Тема 5.1 *Сложение двоичных чисел с фиксированной точкой*

Операция сложения двоичных кодовых слов является ключевой операцией в любой цифровой системе. В двоичной системе счисления сложение двоичных цифр выполняется согласно следующим правилам:

0+0=0,

0+1=1,

1+0=1.

1+1=(1)0 перенос

При сложении двух двоичных цифр 1 имеем: 1+1=0 плюс единица переноса, которую следует учитывать при сложении двоичных цифр в следующем разряде. Проиллюстрируем сложение двух двоичных слов:

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

Сложение двоичных чисел, представленных в форме с фиксированной точкой, можно выполнять в прямом, дополнительном и обратном кодах. Рассмотрим особенности сложения двоичных чисел в каждом из трех кодов.

Сложение в прямом коде

Правила сложения двоичных чисел в прямом коде не отличаются от вышеуказанных. 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.

При вычитании двоичных цифр следует соблюдать следующие правила:

0-0=0,

1-0=1,

1-1=0,

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

Примеры

A=6
B=3
A+B

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

A=9
B=7
A-B

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	

Таким образом, для сложения чисел в прямом коде характерны два недостатка: 1) знаковый и цифровые разряды операндов приходиться обрабатывать раздельно; 2) выполнение операции сложения затруднено тем, что необходимо иметь в составе вычислительного оборудования как суммирующее устройство, так и вычитающее устройство. Из-за указанных недостатков прямой код для выполнения операции сложения не используется, однако этот код удобен при реализации операции умножения.

Сложение в обратном коде.

Из вышесказанного следует, что 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. Докажем это на следующих примерах.

Пусть заданы два целых числа со знаком A и B. Будем полагать, что их значения находятся в допустимом диапазоне представления 8-разрядных двоичных операндов. Кроме того, будем считать что результат суммирования лежит в соответствующем диапазоне.

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

A=   00110100

+

B=   01000111

$A+B=\quad 01111011$

Сумма получена в дополнительном коде и положительна (цифра знака 0). Полученный результат верный, ибо $01111011_{(2)}=123_{(10)}$.

Пусть теперь A= -1001100 ($-76_{(10)}$) и B=+1111111 ($127_{(10)}$). Представляя исходные числа в дополнительном коде, получаем:

$$\begin{array}{ll} A_{дк}= & 10110100 \\ & \qquad\qquad + \\ B= & 01111111 \\ & \\ A_{дк}+B= & \mathbf{1}00110011=(A+B)_{дк} \end{array}$$

Если числа A, B имеют разные знаки, то их сумма всегда принадлежит диапазону допустимых значений. Следовательно, если имеет место перенос из знакового разряда, то цифра переноса игнорируется. Полученная сумма верна и положительна: $00110011_{(2)}=51_{(10)}$.

Пусть далее A=+0101010 ($42_{(10)}$) и B= -1110000 ($-112_{(10)}$). Представляя числа в дополнительном коде, получаем:

$$\begin{array}{ll} A= & 00101010 \\ & \qquad\qquad + \\ B_{дк}= & 10010000 \\ & \\ A+B_{дк}= & 10111010=(A+B)_{дк} \end{array}$$

Результат отрицательный и представлен в дополнительном коде. Для изображения полученной суммы в десятичном виде поступим следующим образом:

исходное число	: 10111010
инвертирование	: 01000101
прибавление 1	: 01000110
абсолютное значение	: $70_{(10)}$
десятичное представление исходного числа	: -70

Итак, полученная сумма верна.

Пусть необходимо сложить два отрицательных числа: A= -0101111 ($-47_{(10)}$) и B= -1001011 ($-75_{(10)}$). Представляя исходные операнды в дополнительном коде и суммируя, получаем:

$$\begin{array}{ll} A_{дк}= & 11010001 \\ & \qquad\qquad + \\ B_{дк}= & 10110101 \end{array}$$

$A_{дк}+B_{дк}=$ $\mathbf{1}10000110=(A+B)_{дк}$

Так как сумма чисел A и B лежит в диапазоне допустимых значений, перенос из знакового разряда игнорируется. Десятичное представление полученной суммы есть:

исходное число	: 10000110
инвертирование	: 01111001
прибавление 1	: 01111010
абсолютное значение	: $122_{(10)}$
десятичное представление исходного числа	: -122

Полученная в дополнительном коде сумма верна.

Переполнение разрядной сетки

В приведенных примерах соблюдалось условие попадания суммы чисел A и B в интервал допустимых значений. Это условие, как было отмечено выше, всегда соблюдается при сложении двух чисел различного знака. При сложении двух чисел одинакового знака возможно нарушение указанного условия.

Приведем следующий пример. Пусть заданы положительные целые числа A= +1011000 ($88_{(10)}$) и B= +0110010 ($50_{(10)}$). Суммируя дополнительные коды заданных чисел, имеем:

$$
\begin{array}{rl}
A= & 01011000 \\
 & \qquad\qquad + \\
B= & 00110010 \\
 & \\
A+B= & \underline{1}0001010\neq(A+B)
\end{array}
$$

Полученный результат не совпадает с дополнительным кодом суммы чисел A и B, так как имеет знак (отрицательный в данном случае) отличный от знака просуммированных чисел. В таких ситуациях говорят что произошло переполнение разрядной сетки. В приведенном примере A+B=$138_{(10)}$, что превышает максимальное значение равное $127_{(10)}$.

Приведем другой пример. Пусть A= -0111100 ($-60_{(10)}$) и B= -1000101 ($-69_{(10)}$). Суммируя дополнительные коды, получаем:

$$
\begin{array}{rl}
A_{дк}= & 11000100 \\
 & \qquad\qquad + \\
B_{дк}= & 10111011 \\
 & \\
A_{дк}+B_{дк}= & 1\quad \underline{0}1111111\neq (A+B)_{дк}
\end{array}
$$

Замечаем, что, как и в предыдущем примере, полученная сумма имеет знак (положительный в данном случае) отличный от знака просуммированных чисел. И в этом случае произошло переполнение разрядной сетки ($A+B=-129_{(10)}$, но в случае 8-разрядного дополнительного кода минимальное значение равно - $128_{(10)}$).

Из последних двух примеров следует, что при сложении двух чисел одинакового знака, представленных в дополнительном коде, признаком переполнения является отличие знака полученной суммы от знаков слагаемых. 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**.

Для обнаружения переполнения разрядной сетки при алгебраическом сложении двоичных чисел в дополнительном и обратном кодах можно использовать **модифицированный дополнительный или обратный коды**. 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

При выполнении в цифровых устройствах определенных операций, может возникнуть необходимость сдвигать цифры двоичного кода. Операции сдвига могут быть логического, арифметического и 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.

Приведем примеры выполнения операций сдвига влево/вправо на один разряд.

- Логический сдвиг. Пусть задано двоичное число A=00100110 ($38_{(10)}$).

SHL(A)=01001100 ($76_{(10)}$) – логический сдвиг влево;
SHR(A)=00010011 ($19_{(10)}$) – логический сдвиг вправо.

Замечание: В случае логического сдвига влево теряется цифра в левом (старшем) разряде, а в случае логического сдвига вправо теряется цифра в младшем разряде.

- 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. Ниже даны примеры арифметического сдвига чисел, представленных в дополнительном коде:
-

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

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

Заметим, что арифметический сдвиг можно выполнять в модифицированном дополнительном или обратном кодах.

- Циклический сдвиг. При таком типе сдвига не теряется ни одна цифра сдвигаемого числа. Это объясняется наличием кольцевой связи между старшим и младшим разрядами числа. Пусть задано число A=11010010.

ROL(A)=10100101 –циклический сдвиг влево;
ROR(A)=01101001 –циклический сдвиг вправо.