

111011 : 110 000101=+5 ;

11011_ 111011= -5

+010

000110

00110_

+110

111101

11101_

+010

$$\underbrace{001}_{\text{остаток}}\underbrace{010}_{\text{частное}}$$

Зн ДО	Зн ДТ	арифм. действие
+	+	вычит.
+	−	слож.
−	+	слож.
−	−	вычит.

21 : −5

0.101 +5
1.011 −5

0001	0101
← 0010	101
+1011	
1101	1011 1
← 1011	011 1 .
+ 0101	
0000	011 1 0
← 0000	11 1 0 .
+ 1011	
1011	1 1 0 1
← 0111	1 0 1 .
+ 0101	
1100	1 1 0 1 1
+ 0101	+ 1
0001 остаток	11100 частное в доп. коде

корр. всегда

(−100) в пр. коде

-21 : 5

0.10101 (+21)

1.01011 (-21)

1110 | 1011   0.101

← 1101 | 0111.

\+ 0101

0010 | 01111

← 0100 | 1111.1

\+ 1011

1111 | 11110

← 1111 | 1110.

\+ 0101

0100 | 11101

← 1001 | 1011.

\+ 1011

0100 | 1,0,1,1

\+ 1011   + 1

1111   1100 д.к.

д.к.

−21 : −5

1110 | 1011 : 1,011

← 1101 | 0111

\+ 0101

0010 | 01110

← 0100 | 1110

\+ 1011

1111 | 11101

← 1111 | 1011

\+ 0101

0100 | 10110

← 1001 | 0110

\+ 1011

0100 | 0100 пр. кор.

\+ 1011

1111

д.к.

x < 0; y < 0;

корр., когда остаток = 0

Ускорить деление возможно, используя следующее:

- замена делителя обратной величиной, с последующим ее умножением на делимое

- сокращение времени вычисления частичных остатков в традиционных методах деления (с восстановлением или без восстановления остатка) за счет ускорения операций суммирования (вычитания);

- сокращение времени вычисления за счет уменьшения количества операций сум мирования (вычитания) при расчете Ч0;

- вычисление частного в избыточной системе счисления.

Замена деления умножением на обратную величину

Операцию деления заменить на умножением

$$z = x \cdot \frac{1}{y}$$

Основная задача - эффективное вычисление *1/y.*

Задача решается одним из двух методов; с помощью ряда Тейлора или метода Ньютона-Рафсона.

При реализации первого метода делитель *ДТ* представляется в виде: *D* = *1+X.* Тогда для двоичного представления *D* можно записать:

$$\frac{1}{D} = (1 - X) \times (1 + X^2) \times (1 + X^4) \times (1 + X^8) \times (1 + X^{16})\ldots$$

Возможные значения сомножителей в правой части выражения извлекались из таблицы емкостью 28 байт, хранящейся в памяти. Операция вычисления *1/D* требует шести умножений.

Вычисление величины *1/D* методом Ньютона-Рафсона сводится к нахождению корня уравнения

$$f(X) = \frac{1}{X} - D = 0,$$

то есть X= 1/D. Решение может быть получено с привлечением рекуррентного соотношения: $X_{i+l} = X_i\ (2 - X_i D).$

Количество итераций определяется требуемой точностью вычисления *X/D*. Реализация метода для n-разрядных чисел требует 2 int($\log_2 n$) - 1 операций умножения.
,

Ускорение вычисления частичных остатков

Возможности данного подхода весьма ограничены и связаны в основном с ускорением операций сложения (вычитания). Способы достижения этой цели ничем не отличаются от тех, что применяются, например, при выполнении умножения. Это различные приемы для ускорения распространения переноса, матричные схемы сложения и т. п.

Алгоритм ускоренного деления SRT

Свое название алгоритм получил по фамилиям авторов (Sweeney, Robertson, Tocher), разработавших его независимо друг от друга приблизительно в одно и то же время.

$$q_i = \begin{cases} 1, \text{ если } 2S^{(i-1)} \geq D; \\ 0, \text{ если } -D \leq 2S^{(i-1)} < D; \\ -1, \text{ если } 2S^{(i-1)} < -D, \end{cases}$$

$$S^{(i)} = 2S^{(i-1)} - q_i D.$$

Делимое и делитель, представленные в дополнительном коде, размещаются в регистре делимого и делителя соответственно. Дальнейшие действия можно описать следующим образом.

1. Если в делителе D имеются k предшествующих нулей (при D > 0) или предшествующих единиц (при *D*< 0), то производится предварительный сдвиг содержимого РгДМ и РгДТвлево на ***к*** разрядов.

2. Для i от 0 до n-1:

- если- три старших цифры частичного остатка в РгДМ совпадают, то q_i =0 и производится сдвиг
содержимого РгДМ на один разряд влево;
- если три старших цифры частичного остатка в РгДМ не совпадают, а сам ЧО отрицателен, то
q_i =-l, делается сдвиг содержимого РгДМ на один разряд влево и к ЧО прибавляется делитель;
- если три старших цифры частичного остатка в РгДМ не совпадают, а сам ЧО положителен, то q_i = +1,выполняется сдвиг содержимого РгДМ на разряд влево и из ЧО вычитается делитель.

3. Если после завершения пункта 2 остаток отрицателен, то производится коррекция (к остатку прибавляется делитель, а из частного вычитается единица).

4. Остаток сдвигается вправо на ***к*** разрядов.

41/9

0.0000101001

0.00101001

0 01010010

0 1010010 1

+1 0111

_0 00010101

0 00101010

0 01010100

0.1001 +9

_1.0111 -9

ЭВМ

38/6

0.0000100110

0.00100110 0.0110

0.01001100 01100 +6

_01001100 _1.0100 -6

+10100

1110110010

_110110010-1 _100

_1011000011 -1

+01100 011

00010 00110

ЭВМ