Тема 7. Двійкова арифметика і додатковий код.

Двоичная арифметика

Сложение, вычитание или умножение двоичных чисел выполняются так же, как и в арифметике десятичных чисел. Большинство микропроцессоров владеет командами сложения и вычитания двоичных чисел, однако некоторые, менее многочисленные выполняют команды умножения и деления (например, микропроцессоры Intel 8086 и Intel 8088).

На рис. 7.1, а представлены простые правила двоичного сложения. Два первых (слева) правила очевидны, третье показывает, что 1 + 1 = 10, т. е. наиболее значимая 1 пере носится в ближайший старший разряд. Четвертое правило, наконец, показывает, что 1 + 1 + 1 = 11. В этом случае пер вое, второе слагаемые и запоминаемое в результате сложения в младшем разряде число — все 1. Результатом является сумма — 1 с переносом 1.

Рис. 7.1. Двоичное сложение: а — правила: б — пример

Сложим двоичные числа 0011 1011 и 0010 1010 (операция показана на рис. 6.1,б). Для большей ясности действия с десятичными эквивалентами обрабатываемых чисел показаны на рисунке справа. Суммой двух чисел 0011 1011 и 0010 1010 будет 0110 0101_2.

На рис. 6.2, а приведены правила двоичного вычитания. Первые три аналогичны десятичному вычитанию. Последнее требует заема из более значимого предшествующего разряда (в этом случае вес 2). Уменьшаемым является двоичное число 10, вычитаемым 1, разностью—1.

Вычтем двоичное число 0011 1001 из 0101 0101. Этот пример приведен на рис. 7.2, б. Разряды весов 1, 2 и 4 это го двоичного вычитания просты для выполнения и относятся к первым трем правилам на рис. 7.2, а. В колонке веса 8 имеет место вычитание 1 из 0. Тогда 1 занимается из колонки веса 16. Единица вычитается из 10_2, что дает разность 1 согласно четвертому правилу на рис. 7.2, а. После этого заема в колонке веса 16 имеет место вычитание 1 из нового вычитаемого 0. Согласно четвертому правилу 1 должна быть занята из следующей, более значимой позиции (колонка веса 32), но в колонке 32 имеем 0; поэтому колонка 32 должна сделать заем из колонки веса 64, что и вы полнено. Окончательно колонка 16 делает заем из колонки 32, уменьшаемым в колонке 16 становится 10_2, вычитаемым 1, разностью 1. В колонке 32 имеем 1 - 1 = 0, в колонке 64 — 0 - 0 = 0, в колонке 128 — 0 - 0 = 0. Таким образом, рис. 7.2, б иллюстрирует операцию вычитания 0011 1001_2цз 0101 0101_2 (справа эта задача решена в десятичной записи) .

				0 / 10	1 10 / 0 10 0 10	
Уменьшаемое	0	1	1	10	01010101	85
Вычитаемое	0	0	1	1	00111001	57
Разность	0	1	0	1	00011100_2	28_{10}

а) б)

Рис. 7.2. Двоичное вычитание: *а* — правила; *б* — пример

Приведем правила десятичного умножения:

Множимые	0	1	0	1
	×	×	×	×
Множители	0	0	1	1
Произведения	0	0	0	1

Два первых правила не требуют никаких пояснений. В двух следующих множителем является 1: когда множителем является 1 при двоичном умножении, *множимое становится результатом и представляет собой произведение.* Когда множитель 0, произведение всегда 0.

Выполним умножение 1101 на 101. Как и в случае умножения десятичных чисел, множимое сначала умножается на число, стоящее в младшем разряде (в рассматриваемом случае — бит в колонке веса 1).

Множимое	1101	13
	×	×
Множитель	101	5
1-е частичное произведение	1101	65_{10}
2-е частичное произведение	0000	
3-е частичное произведение	1101	
Конечное произведение	1000001_2	

Поскольку бит множителя в разряде веса 1 является 1, множимое копируется и составляет первое частичное про изведение. Вторым битом множителя является 0, тогда второе частичное произведение есть 0000 (заметим, что оно сдвинуто на одну позицию влево). Битом разряда веса 4 множителя является 1, тогда для получения третьего частичного произведения снова следует копирование множимого (заметим, что копирование завершается новым сдвигом на одну позицию влево). После этого выполняем сложение трех частичных произведений, что дает результат 1000001_2. Полученный результат $1101_2 X 101_2 = 1000001_2$ соответствует произведению десятичных чисел $13_{10}X5_{10}=65_{10}$.

Дополнительный код

Сама ЭВМ обрабатывает информацию обычно в двоичном коде. Однако если нужно использовать числа со знаком, используется специальный *дополнительный код,* что упрощает аппаратные средства ЭВМ.

На рис. 7.3, *а* приведено обычное изображение регистра МП или ячейки памяти вне МП. Такой регистр представляют пространством из 8 бит данных. Позиции бит пронумерованы от 7 до 0, а веса двоичных позиций указаны в основании регистра, бит 7 имеет вес 128, бит 6 — 64 и т. д.

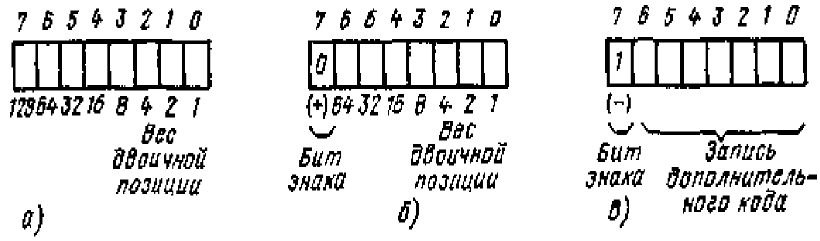

Рис. 7.3. Изображение регистра МП или ячейки памяти:
а — расположение двоичных позиций; б — идентификация положительных чисел нулем в знаковом бите; в — идентификация отрицательных чисел единицей в знаковом бите

На рис.7.3, б и *в* показаны типовые структуры 8-разрядных регистров для размещения *чисел со знаком.* В обоих случаях бит 7 является знаковым. Он указывает, является ли число положительным (+) или отрицательным (-). При 0 в знаковом бите число положительно, при 1 — отрицательно.

Если, как показано на рис. 7.3, б, число положительно, оставшиеся ячейки памяти (6—0) содержат двоичное 7- разрядное число. Например, если регистр на рис. 7.3, б содержит 0100 0001, это соответствует числу $+65_{10}$ (64+1, знаковый бит положителен). Если в него записано 0111 1111, содержимым будет $+127_{10}$ (знаковый бит положителен: +64+32+16+8+4+2+1), что является наибольшим положительным числом, которое может содержать 7-разрядный регистр.

Если, как это показано на рис. 7.3, а, регистр содержит то же число со знаком, но отрицательное, он будет содержать дополнительный код этого числа. В табл. 7.1 приведена запись в дополнительном коде положительных и отрицательных чисел. Заметим, что все положительные числа имеют 0 в старшем бите, остальные биты составляют двоичное число. Все отрицательные числа имеют 1 в старшем разряде. Рассмотрим строку +0 в табл. 7.1: запись в дополнительном коде +0 будет 0000 0000. В ближайшей нижней строке видим, что запись в дополнительном коде — 1 следующая: 1111 1111. Рассмотрим пошаговое перемещение в обратном направлении от 0000 0000 до 1111 1111.

Таблица 7.1. Десятичные числа со знаком и их представление в дополнительном коде

Десятичные	Представление чисел со знаком	Примечания
+127	0111 1111	
.	.	
.	.	
.	.	
+8	0000 1111	Положительные числа представлены в той же форме, что и прямые двоичные числа
+7	0000 0111	
+6	0000 0110	
+5	0000 0101	
+6	0000 0100	
+3	0000 0011	
+2	0000 0010	
+1	0000 0001	
+0	0000 0000	
—1	1111 1111	Отрицательные числа представлены в форме дополнительного кода
—2	1111 1110	
—3	1111 1101	
—4	1111 1100	
—5	1111 1011	
—6	1111 1010	
—7	1111 1001	
—8	1111 1000	
.	.	
.	.	
—128	1000 0000	

Какой будет запись в дополнительном коде числа -9? Рассмотрим этапы преобразования. Они следующие:

Десятичное число	9	Этап 1.	Запись десятичного числа без знака (9)
Двоичное число	0000 1001	Этап 2.	Преобразование десятичного числа в двоичный код (0000 1001)
Дополнение до 1 (обратный или инверсный код)	1111 0110	Этап 3.	Получить обратный код двоичного числа заменой нулей единицами, а единиц — нулями (1111 0110)
Дополнение до 2 (дополнительный код)	+1 1111 0111	Этап 4.	Прибавить единицу к обратному коду. Здесь прибавить 1 к 1111 0110, что дает 1111 0111

Полученный результат является дополнительным кодом положительного десятичного числа. В приведенном приме ре дополнительным кодом числа 9 является 1111 0111. За метим, что знаковый бит — 1, это означает, что рассматриваемое число (1111 0111) отрицательно.

Каким будет десятичный эквивалент числа 1111 0000, записанного в форме дополнительного кода? Процедура преобразований в этом случае следующая:

Дополнительный код	1111 0000	Этап 1.	Запись дополнительного кода (1111 0000).
Дополнение до 1	0000 1111	Этап 2.	Получается обратный код дополнительного кода заменой нулей единицами, а единиц — нулями (0000 1111)
Двоичное число	+1 0001 0000=16	Этап 3.	Добавить 1

Таким образом, формирование обратного кода и добавление 1 являются теми же процедурами, которые мы про водили при преобразовании двоичного числа в дополнительный код. Однако следует отметить, что, хотя мы получили двоичное число 0001 0000 = 16_{10}, исходная запись дополни тельного кода 1111 0000 = -16, т. е. имеем отрицательное число, поскольку старший бит в дополнительном коде является 1.

Арифметика в дополнительном коде

Микропроцессор может использовать числа в форме дополнительного кода, потому что он в состоянии выполнять операции *дополнения (инверсии), инкрементирования* (добавления 1 к числу) и *сложения* двоичных чисел. Микро процессор не приспособлен для прямого вычитания. Он использует сумматоры и для выполнения вычитания оперирует над дополнительным кодом.

Сложим десятичные числа +5 и +3. Рассмотрим процедуру действии в случае одновременного сложения чисел в десятичном и в дополнительном кодах: Согласно табл. 2.10 +5 = 0000 0101 в дополнительном коде аналогично +3 = 0000 0011. Тогда числа в дополни тельном коде 0000 0101 и 0000 0011 складываются, как обычные двоичные числа, давая сумму 0000 1000 в дополнительном коде, т. е. 0000 1000 = $+8_{10}$.

1-е число	(+5)	0000 0101
	+	+
2-е число	(+3)	0000 0011
	(+8)	0000 1000

Пусть надо сложить десятичные числа +7 и -3. Согласно табл. 7.1 +7 = 0000 0111 и -3 = 1111 1101 соответственно в дополнительном коде. Они затем складываются, как обычные двоичные числа, и результат 1 0000 0100 получается в дополнительном коде:

1-е число	(+7)	0000 0111
	+	+
2-е число	(—3)	1111 1101
	(+4)	1 0000 0100

Пренебречь переполнением.

Старший бит является переполнением 8-разрядного регистра, и им можно пренебречь. Получаем сумму 0000 0100 или 4-410.

Сложим десятичные числа +3 и -8. Согласно все той же табл. 7.1 +3 = 0000 0011 и -8 = 1111 1000. Их дополнительные коды 0000 0011 и 1111 1000 складываются, как обычные двоичные числа, что дает 1111 1011 = $—5_{10}$:

1-е число	(+3)	0000 0011
	+	+
2-е число	(—8)	1111 1000
	(—5)	1111 1011

Сложим десятичные числа -2 и -5. В дополнительном коде согласно табл. 2.10 -2 = 1111 1110 и -5 = 1111 1011. Два числа 1111 1110 и 1111 1011 складываются, как обычные десятичные числа, что дает 1 1111 1001:

1-е число	(—2)	1111 1110
	+	+
2-е число	(—5)	1111 1011
	(—7)	1 1111 1001

Пренебречь переполнением.

Старший бит результата является переполнением 8-раз рядного регистра, и им пренебрегаем. Таким образом, суммой двух чисел 1111 1110 и 1111 1011 в дополни тельном коде будет 1111 1001. Согласно табл. 2.10 сумма 1111 1001= - $7_{|0}$.

Вычтем теперь десятичное число +5 из десятичного числа +8. Первое число +8 = 0000 1000, второе +5 = 0000 0101. В дополнительный код (инвертировать и добавить 1) должно быть преобразовано число 0000 0101, что дает 1111 1011. Затем первое число 0000 1000 складывается с дополнительным кодом второго 1111 1011, как с обычным двоичным числом, что дает 1 0000 0011:

1-е число	(+8)		0000 1000
	−	Дополнительный код	
2-е число	(+5) = 0000 0101	——————→	1111 1011
	(+3)		1 0000 0011
			Пренебречь переполнением.

Старший бит является переполнением регистра, им пренебрегаем, что дает результат 0000 0011 = $+3_{10}$- Заметим, что второе число было представлено в дополнительном ко де, затем сложено с первым. *Используя дополнительный код и сумматор, микропроцессор выполняет вычитание.*

Вычтем теперь большее десятичное число +6 из десятичного числа +2:

1-е число	(+2)		0000 0010
	−	Дополнительный код	+
2-е число	(+6) = 0000 0110	——————→	1111 1010
	(−4)		1111 1100

Дополнительный код первого числа +2 = 0000 0010, второе число +6 = 0000 0110, его дополнительный код (ин версия и добавление 1) — 1111 1010. Оба эти кода сложены затем, как обычные двоичные числа, что дает 1111 1100, а согласно табл. 2.10 1111 1100 = -4_{10}.