

## 6. Логические команды

Особенность логических команд – обработка отдельных битов в байте.

Все логические команды можно разбить на две группы:

- выполнение основных логических операций
- команды, выполняющие сдвиги битов в байте

В первую группу входят следующие команды:

- логическое отрицание: NOT операнд ;инверсия битов
- логическое сложение: OR op1, op2 ;  $op1 = op1 \text{ OR } op2$
- логическое умножение: AND op1, op2 ;  $op1 = op1 \text{ AND } op2$
- исключающее ИЛИ: XOR op1, op2 ;  $op1 = op1 \text{ XOR } op2$

Особенности выполнения логических операций:

- операции – только над одноименными битами, причем – параллельно сразу со всеми
- длины операндов – согласованные (либо байты, либо слова)
- операнд1 – либо регистр, либо память, а операнд 2 – либо регистр, либо память, либо непосредственный операнд.

Примеры:

MOV AL, 1100b ; (AL) = 0000 1100

NOT AL ; (AL) = 1111 0011

AND AL, 1010b ; (AL) = 0000 0010

OR AL, 0011b ; (AL) = 0000 0011

XOR AL, 1111b ; (AL) = 0000 1100

Команда XOR имеет одно интересное применение: если операнды команды совпадают, то результат будет нулевым. Это можно использовать для очистки регистров, что быстрее записи туда нуля или вычитания:

XOR AX, AX ; (AX) = (AX) XOR (AX) = 0, что бы ни было в AX

Команды второй группы позволяют сдвигать набор битов в байте или слове влево или вправо на заданное количество позиций разными способами.

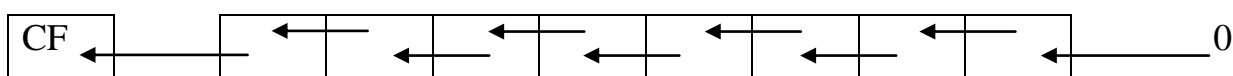
Различают команды логического сдвига, арифметического сдвига и циклического сдвига. Общий формат команд сдвига:

МКОП операнд, величина

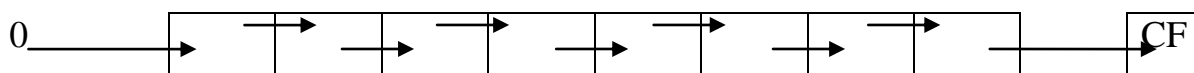
Здесь “операнд” - это байт или слово в регистре или области памяти, а “величина” - целое число позиций сдвига.

Логический сдвиг реализуется следующим образом: крайний выталкиваемый за разрядную сетку бит сохраняется в битовом флаге переноса CF, а на освободившееся место с другого конца заносится ноль:

сдвиг влево на один разряд: SHL op, 1 ; SHL - это SHift Left



сдвиг вправо на один разряд: SHR op, 1 ; SHR – это SHift Right



Пример.

MOV AL, 01010101b ; (AL) = 01010101

SHL AL, 1 ; влево на 1: (CF) = 0, (AL) = 10101010

SHR AL, 1 ; вправо на 1: (CF) = 0, (AL) = 01010101

SHL AL, 3 ; влево на 3: 10101(000)

SHR AL, 2 ; вправо на 2: (00)101010

Одно из применений логических сдвигов – умножение и деление целых чисел на степени 2. Например, вместо умножения на 8 можно выполнить сдвиг влево на 3 позиции. Например:  $5 = 101_2$  после сдвига влево на 3 будет  $101000_2 = 40_{10}$ .

Арифметические сдвиги:

- влево – команда SAL (т.е. Shift Arithmetic Left), которая работает точно так же, как и SHL
- вправо – команда SAR, которая в старший бит заносит не 0, а 0 или 1 в зависимости от знака исходного числа, т.е. число сохраняет свой знак

MOV BH, 10001110b ; (BH) = 10001110

```
SAR BH, 1 ; (BH) = 11000111
MOV BH, 00001111b ; (BH) = 00001111
SAR BH, 1 ; (BH) = 00000111
```

Циклический сдвиг имеет следующую особенность: “выталкиваемый” бит восстанавливается с другого конца с одновременным занесением в CF. Циклический сдвиг влево – команда ROL (т.е. Rotate Left), соответственно вправо – ROR.

Пример:

```
MOV CL, 11000011b ; (CL) = 11000011
ROL CL, 1 ; (CL) = 10000111
```

Основное использование логических команд – выделение в ячейке заданных битов с обнулением остальных. Для этого используется команда AND, в которой второй операнд имеет 1 в тех битах, которые надо оставить, и 0 – там, где надо обнулить. Такие операнды называются масками.