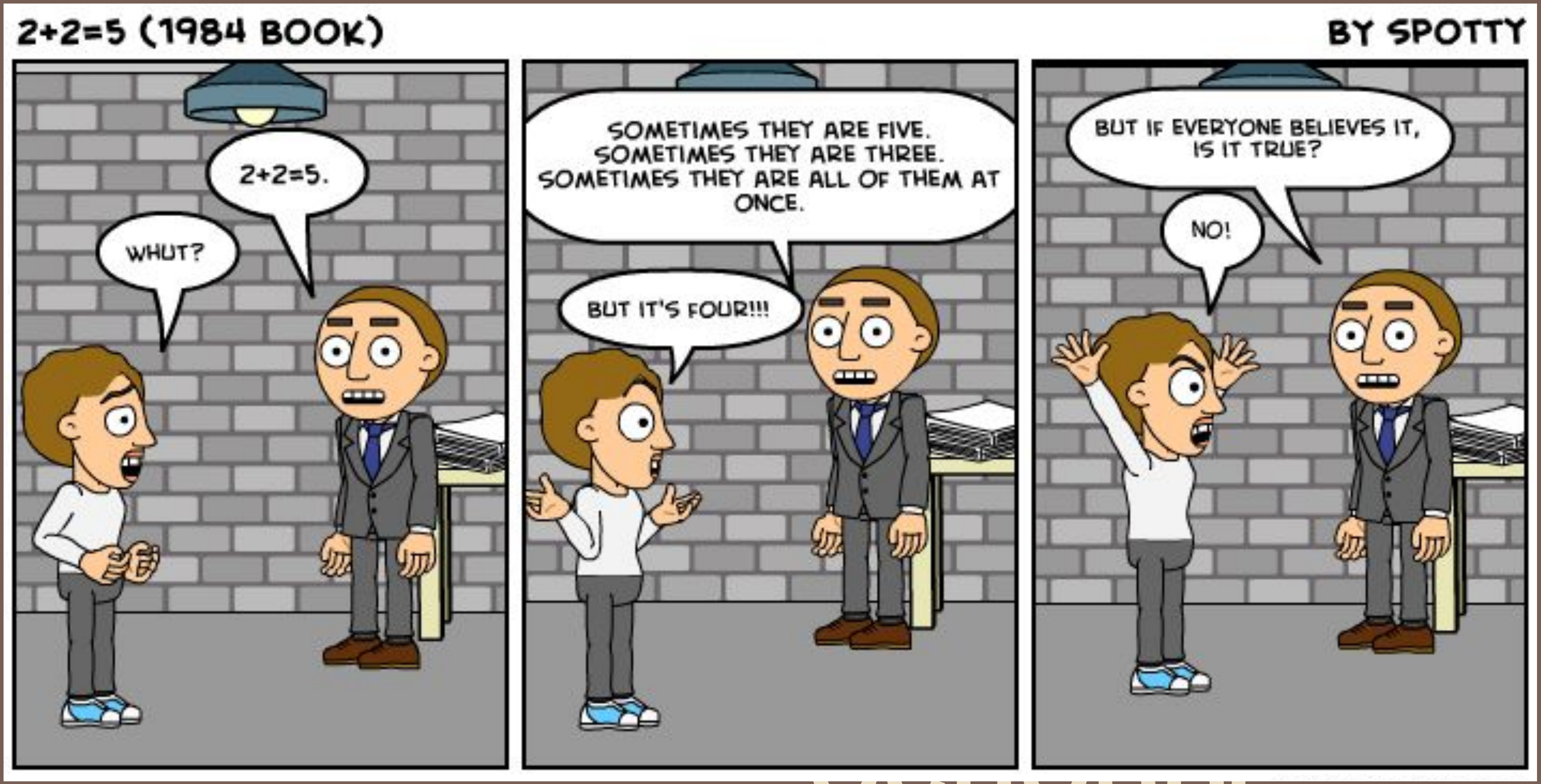


КОМПЬЮТЕРНОЙ ТЕХНИКИ

2016

Парамонов А.И.

Арифметические операции над положительными числами

ДВОИЧНАЯ АРИФМЕТИКА

1+1 = 10? !!!

Таблица двоичного сложения	Таблица двоичного вычитания	Таблица двоичного умножения
0+0=0 0+1=1 1+0=1 1+1=10	0-0=0 1-0=1 1-1=0 10-1=1	0×0=0 0×1=0 1×0=0 1×1=1

Пример: Выполнить сложение двоичных чисел **X=1101, Y=101**

единицы переноса

```
      1  1
X=     1101
    +
Y=      101
     ------
X+Y=  10010
```

Таблица двоичного сложения
0+0=0 0+1=1 1+0=1 1+1=10

Результат 1101+101=10010.

Пример: Выполнить сложение двоичных чисел **X=1101, Y=101, Z=111**

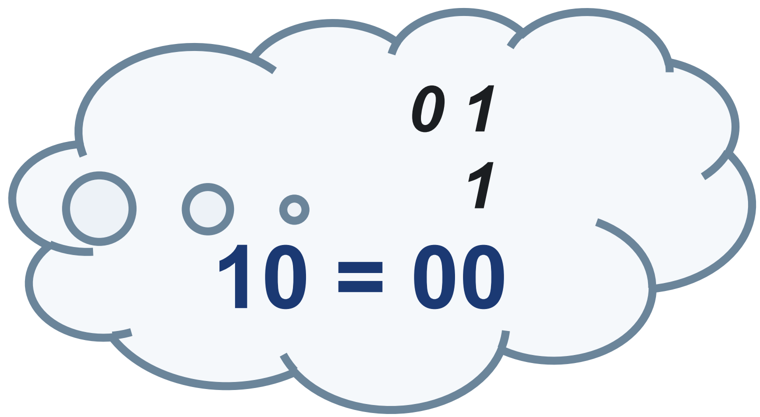
единицы переноса

$$\begin{array}{lr}
 & 1 \\
 & 111 \\
X= & 1101 \\
+ & \\
Y= & 101 \\
+ & \\
Z= & \underline{111} \\
X+Y+Z= & 11001
\end{array}$$

Результат 1101+101+111=11001.

Пример: Выполнить вычитание из двоичного числа **X=10010** числа **Y=101**

$$\begin{array}{r} \dot{1}00\dot{1}0 \\ -\,101 \\ \hline 01101 \end{array}$$



Результат 10010 - 101=1101

Пример: Выполнить умножение двоичных чисел **X=1001, Y=101**

$$
\begin{array}{r}
1001 \\
\times \quad 101 \\
\hline
1001 \\
1001 \\
\hline
101101
\end{array}
$$

Таблица двоичного умножения
$0 \times 0 = 0$ $0 \times 1 = 0$ $1 \times 0 = 0$ $1 \times 1 = 1$

Результат $1001 \times 101 = 101101$

Пример: Выполнить деление двоичного числа **X=1100.011** на число **Y=10.01**

$$
\begin{array}{rl}
-\,110001.1 & \big|\ 1001 \\
\underline{1001} & \big|\overline{\ 101.1\ } \\
-\,1101 & \\
\underline{1001} & \\
-\,1001 & \\
\underline{1001} & \\
0 &
\end{array}
$$

Результат 1100.011 : 10.01=101.1

Арифметические операции в Р-ичных системах счисления

Во всех позиционных с/с арифметические операции выполняются по одним и тем же правилам согласно соответствующим таблицам сложения и умножения.

Для всех систем счисления справедливы одни и те же законы арифметики:

- *коммутативный,*
- *ассоциативный,*
- *дистрибутивный,*

а также правила сложения, вычитания, умножения и деления столбиком.

Сложение

В *Р*-ичной системе счисления таблица сложения представляет собой результаты сложения каждой цифры алфавита *Р*-ичной системы с любой другой цифрой этой же

+	0	1
0	0	1
1	1	10

+	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

Таблицы сложения двоичной и троичной системы счисления

Деление

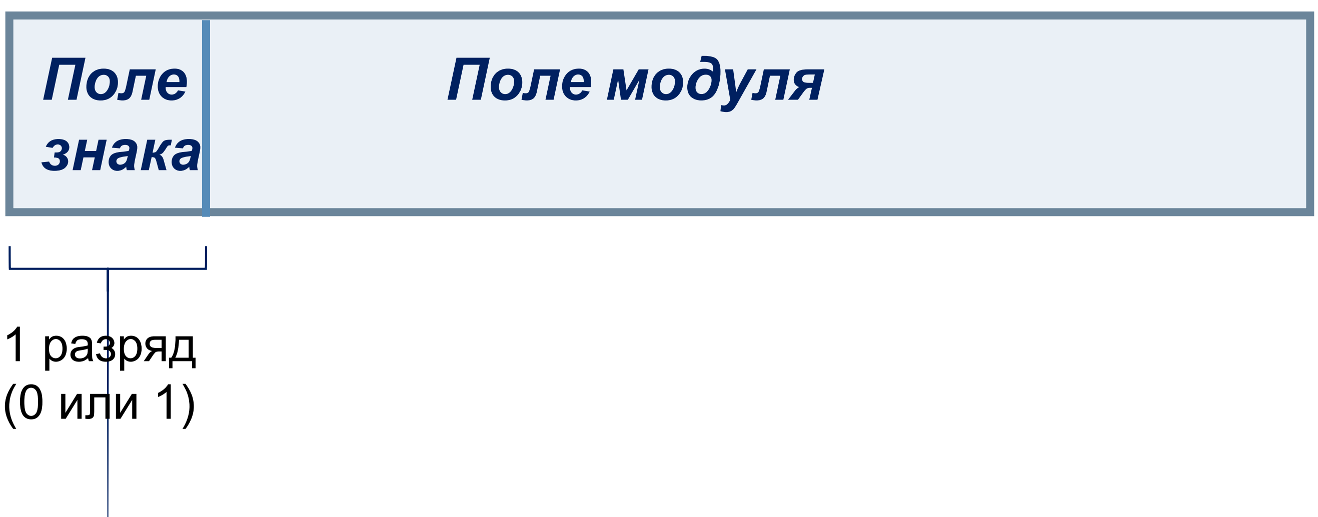
При делении столбиком в *P*-ичной системе счисления приходится в качестве промежуточных вычислений выполнять действия умножения и вычитания, *следовательно*, используя таблицы умножения и сложения.

$$
\begin{array}{r|l}
10_3 & 2_3 \\ \cline{2-2}
-\underline{\;2\;} & 1,(1)_3 \\
10 & \\
-\underline{\;2\;} & \\
1 &
\end{array}
$$

Арифметика с алгебраическими числами

Способы представления целых чисел

- Прямой код
- Обратный код
- Дополнительный код



Прямой код

- Число записывается «как есть», дополняется нулями в старших разрядах до нужного размера.

$$00101101 \qquad 45_{(10)}$$

Запись оваться по
прави

$$[A]_{пк} = \begin{cases} 0.\ A, \text{ если } A >= 0; \\ 1.\ |A|, \text{ если } A < 0. \end{cases}$$

Обратный код

- Положительное число записывается «как есть», дополняется нулями в старших разрядах до нужного размера.
- Отрицательное число записывается **инвертированием** разрядов модуля числа.
- Старший бит определяет знак числа (1 — отрицательное, 0 — положительное).

Запись целого числа в обратном коде

$$[A]_{\text{ок}} = \begin{cases} 0.\,A, \text{ если } A \geq 0; \\ 1.\,((q^n - 1) + A), \text{ если } A < 0, \end{cases}$$

где **n** – разрядность модульного поля;

q – основание системы счисления;

(2*n* – 1) – максимальная включенная граница диапазона изменения представляемых чисел.

диапазон изменения чисел *A*: $(q^n - 1) \geq |A| \geq 0$

Запись правильной дроби в обратном коде

$$[A]_{\text{ок}} = \begin{cases} 0.\,A, \text{ если } A \ge 0; \\ 1.\,((1 - q^{-n}) + A), \text{ если } A < 0, \end{cases}$$

где **n** – разрядность поля модуля;

$1 – q^{-n}$ – верхняя включенная граница представляемых чисел.

$$(1 - q^{-n}) \ge |\,A\,| \ge 0$$

Примеры чисел в обратном коде

01100111	$103_{(10)}$
10011000	$-103_{(10)}$
00000000	$0_{(10)}$
11111111	$-0_{(10)}$

Дополнительный код

- Положительное число записывается «как есть», дополняется нулями в старших разрядах до нужного размера.
- Отрицательное число записывается **инвертированием** разрядов модуля числа **и прибавлением 1**.
- Старший бит определяет знак числа (1 — отрицательное, 0 — положительное).

Запись целого числа в дополнительном коде:

Запись числа формируется по следующему правилу:

$$[A]_{ДК} = \begin{cases} 0.\ A, \text{ если } A >= 0; \\ 1.\ q^n + A, \text{ если } A < 0, \end{cases}$$

где **n** – разрядность модульного поля,

q – основание с/с,

q^n – максимальная невключенная граница диапазона изменения чисел.

$$(q^n) > |A| \geq 0;$$

Запись правильной дроби в дополнительном коде:

$$[A]_{\text{дк}} = \begin{cases} 0.\,A, \text{ если } A >= 0; \\ 1.\,(1 + A\,), \text{ если } A < 0, \end{cases}$$

где 1 – максимальная невключенная граница диапазона изменения представляемых чисел.

$$1 > |A| \geq 0$$

Примеры чисел в дополнительном коде

$$42 = \mathit{00101010}_{(2)}$$

$$-42 =$$

00101010

11010101

$\boldsymbol{\mathit{11010110}}_{(2)}$

Примеры чисел в дополнительном коде

-1	00000001
	11111110
	$11111111_{(2)}$
0	$00000000_{(2)}$

Перевод отрицательного числа из обратного (*или дополнительного*) кода в прямой выполняется по тем же правилам, что и перевод числа из прямого кода в обратный (*или дополнительный)*:

- для перевода отрицательного числа из обратного в *прямой код* необходимо дополнить его модуль до включенной границы;
- для перевода отрицательного числа из дополнительного кода в *прямой код* необходимо дополнить его модуль до

Правило формирования модуля обратного кода отрицательного двоичного числа

Для формирования модульной части записи отрицательного числа в обратном коде достаточно в модульной части записи этого числа в прямом коде взять обратные значения всех двоичных разрядов, т. е. необходимо проинвертировать модуль прямого кода.

Правило формирования модуля дополнительного кода отрицательного числа

Для формирования модульной части записи отрицательного числа в дополнительном коде достаточно в модульной части записи этого числа в прямом коде взять обратные значения всех двоичных разрядов, т. е. необходимо проинвертировать модуль прямого кода и к полученному коду прибавить 1 в младший разряд.

При выполнении операций *над числами со знаком* в ЭВМ используется одно из его представлений (прямой, дополнительный или обратный код).

Как правило, информация в памяти ЭВМ хранится в прямом коде, а при выполнении операций применяется обратный или дополнительный код.

При использовании дополнительного кода (или обратного) операции вычитания заменяются операциями сложения с изменением знака одного

Операции в дополнительном коде

- При сложении чисел, представленных в дополнительном коде, выполняется сложение разрядов, представляющих запись операндов, по правилам двоичной арифметики по всей длине записи чисел, не обращая внимания на границу, которая отделает знаковое поле от модульного.
- *Переполнение знакового поля игнорируется (т.е. выход за левую границу).*
- В результате такого сложения будет получен дополнительный код суммы

Операции в обратном коде

- При сложении чисел, представленных в обратном коде, выполняется сложение разрядов, представляющих запись операндов, по правилам двоичной арифметики по всей длине записи чисел, не обращая внимания на границу, которая отделает знаковое поле от модульного.
- *Переполнение знакового поля (т.е. выход за левую границу) учитывается как +1 к младшему разряду полученной суммы.*
- В результате такого сложения будет получен обратный код суммы операндов

Примеры:

Рассчитать значение С = A-B, если А= 10, В=3.

Десятичная запись

$$\begin{array}{r} +\; 10 \\ -3 \\ \hline 7 \end{array}$$

Двоичные коды

$$\begin{array}{rl} +\; 0\,0001010 & \\ 1\,1111101 & \text{Дополнительный код числа -3} \\ \hline 0\,0000111 & \end{array}$$

перенос отбрасывается

Двоичные коды

$$\begin{array}{rl} +\; 0\,0001010 & \\ 1\,1111100 & \text{Обратный код числа -3} \\ \hline 0\,0000110 & \\ +1 & \\ \hline 0\,0000111 & \end{array}$$

Примеры:

6 – 4 = ?

6 - положительное число с кодом 0110

–4 - отрицательное число с дополнительным кодом 1100

$$\begin{array}{r} 0110 \\ +\,1100 \\ \hline 10010 \end{array}$$

(перенос игнорируется): **6 – 4 = 2**

-5 + 2 = ?

2 - положительное число с кодом 0010

–5 - отрицательное число с дополнительным кодом 1011

$$\begin{array}{r} 1011 \\ +\,0010 \\ \hline 1101 \end{array}$$

Число с кодом 1101 является отрицательным, модуль этого числа имеет код $\mathbf{0011_2 = 3_{10}}$

Модифицированные коды

При расчете разрядности *n* модульного поля весьма трудно бывает учесть диапазон значений результатов, особенно когда последовательность операций, представленных в подлежащих реализации выражениях, достаточно сложна.

- *Расчёт, выполненный по всем формальным правилам, может дать «абсурдный» результат, так ,например, результат может отличаться по знаку от операндов при одинаковых знаках обоих операндов*

Модифицированные коды

Более просто ситуация переполнения определяется при применении *модифицированного* кода (обратного или дополнительного).

Модифицированные коды отличаются от базовых кодов только тем, что поле знака операндов имеет два разряда, и эти разряды имеют одинаковые значения:

00 – для положительных чисел;

11 – для отрицательных чисел.

Переполнение модифицированного кода

Если в результате сложения чисел в модифицированном коде полученный результат имеет в поле знака одинаковые значения в обоих разрядах (00 или 11), **то *переполнения нет***,
если же разряды знакового поля имеют неодинаковые значения (10 или 01), **то *имеет мест переполнение*.**

Переполнение модифицированного кода

При этом, если в поле знака имеет место значение **01** – результат *положительный* (фиксируется *положительное переполнение*), а если **10**, то полученный результат – *отрицательный* (фиксируется *отрицательное переполнение*).

Основным носителем знака числа является
левый разряд знакового поля.

Логическая схема выявления переполнения

Логические операции с двоичными кодами

- Логическое суммирование
- Логическое умножение
- Логическое отрицание
- Суммирование по модулю 2
- Операции сдвига

Логическое суммирование

Обозначения: **ИЛИ, OR, «V»**

Выполняется над двумя кодами заданной разрядности и генерирует код той же разрядности.

A	B	A V B
0	0	0
1	0	1
0	1	1
1	1	1

Пример:

10001101 V 11110000 = **?**

= $11111101_{(2)}$

Логическое умножение

Обозначения: **И, AND, «^»**

Выполняется над двумя кодами заданной разрядности и генерирует код той же разрядности.

A	B	A ^ B
0	0	0
1	0	0
0	1	0
1	1	1

Пример:

10001101 **^** 11110000 = **?**

$= 10000000_{(2)}$

Суммирование по модулю 2

Обозначения: **mod 2, «⊕»**

Выполняется над двумя кодами заданной разрядности и генерирует код той же разрядности.

A	B	A ⊕ B
0	0	0
1	0	1
0	1	1
1	1	0

Пример:

$10001101 \oplus 11110000 = ?$

$= 01111101_{(2)}$

«Исключающее ИЛИ»

Логическое отрицание

Обозначения: **НЕТ, NOT, «¬», ‾, «x»**

Выполняется над одним кодом и генерирует результирующий код той же разрядности.

A	NOT A
0	1
1	0

Пример:

¬(10001101) = NOT(10001101) = $\overline{(10001101)}$ = **?**

= $01110010_{(2)}$

Операции сдвига

- Логические сдвиги
  - Сдвиг вправо
  - Сдвиг влево
  - Циклический сдвиг вправо
  - Циклический сдвиг влево
- Арифметический сдвиг
  - Вправо
  - Влево

Логические сдвиги

- Сдвиг влево выполняется за счет смещения значений разрядов от младшего к старшему, в самом младшем устанавливается 0, а «выталкиваемый» разряд пропадает.
- Сдвиг вправо выполняется за счет смещения значений разрядов от старшего к младшему, самый старший разряд устанавливается в 0, а «выталкиваемый» разряд пропадает.

Примеры:

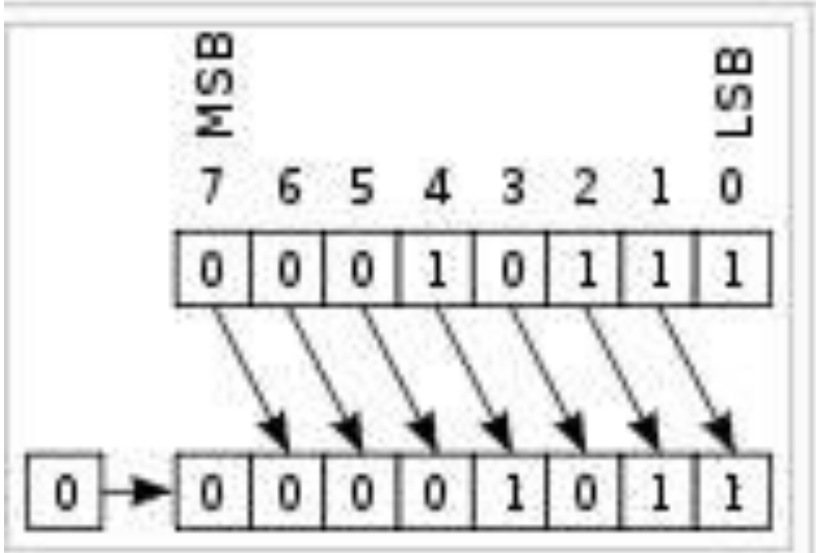
- Код **11001110** после сдвига влево равен **10011100**
- Код **11001110** после сдвига вправо равен **01100111**

Логические сдвиги

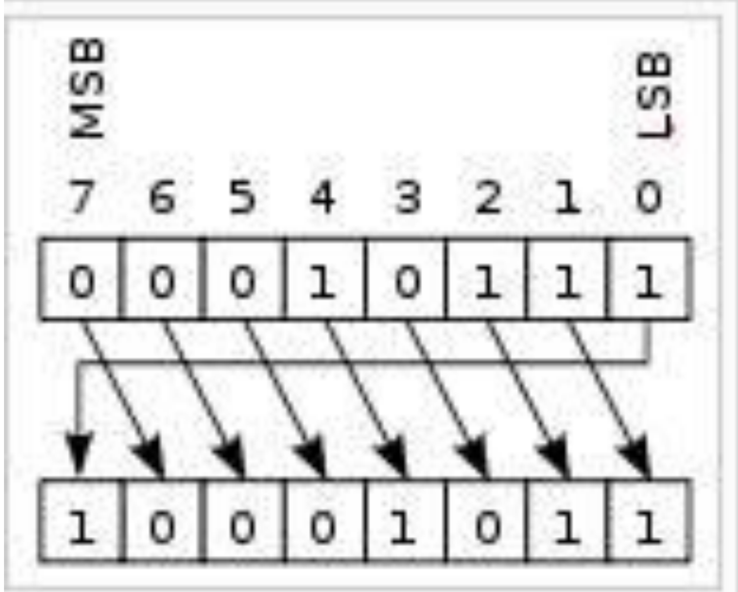
- Циклический сдвиг влево выполняется за счет смещения значений разрядов от младшего к старшему, в самом младшем разряде устанавливается значение «выталкиваемого» разряда.
- Сдвиг вправо выполняется за счет смещения значений разрядов от старшего к младшему, в самый старший разряд устанавливается «выталкиваемый» младший разряд.

Примеры:

- Код **11001110** после цикл. сдвига влево равен **10011101**
- Код **11001110** после цикл. сдвига вправо равен **01100111**



Логический сдвиг



Циклический сдвиг

Арифметический сдвиг

Арифметические сдвиги обеспечивают выполнение *умножения* (сдвиги влево) или операции *деления* (сдвиги вправо) двоичных кодов ***на два***.

Аналогично, сдвиги десятичного числа обеспечивают выполнение *деления* *или* *умножение* **на 10**.

Арифметические сдвиги влево и вправо *реализуются* ***по-разному*** в зависимости как **от знака числа**, так и **от используемого кода** (прямого обратного, дополнительного).

Арифметические **сдвиги влево** положительных двоичных чисел выполняются *независимо* от используемого кода.

Арифметические **сдвиги влево** двоичного ***прямого*** кода выполняются в зависимости от того, какое сдвигается число – *положительное или отрицательное.*

Если *сдвигается* ***положительное число,*** то сдвиг (вправо или влево) выполняется как соответствующий логический сдвиг (влево или вправо), с той лишь разницей, что при сдвиге влево предусматриваются **средства определения факта переполнения**.

При любом **сдвиге вправо** необходимо предусмотреть ***средства для округления*** после завершения нужного количества сдвигов и ***средства обнаружения обнуления***

Примеры:
Найти результат арифметического сдвига …

- **…** влево на **три** разряда двоичного прямого кода числа
  [*A*]$_{\text{пк}}$ = 00.00000101

первый сдвиг: **00.00000101** ← 00.00001010

второй сдвиг: 00.00001010 ← 00.00010100

третий сдвиг: 00.00010100 ← **00.00101000**

- **…** влево на **четыре** разряда двоичного прямого кода числа
  [*A*]$_{\text{пк}}$ = 00.00101

первый сдвиг**: 00.00101** ← 00.01010

второй сдвиг: 00.01010 ← 00.10100

третий сдвиг: 00.10100 ← **01.01000**

четвертый сдвиг ???

разные значения – это сигнал переполнения !!!

Примеры: Найти результат арифметического сдвига ...

- ... вправо на **два** разряда двоичного прямого кода числа **$[A]_{пк}$ = 00.00000110**

первый сдвиг: **00. 00000110** → 00. 00000011

второй сдвиг: 00. 00000011 → **00. 00000001** ***1***

«вытолкнутый» разряд

Результат выполнения заданного сдвига будет равен **00.00000010**

Примеры: Найти результат арифметического сдвига ...

- ... вправо на **четыре** разряда двоичного прямого кода числа **$[A]_{пк}$ = 00.00000110**

первый сдвиг: **00.00000110** → 00.00000011

второй сдвиг: 00.00000011 → 00.00000001

третий сдвиг: 00.00000001 → **00.00000000**

сигнал о получении нулевого результата

Оставшиеся сдвиги могут не

Арифметические сдвиги отрицательных двоичных чисел

- ***Арифметический сдвиг вправо*** может выполняться над отрицательными числами с ***переполнением***
(в *модифицированном* прямом, обратном или дополнительном коде такие числа имеют в знаковом поле ***10***).
  - После сдвига в знаковом поле будет ***11***.
  - В старшем разряде устанавливается **единица**, если число представлено в прямом коде.
  - В старшем разряде устанавливается **ноль**, если число представлено в обратном или

Арифметические сдвиги отрицательных двоичных чисел в прямом коде

Арифметический сдвиг числа, *в прямом коде*, осуществляется как соответствующий сдвиг **только модульного поля** записи числа.

Арифметические сдвиги **отрицательных** двоичных чисел в **обратном коде**

- **Арифметический сдвиг ВЛЕВО – это** **циклический сдвиг** исходного кода с контролем за переполнением.
- **Арифметический сдвиг ВПРАВО – это** **сдвиг только модульной части** записи числа с установкой единицы в освобождающийся разряд с контролем за обнулением результата сдвига *(появление единичных значений во всех разрядах)* и округление результата (ПОСЛЕ выполнения заданного количества сдвигов). Освобождающийся разряд заполняется

Арифметические сдвиги **отрицательных** двоичных чисел в **дополнительном коде**

- **Арифметический сдвиг *ВЛЕВО – это*** ***логический сдвиг влево модуля*** исходного кода
  с контролем за переполнением.
  Освобождающийся разряд заполняется ***нулем***.

- **Арифметический сдвиг *ВПРАВО – это*** **логический сдвиг вправо модуля** исходного числа
  с контролем за обнулением результата сдвига *(появление единичных значений во всех разрядах)*

Примеры:

Пример 1.

Выполнить сдвиг вправо на 2 разряда числа $[A]_{пк}$ = 10.01000110 (A_{10} = –326).

Первый сдвиг: 10. 01000110 → 11.10100011 (–163_{10});

Второй сдвиг: 11.10100011 → 11.01010001 (–81_{10}) и последний вытолкнутый разряд равен 1).

С учетом округления имеем окончательный результат: **$[A2]_{пк}$ = 11.01010010.**

Пример 2.

Выполнить сдвиг вправо на 2 разряда числа $[A]_{ок}$ = 10.10111001 (A_{10} = –326).

Первый сдвиг: 10.10111001 → 11.01011100 (–163_{10});

Второй сдвиг: 11.01011100 → **11.10101110** *(–82_{10}).*

Пример 3.

Выполнить сдвиг вправо на 2 разряда число $[A]_{дк}$ = 10. 10111010 (A_{10} = –326).

Первый сдвиг: 10.10111010 → 11.01011101 (–163_{10});

Второй сдвиг: 11.01011101 → 11.10101110 (–81_{10}) и последний вытолкнутый

Литература для самостоятельной работы

- Гашков С.Б. **Системы счисления и их применение**. Серия: Библиотека «Математическое просвещение». // М.: МЦНМО, 2004. - 52 с.: ил.

- Фомин С. В. **Системы счисления.** Серия «Популярные лекции по математике», выпуск 40. // М.: Наука, 1987. - 48 с.