V. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

ТЕМА 5.1 *СЛОЖЕНИЕ ДВОИЧНЫХ ЧИСЕЛ С ФИКСИРОВАННОЙ ТОЧКОЙ*

- Правила сложения

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = (1)0 где (1) перенос

перенос		1	1		
A=6	0	0	1	1	0
B=3	0	0	0	1	1
A+B	0	1	0	0	1

- Правила вычитания

0 - 0 = 0

0 - 1 = (1)1, где (1) единица заема в старшем разряде.

1 - 0 = 1

1 - 1 = 0

заем		0	1		
A=9	0	~~1~~	~~0~~	0	1
B=7	0	0	1	1	1
A-B	0	0	0	1	0

СЛОЖЕНИЕ В ПРЯМОМ КОДЕ

- Правила сложения двоичных чисел в прямом коде.
- Если оба слагаемых (операнда) имеют одинаковые знаки, то следует сложить разряды цифровой части, а сумме приписать знак одного из операндов.
- Если знаки операндов разные, то из разрядов цифровой части большего по модулю числа вычитается меньшее, а сумме приписывается знак большего числа.

перенос		1	1		
A=6	0	0	1	1	0
B=3	0	0	0	1	1
A+B	0	1	0	0	1

заем		0	1		
A=9	0	~~1~~	~~0~~	0	1
B=7	0	0	1	1	1
A-B	0	0	0	1	0

A=3
B=-14
A+B

заем			0	0		
B=-14	1	1	~~1~~	~~1~~	0	
A=3	0	0	0	1	1	вычитание 14-10
|B|-|A|	1	1	0	1	1	

СЛОЖЕНИЕ В ОБРАТНОМ КОДЕ

Операцию вычитания двух чисел желательно заменить сложением соответствующих чисел:

$$A\text{-}B=A+(-B).$$

Нетрудно заметить, что замена вычитания сложением возможна в случае представления отрицательного числа в обратном коде. Используя представление чисел в обратном коде, операцию вычитания A-B выполняют путем сложения числа A с обратным кодом числа B, то есть

$$A\text{-}B=A+B_{ок}$$

При этом знаковый разряд и цифровая часть операнда обрабатываются как единое целое. Правильный знак суммы получается автоматически в процессе суммирования цифр знаковых разрядов операндов и единицы переноса из цифровой части, если она появляется.

- Характерной особенностью сложения в обратном коде является наличие циклического переноса (если он возникает) из знакового разряда в младший разряд цифровой части, в результате чего получается правильная сумма.

ПРИМЕРЫ

A=46, B=73   A+B

		перенос				1				
	46		0.	0	1	0	1	1	1	0
	73		0.	1	0	0	1	0	0	1
1	19		0.	1	1	1	0	1	1	1

A=-76, B=100 A+B

A=-76
$A_{пк}$= 1.1001100
$A_{ок}$= 1.0110011

		1	1	1							
	-76		1.	0	1	1	0	0	1	1	
1	00		0.	1	1	0	0	1	0	0	
	24	1	0.	0	0	1	0	1	1	1	
										1	
			0.	0	0	1	1	0	0	0	
						16	8				24

СЛОЖЕНИЕ В ДОПОЛНИТЕЛЬНОМ КОДЕ

В дополнительном коде, как и в обратном коде, операция вычитания заменяется операцией алгебраического сложения. При этом знаковый разряд и цифровая часть операнда обрабатываются так же, как единое целое. Правильный знак суммы получается автоматически в процессе суммирования цифр знаковых разрядов операндов и единицы переноса из цифровой части, если она появляется.

$$A-B=A+(-B).$$
$$A-B=A+B_{ДК}.$$

При сложении двух чисел в дополнительном коде знаковый разряд и цифровая часть обрабатываются как единое целое. Результат сложения получается в дополнительном коде. Если результат суммирования лежит в диапазоне, определяемом разрядностью обрабатываемых операндов, то полученный результат верный.

ПРИМЕРЫ

A=+0110100 ($52_{(10)}$)
B=+1000111 ($71_{(10)}$).
A+B=52+71=123

A=	0.	0	1	1	0	1	0	0	
B=	0.	1	0	0	0	1	1	1	
Z=	0.	1	1	1	1	0	1	1	
		64	32	16	8		2	1	123

A= -1001100 ($-76_{(10)}$)
B=+1111111 ($127_{(10)}$).
A+B=-76+127=51

1	1	1	1	1	1				
Адк=	1.	0	1	1	0	1	0	0	
B=	0.	1	1	1	1	1	1	1	
	0.	0	1	1	0	0	1	1	
			32	16			2	1	51

A= +0101010 ($42_{(10)}$)
B= +1110000 ($112_{(10)}$).
A-B.
A-B=A+(-B)=42+(-112)=-70

-B= -112= -1110000
$-B_{ПК}$=1.1110000
$-B_{ДК}$=1.0010000

A=	0.	0	1	0	1	0	1	0	
-Вдк	1.	0	0	1	0	0	0	0	
$Z_{дк}$=	1.	0	1	1	1	0	1	0	
инверсия	1.	1	0	0	0	1	0	1	
+1								1	
Zпк=	1.	1	0	0	0	1	1	0	
	sg=-	64				4	2		-70

A= -0101111 ($-47_{(10)}$)
B= 1001011 ($75_{(10)}$).
A-B.
A-B=A+(-B)=-47+(-75)=-122

$A_{ДК}$=1.1010001

-B= -75= -1001011
$-B_{ПК}$=1. 1001011
$-B_{ДК}$=1.0110101

1	1	1	1				1		
Адк=	1.	1	0	1	0	0	0	1	
-Вдк	1.	0	1	1	0	1	0	1	
$Z_{дк}$=	1.	0	0	0	0	1	1	0	
инверсия	1.	1	1	1	1	0	0	1	
+1								1	
Zпк=	1.	1	1	1	1	0	1	0	
	sg=-	64	32	16	8		2		-122

ПЕРЕПОЛНЕНИЕ РАЗРЯДНОЙ СЕТКИ

При сложении двух чисел одинакового знака возможно переполнение разрядной сетки

A=100
B=120
A+B
A+B=100+120=220 (>127)

	1	1						
A=	0.	1	1	0	0	1	0	0
B=	0.	1	1	1	1	0	0	0
Z=	1.	1	0	1	1	1	0	0

положительное переполнение.

A=-100
B=-120
A+B
A+B=-100+(-120)=-220 (<-128)

1			1	1	1			
Адк=	1.	0	0	1	1	1	0	0
Вдк	1.	0	0	0	0	1	0	0
Zдк=	0.	0	1	0	0	0	0	0

отрицательное переполнение.

В случае сложения двух положительных чисел, переполнение разрядной сетки сопровождается появлением переноса в знаковый разряд суммы при отсутствии переноса из знакового разряда. В этом случае имеет место **положительное переполнение**.

В случае сложения двух отрицательных чисел, переполнение разрядной сетки сопровождается появлением переноса из знакового разряда суммы при отсутствии переноса в знаковый разряд. В этом случае имеет место **отрицательное переполнение**.

МОДИФИЦИРОВАННЫЙ ДОПОЛНИТЕЛЬНЫЙ ИЛИ ОБРАТНЫЙ КОДЫ

	1	1						
A=	0 0.	1	1	0	0	1	0	0
B=	0 0.	1	1	1	1	0	0	0
Z=	0 1.	1	0	1	1	1	0	0
	1		1	1	1			
Acc=	1 1.	0	0	1	1	1	0	0
Bcc	1 1.	0	0	0	0	1	0	0
Zcc=	1 0.	0	1	0	0	0	0	0

Представление числа, в котором используется два знаковых разряда, называется модифицированным. Модифицированный код положительного числа содержит в поле знака два нуля. В случае отрицательных чисел, в поле знака присутствуют две единицы. Признаком переполнения разрядной сетки, при использовании модифицированного кода, является наличие разных цифр в знаковых разрядах результата.

СДВИГ ДВОИЧНЫХ ЧИСЕЛ

Операции сдвига могут быть:
логического,
арифметического
циклического типов.

Двоичное число может быть сдвинуто влево или вправо.

Сдвиг влево на один разряд соответствует умножению на 2, а сдвиг вправо на один разряд соответствует делению на 2.

Сдвиг двоичных чисел необходим при выполнении таких операций, как умножение и деление, сложение чисел с плавающей точкой.

ЛОГИЧЕСКИЙ СДВИГ

A=00100110 ($38_{(10)}$)
SHL(A)=01001100 ($76_{(10)}$) – логический сдвиг влево;

A=00100110 ($38_{(10)}$)
SHR(A)=00010011 ($19_{(10)}$) – логический сдвиг вправо

АРИФМЕТИЧЕСКИЙ СДВИГ

В случае арифметического сдвига цифра знака сохраняется. При сдвиге влево теряется цифра следующая за знаковой, а в случае сдвига вправо цифра знака дублируется и теряется цифра в младшем разряде

$$\left.\begin{array}{l} Acc=00010101\ \ (21_{(10)}),\ \ ASL(Acc)=00101010\ \ (42_{(10)}); \\ Acc=11011111\ \ (-33_{(10)}),\ \ ASL(Acc)=10111110\ \ (-66_{(10)}). \end{array}\right\}\ \text{арифметический сдвиг влево;}$$

$$\left.\begin{array}{l} Acc=00010101\ \ (21_{(10)}),\ \ ASR(Acc)=00001010\ \ (10_{(10)}); \\ Acc=11011111\ \ (-33_{(10)}),\ \ ASR(Acc)=11101111\ \ (-17_{(10)}). \end{array}\right\}\ \text{арифметический сдвиг вправо.}$$

ЦИКЛИЧЕСКИЙ СДВИГ

În acest caz nu se pierde nici un bit din numărul deplasat.

A=11010010

ROL(A)=10100101 –циклический сдвиг влево;

A=11010010

ROR(A)=01101001 – циклический сдвиг вправо.