

Представление целых чисел в памяти ЭВМ. Индикаторы переноса и переполнения

Для упрощения будем разбирать четырехбитовые машинные слова. Такой размер слова обеспечивает хранение десятичных чисел всего лишь от 0 до 15 (если брать только положительные целые числа). Однако закономерности, обнаруженные на примере четырехбитовых слов, сохраняют силу для машинного слова любого размера.

Предположим, что процессор ЭВМ способен увеличивать (прибавлять 1) и дополнять (инвертировать) четырехбитовые слова. Например, результатом увеличения слова 1100 является 1101, а результатом дополнения этого слова является 0011. Рассмотрим слово 0000, представляющее десятичное число 0. В результате увеличения содержимое этого слова станет равным 0001, что соответствует десятичному числу 1. Продолжая последовательно увеличивать четырехбитовые слова, придем к ситуации, когда, увеличивая слово 1111 (которое представляет десятичное число 15), получим в результате слово 0000, т.е. $1111 + 1 = (1)0000$. Т.е. Получили неверную арифметическую операцию и вернулись в исходное состояние. Это произошло из-за того, что слово памяти может состоять из конечного числа бит. Таким образом, числовая система ЭВМ является конечной и цикличной.

Можно битовую конфигурацию 1111 принять за код для -1. Тогда 1110 интерпретируется как -2, 1101 как -3, 1000 как -8. Тем самым получили другую числовую систему — со знаком, содержащую как положительные, так и отрицательные числа. В этой системе половина четырехбитовых конфигураций, начинающаяся с единицы, интерпретируется как отрицательные числа, а другая половина, начинающаяся с нуля, — как положительные числа или нуль. Поэтому старший бит числа (третий по счету, если нумерацию бит начинать с нуля справа налево) называется знаковым битом. Числовая система со знаком также конечна и циклична, арифметически неверный результат даст попытка увеличить число 7 на единицу ($0111 + 1 = 1000 = -8$).

Если знаковый бит равен нулю, то значение числа легко вычисляется — игнорируется знаковый бит, а оставшиеся три бита интерпретируются как двоичный код десятичного числа. Например, слово 0110 представляет двоичное число 110, которое равно

десятичному числу 6.

Для оценки отрицательного числа нужно изменить его знак. Рассмотрим четырехбитовое число k в системе со знаком. Тогда $-k = (-1 - k) + 1$, следовательно, для вычисления значения $-k$

необходимо вычесть

k

из -1 (т.е. из 1111) и затем прибавить единицу. Заметим, что операция вычитания всегда возможна, никогда не требует заема и равнозначна операции инвертирования битов вычитаемого. Например, $1111 - 1011 = 0100$; здесь в вычитаемом, равном 1011 , единицы перешли в нули, а нуль — в единицу. Инвертирование бит в слове называется дополнением до единицы. Для определения отрицательного значения числа

k

надо к его дополнению до единицы прибавить единицу (согласно вышеприведенному равенству). Инвертирование бит в слове с добавлением единицы к младшему биту называется дополнением до двух. Например, требуется найти, какое число закодировано в слове 1001 . Для этого сначала выполняем операцию инвертирования: $1001 \rightarrow 0110$, а затем к полученному результату прибавляем единицу $0110 + 1 = 0111$, что является двоичным кодом числа 7. Таким образом, значением 1001 является число -7 .

Индикаторы переноса и переполнения

Рассмотрим более подробно ситуацию, приводящую при увеличении четырехбитового числа (т.е. при прибавлении к нему единицы) к неверному арифметическому результату, возникающему из-за конечности числовой системы ЭВМ. В числовой системе без знака такая проблема встает при увеличении слова 1111 , при этом имеет место перенос единицы из знакового бита в никуда. В случае системы чисел со знаком перенос из старшего бита дает верный результат: $1111 + 0001 = (1)0000$ (что правильно: $-1 + 1 = 0$), но увеличение слова 0111 приводит к ошибочной ситуации: $0111 + 1 = 1000$ ($7 + 1 = -8$), при этом имеет место перенос в знаковый бит.

В процессоре ЭВМ (той его части, которая ответственна за выполнение арифметических операций) имеется два индикатора — индикатор переноса и индикатор переполнения. Каждый индикатор содержит один бит информации и может быть установлен процессором (в этом случае ему придается значение, равное единице) или сброшен (равен нулю). Индикатор переноса указывает на перенос из знакового бита за пределы слова, а индикатор переполнения на перенос в знаковый бит. Таким

образом, после завершения операции, в которой происходит перенос в самый старший бит, процессор устанавливает индикатор переполнения; если такого переноса нет, то индикатор переполнения сбрасывается. Индикатор переноса обрабатывается аналогичным образом.

Важно то, что после выполнения операции процессором ЭВМ сигнализирует о состоянии индикаторов, и их можно проверить. Если состояние индикаторов указывает на неправильный арифметический результат, то необходимо исправить эту ситуацию, т.е. пользователю ЭВМ предоставляется возможность контролировать правильность выполнения арифметических операций.

Пример 1

Рассмотрим сложение двух чисел: $0101 + 0011 = 1000$. В результате выполнения операции сложения произошел перенос в знаковый бит, а переноса из знакового бита не было. Таким образом, после завершения этой операции индикатор переноса будет сброшен, а индикатор переполнения установлен. Поэтому если рассматривать эти два числа как целые без знака, то результат является арифметически правильным, т.к. индикатор переноса сброшен. Если же рассматривать числа в системе со знаком, то бит переполнения показывает, что произошло изменение знака (перенос в знаковый бит есть, а перенос из него — нет), поэтому арифметически результат неправильный.

Пример 2

В результате сложения чисел $1101 + 0101 = 0010$ происходит перенос в знаковый бит и из знакового бита. Поэтому будет установлен индикатор переноса, а индикатор переполнения сброшен. Следовательно, в системе чисел без знака результат является арифметически неправильным, а в системе чисел со знаком — правильным.

Пример 3

В результате выполнения операции вычитания: $1001 - 0011 = 1001 + (-0011) = 1001 +$

$(1100 + 1) = 1001 + 1101 = 0110$ происходит перенос из знакового бита, а перенос в знаковый бит нет. Следовательно, индикатор переноса будет сброшен, а индикатор переполнения установлен. Это указывает на то, что в данном примере в системе без знака результат арифметически правильный, а в системе со знаком — неправильный.