А. А. Вылиток

Операции над битовыми наборами. Установка флагов и проверка условий

1. Определение операций сложения и вычитания битовых наборов

Битовые наборы используются для представления данных в ЭВМ, в частности для представления чисел. Сами наборы можно считать словами в алфавите $\{0,\ 1\}$. Сначала мы формально определим операции сложения и вычитания на множестве битовых наборов длины k. Затем исследуем связь этих операций с операциями сложения и вычитания чисел.

Пусть x и y — битовые наборы. Перенумеруем биты наборов справа налево от 0 до $k-1$. Через z_i будем обозначать i-й бит набора z, $z_i \in \{0,\ 1\}$. Для определения результатов (суммы и разности) воспользуемся некоторыми логическими операциями (дизъюнкция, конъюнкция, отрицание, сложение по модулю два), интерпретируя 0 как ложь, 1 — как истину.

Суммой (соответственно, *разностью*) *битовых наборов* x и y назовем набор z, разряды которого определяются ниже. Попутно определим набор величин c_i, $i = 0, \ldots, k$, $c_i \in \{0,\ 1\}$, где c_i означает наличие (при $c_i = 1$) или отсутствие (при $c_i = 0$) переноса единицы в i-й разряд, а в случае вычитания — заёма из i-го разряда. Операцию вычисления суммы (соответственно, разности) битовых наборов назовем *сложением* (соответственно, *вычитанием*):

сложение $z = x + y$

$$\begin{cases} c_0 = 0, \\ \text{для } i = 0, \ldots, k-1 \\ z_i = x_i \oplus y_i \oplus c_i\ , \\ c_{i+1} = x_i y_i \vee x_i c_i \vee y_i c_i \end{cases}$$

вычитание $z = x - y$

$$\begin{cases} c_0 = 0, \\ \text{для } i = 0, \ldots, k-1 \\ z_i = x_i \oplus y_i \oplus c_i\ , \\ c_{i+1} = y_i c_i \vee z_i c_i \vee y_i z_i \end{cases}$$

Здесь $\oplus$ означает «сложение по модулю два», $\vee$ — дизъюнкцию, знак конъюнкции (логического умножения) опущен.

Если рассматривать битовые наборы как представление беззнаковых целых чисел или знаковых в дополнительном коде, то определенные выше операции над битовыми наборами реализуют сложение и вычитание целых чисел по модулю 2^k. Во многих ЭВМ схемы сложения и вычитания чисел устроены именно так. Поскольку при подобных операциях результат может быть искажен (из-за переполнения разрядной сетки или мантиссы), для отслеживания этой ситуации одновременно с результатом устанавливаются специальные флаги.

2. Установка флагов при сложении и вычитании

Флаг — это бит, принимающий значение 1 («флаг установлен»), если выполнено некоторое условие, и значение 0 («флаг сброшен») в противном случае. Наиболее употребительны флаги:

- *CF* — *carry flag* (флаг переноса) фиксирует искажение результата операции для чисел без знака (он устанавливается, например, когда полученная сумма k-разрядных чисел «не умещается» в k-разрядную ячейку);

- *OF* — *overflow flag* (флаг переполнения мантиссы) фиксирует искажение результата для знаковых чисел;
- *ZF* — *zero flag* (флаг нуля) фиксирует нулевой результат;
- *SF* — *sign flag* (флаг знака) устанавливается в 1, если результат операции над знаковыми числами отрицательный.

Заметим, что сложение и вычитание битовых наборов приводит к установке значений всех флагов, независимо от того, трактуем ли мы эти наборы как числа со знаком или как беззнаковые числа. Для анализа корректности результата в знаковом случае бесполезен *CF*, для случая беззнаковых чисел бесполезны *OF* и *SF*.

2.1. Формальное определение флагов

Приведем формулы для определения значений флагов в терминах битовых наборов, используя введенные выше обозначения. После сложения или вычитания наборов x и y (результат в z) флаги *CF*, *SF*, *ZF*, *OF* получают следующие значения [1]:

$$CF = c_k \quad SF = z_{k-1} \quad ZF = \bar{z}_{k-1}\bar{z}_{k-2}\ldots\bar{z}_0\,, \quad OF = \begin{cases} \bar{x}_{k-1}\bar{y}_{k-1}z_{k-1} \vee x_{k-1}y_{k-1}\bar{z}_{k-1} & \textit{при сложении,} \\ \bar{x}_{k-1}y_{k-1}z_{k-1} \vee x_{k-1}\bar{y}_{k-1}\bar{z}_{k-1} & \textit{при вычитании} \end{cases}$$

2.2. Смысловое определение флагов

Правила установки значений флагов можно описать и в терминах беззнаковых или знаковых (в дополнительном коде) чисел. В следующей таблице наборы x, y и z интерпретируются как числа (со знаком или без знака), набор z — сумма или разность (по модулю 2^k) наборов x и y, знаки +, − означают обычные (не по модулю 2^k) операции сложения и вычитания над целыми числами.[2]

Флаг	Условия установки значения флага в 0 или в 1 для			
	беззнаковых чисел		знаковых чисел	
	0	1	0	1
ZF	$z \neq 0$	$z = 0$	$z \neq 0$	$z = 0$
SF	$z \in [0, 2^{k-1} - 1]$	$z \in [2^{k-1}, 2^k - 1]$	$z \geq 0$	$z < 0$
CF	*при сложении*: $x + y < 2^k$ *при вычитании*: $x \geq y$	*при сложении*: $x + y \geq 2^k$ *при вычитании*: $x < y$	*при сложении*: $x \geq 0, y \geq 0$ *или* $xy \leq 0, x + y < 0$ *при вычитании*: $x < 0, y \geq 0$ *или*	*при сложении*: $x < 0, y < 0$ *или* $xy < 0, x + y \geq 0$ *при вычитании*: $x \geq 0, y < 0$ *или*

[1] Горизонтальная черта над переменными (надчеркивание) означает логическое отрицание.

[2] Доказательство эквивалентности правил, приведенных в таблице, и приведенных выше формальных правил для битовых наборов, = предлагается в качестве упражнения.

<table>
<tr><th rowspan="3">Флаг</th><th colspan="4">Условия установки значения флага в 0 или в 1 для</th></tr>
<tr><th colspan="2">беззнаковых чисел</th><th colspan="2">знаковых чисел</th></tr>
<tr><th>0</th><th>1</th><th>0</th><th>1</th></tr>
<tr><td></td><td></td><td></td><td>$x \ge y \ge 0$ или $0 > x \ge y$</td><td>$x< y < 0$ или $0 \le x < y$</td></tr>
<tr><td>OF</td><td>при сложении:
$x, y, z< (\ge)2^{k-1}$ или
$x < (\ge)2^{k-1} \le (>)\ y$
при вычитании:
$x, y < (\ge)2^{k-1}$ или
$x, z < (\ge)2^{k-1}$,
$y \ge (<)2^{k-1}$</td><td>при сложении:
$x, y < (\ge)2^{k-1}$,
$z \ge (<)2^{k-1}$
при вычитании:
$y, z < (\ge)2^{k-1}$,
$x \ge (<)2^{k-1}$</td><td>при сложении и вычитании:
$-2^{k-1} \le x \pm y < 2^{k-1}$</td><td>при сложении и вычитании:
$x \pm y < -2^{k-1}$ или
$x \pm y \ge 2^{k-1}$</td></tr>
</table>

2.3. Примеры

Приведем для каждого допустимого набора значений флагов *CF*, *SF*, *ZF*, *OF* пример операций сложения и вычитания, после выполнения которых устанавливается данный набор значений. Формат чисел во всех примерах — 1 байт. Операнды будем записывать в десятичной системе счисления, результат указывать и как число без знака, и как число со знаком.

Составим таблицу: в правой колонке перечислим все возможные комбинации значений флагов, слева для каждой комбинации будем выписывать по одному примеру на сложение и на вычитание, если такие примеры существуют.

<table>
<tr><th rowspan="2">Пример операции (сложение, вычитание)</th><th colspan="2">Результат как число</th><th colspan="4">Значение флага</th></tr>
<tr><th>без знака</th><th>со знаком</th><th>CF</th><th>SF</th><th>ZF</th><th>OF</th></tr>
<tr><td>1 + 2</td><td>3</td><td>+3</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td></tr>
<tr><td>3 – 1</td><td>2</td><td>+2</td></tr>
<tr><td colspan="3">не существует</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td></tr>
<tr><td>−80 − 50</td><td>126</td><td>+126</td></tr>
<tr><td>0 + 0</td><td>0</td><td>0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td></tr>
<tr><td>1 – 1</td><td>0</td><td>0</td></tr>
<tr><td colspan="3">не существует</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">1</td></tr>
<tr><td colspan="3">не существует</td></tr>
<tr><td>−40 + 20</td><td>236</td><td>−20</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0</td></tr>
<tr><td>−20 − 40</td><td>196</td><td>−60</td></tr>
<tr><td>120 + 120</td><td>240</td><td>−16</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">1</td></tr>
<tr><td colspan="3">не существует</td></tr>
<tr><td colspan="3">не существует</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">0</td></tr>
<tr><td colspan="3">не существует</td></tr>
</table>

не существует			0	1	1	1
не существует						
−2 + 3	1	+1	1	0	0	0
40 − (−20)	60	+60				
140 + 140	24	+24	1	0	0	1
не существует						
−1 + 1	0	0	1	0	1	0
не существует						
−128 + (−128)	0	0	1	0	1	1
не существует						
−20 + (− 40)	196	−60	1	1	0	0
20 − 40	236	−20				
не существует			1	1	0	1
127 − 128	255	−1				
не существует			1	1	1	0
не существует						
не существует			1	1	1	1
не существует						

3. Проверка условий с помощью вычитания и флагов

Практически во всех типах ЭВМ существуют команды перехода по условию. Многие такие команды укладываются в схему *«Если* $op_1 \diamond op_2$, *то переход по адресу A»*, где op_1, op_2 — операнды (битовые наборы длины k), $\diamond$ — одна из операций отношения: $-_б$, $\neq_б$, $<_б$, $>_б$, $\leq_б$, $\geq_б$, $=_{зн}$, $\neq_{зн}$, $<_{зн}$, $>_{зн}$, $\leq_{зн}$, $_{зн}$ (здесь индекс *«б»* означает, что операнды трактуются как числа без знака, *«зн»* — со знаком).

Пусть для представления знаковых чисел используется дополнительный код. Приведем метод, позволяющий проверить истинность условия $op_1 \diamond op_2$ по значениям флагов (OF, CF, SF, ZF), получаемым при вычитании: $op_1 - op_2$ по модулю 2^k.

Поскольку разность (по модулю 2^k) двух одинаковых чисел (знаковых или беззнаковых) равна нулю, разность неравных чисел отлична от нуля и флаг ZF равен единице, если и только если результат операции — ноль, справедливы утверждения:

$$op_1 = op_2 \Leftrightarrow ZF = 1, \qquad\qquad op_1 \neq op_2 \Leftrightarrow ZF = 0.$$

Итак, для того чтобы установить, равны или нет два числа, достаточно знать, чему равен флаг ZF.

Значения флагов при других соотношениях между операндами (кроме ≠ и =) различны в знаковом и беззнаковом случаях. Поэтому рассмотрим эти случаи отдельно.

Вспомним, что в терминах беззнаковых чисел флаг CF равен единице, если уменьшаемое меньше вычитаемого, и равен нулю в противном случае. Тогда можно утверждать, что:

$$op_1 <_б op_2 \Leftrightarrow CF = 1, \qquad op_1 \geq_б op_2 \Leftrightarrow CF = 0.$$

Условия для остальных отношений можно логически выразить через уже установленные. Например, $op_1 >_б op_2 \Leftrightarrow (op_1 \geq_б op_2) \wedge (op_1 \neq op_2)$, где $\wedge$ означает конъюнкцию. Следовательно, $op_1 >_б op_2 \Leftrightarrow (CF = 0) \wedge (ZF = 0)$.

Для случая знаковых чисел нам важны флаги OF и SF [3]. Посмотрим, какие значения принимают эти флаги в зависимости от знаков операндов и от соотношений [4] между операндами:

Знак op_1	**Знак** op_2	**Возможное при данных знаках соотношение**	**Возможные значения флагов при данном соотношении** [5]	
			OF	SF
+	+	<	0	1
		≥	0	0
+	−	≥	0	0
			1	1
−	+	<	0	1
			1	0
−	−	<	0	1
		≥	0	0

Проанализируем эту таблицу. Заметим, что множества пар значений флагов (OF, SF) для отношений < и ≥ не пересекаются, а их объединение дает все возможные пары из нулей и единиц. Следовательно, по паре значений (OF, SF) можно определить, какое из двух взаимоисключающих соотношений (< или ≥) выполняется. В случае $op_1 < op_2$ имеем: либо $OF = 0$ и $SF = 1$, либо $OF = 1$ и $SF = 0$, или, короче, $OF \neq SF$. В случае $op_1 \geq op_2$ справедливо равенство $OF = SF$. Таким образом, можно утверждать:

$$op_1 <_{зн} op_2 \Leftrightarrow OF \neq SF, \qquad op_1 \geq_{зн} op_2 \Leftrightarrow OF = SF$$

Подытожим полученные результаты в виде таблицы.

Соотношение $op_1 \diamond op_2$	**Критерий истинности данного соотношения в терминах значений флагов после операции вычитания** $op_1 - op_2$ **по модулю** 2^k	
	для чисел со знаком	**для чисел без знака**
=	$ZF = 1$	$ZF = 1$

---

3) В «бесполезности» остальных флагов можно было бы убедиться, включив и их в рассмотрение, но это заняло бы больше места.

4) Рассмотрим, как и в беззнаковом случае, только отношения < и ≥ — остальные можно выразить при помощи логических связок.

5) Читателю предлагается самостоятельно доказать, что именно такие значения флагов возможны в каждом из рассматриваемых случаев.

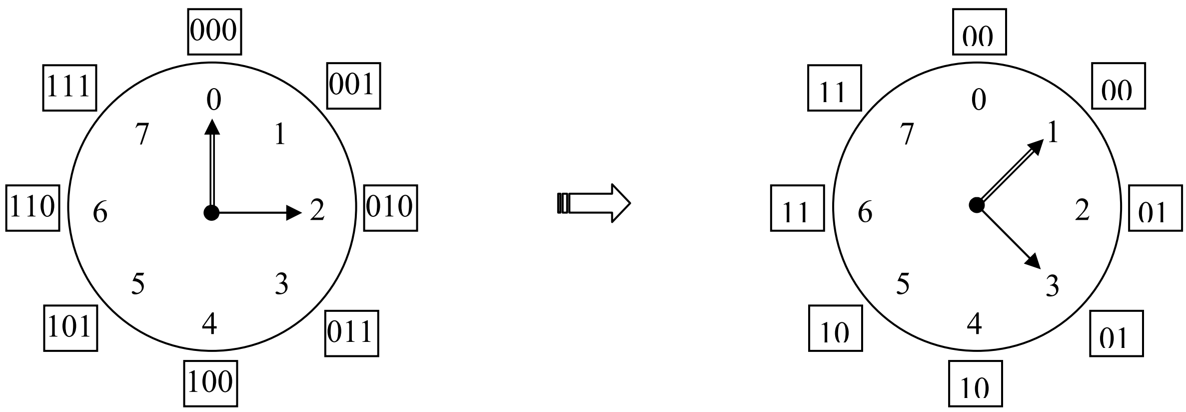
$\neq$	$ZF = 0$	$ZF = 0$
$<$	$OF \neq SF$	$CF = 1$
$\geq$	$OF = SF$	$CF = 0$
$>$	$(OF = SF) \wedge (ZF = 0)$	$(CF = 0) \wedge (ZF = 0)$
$\leq$	$(OF \neq SF) \vee (ZF = 1)$	$(CF = 1) \vee (ZF = 1)$

4. Реализация сложения и вычитания по модулю *N* с помощью «механики часов» (на примере *N* = 3)

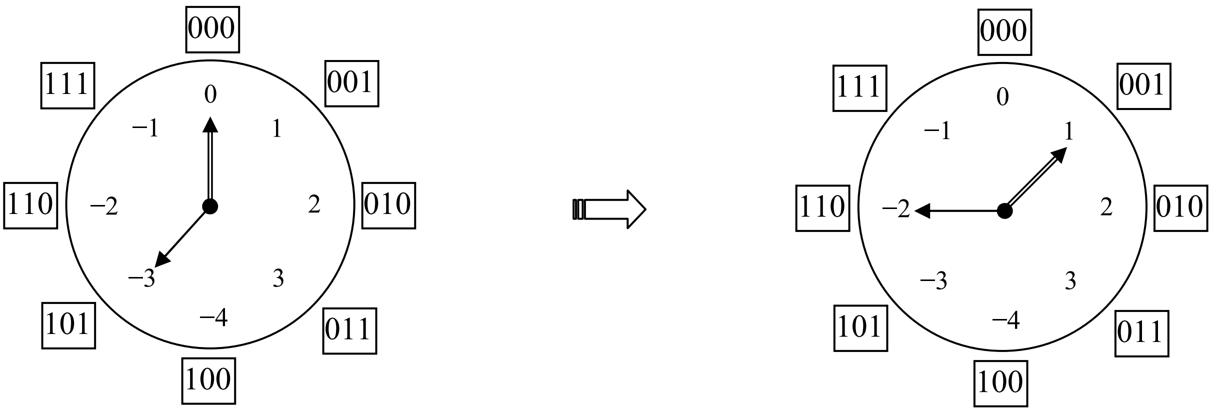
Покажем, как можно механически реализовать сложение и вычитание с помощью «циферблата» и двух часовых стрелок. По окружности циферблата равномерно расположим 2^N чисел. Первую (часовую) стрелку будем изображать более широкой, чем вторую (минутную). Заметим, что от пользователя данного механизма не требуется умения складывать или вычитать числа, прибавлять единицу и т.п.

Для сложения: первую стрелку установить на ноль, вторую — на второе слагаемое. Зафиксировать угол между стрелками и совместить первую стрелку с первым слагаемым, тогда вторая укажет на результат сложения. Этот прием годится и для чисел со знаком (в дополнительном коде), и для чисел без знака.

Пример для чисел без знака: 1 + 2 = 3



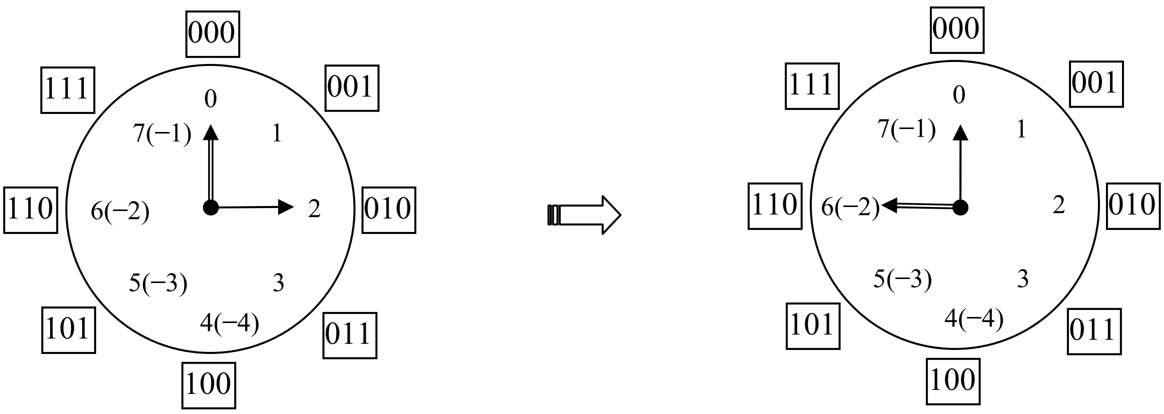
Пример для чисел со знаком: −3 + 1 = −2



Для вычитания: первую стрелку установить на ноль, вторую — на вычитаемое. Зафиксировать угол и совместить вторую стрелку с уменьшаемым. Тогда первая стрелка укажет результат вычитания.

Пример: $0 - 2 = -2$ (для знаковых чисел)

или $0 - 2 = 6$ (беззнаковые вычитаем по модулю 8)



Задача. Сформулировать правила установки флагов с помощью механики часов.