

Макросы: [macro.s]

%macro NAME 1

...

%endmacro

- NAME - название

- 1 - количество аргументов (может быть 0)

- При использовании в коде происходит подстановка кода макроса вместо имени

Битовые операции:

• AND, OR, XOR, NOT -

- в целом из названия уже видно, что они делают

• Выполняются побитово ↗

[Пример: bin-op.s]

[Пример shift.s] ↓

• Битовые сдвиги: арифметический  
логический

• SHL - сдвиг влево (sal - то же самое)

$\text{shl dest, count} ; \text{dest} = \text{dest} \ll \text{count}$

- Младшие биты заполняются нулями
- Эквивалентно умножению на  $2^n$  (для беззнаковых чисел)
- $\text{dest}$ : регистр или память,  $\text{count}$ :立即数 или  $\text{CF}$
- Последний выполненный бит выставляется в  $\text{CF}$
- $\text{PF}$   $\text{DF}$  выставляются, если при сдвиге на 1 изменился знак бит
- $\text{ZF}, \text{SF}$  - как обычно

•  $\text{SHR}$  - сдвиг вправо (логический)

$\text{shr dest, count} ; \text{dest} = \text{dest} \gg \text{count}$

- Старшие биты заполняются нулями
- Эквивалентно делению на  $2^n$  (для беззнаковых чисел)
- $\text{CF}, \text{ZF}$  - аналогично сдвигу влево
- $\text{OF}$  - равен старшему биту до сдвига (почему?)

SFD - всегда 0 (ноль?)

• Сдвиг вправо: (арифметический):

`SAR dest, count`

- Сдвиг вправо с сохранением знака
- Старшие биты заполняются копией знаменного бита
- CF, SF, ZF - аналогично
- OF - всегда 0

• SHLD - сдвиг влево глоб и лока

`SHLD dest, src, count`

- Младшие биты заполняются старшими битами `src`
- OF выставляется только при сдвиге на 1 бит, если изменилось значение старшего бита
- CF, ZF, SF - аналогично предыдущим

## • SHRD

shr d dest, src, count

- Старшие бита заполняются младшими битами src
- Остальные аналогично

## Циклические сдвиги: [cyclic-shifts]

- ROR — циклический сдвиг вправо
- ROL — циклический сдвиг влево

ror dest, count

rol dest, count

- Последний сдвинутый бит копируется в CF
- При сдвиге на 1 бит меняется OF; если изменилось значение старшего бита
- ZF, SF — не обновляются этими инструкциями

\* Циклические сдвиги с флагом переноса:

RCL — влево RCR — вправо

- Можно думать о них, как об обычных арифметических выражениях, но в области:



- OF устанавливается только при сдвиге на 1 бит, если изменилось значение старшего бита
- Остальные флаги не изменяются

Сравнение:

- $\text{test } op1, op2 \Rightarrow op1 \& op2$  (по битам)
- $\text{cmp } op1, op2 \Rightarrow op1 - op2$
- Без сохранения результата, но с выставлением флагов
- $\text{test}$  - изменяет  $ZF, SF$   
сбрасывает  $CF, OF$
- $\text{cmp}$  - изменяет  $ZF, SF, CF, OF$
- После  $\text{cmp } a, b$ :



1.  $a \leq b$   $\text{cmp } a, b$

$$a = b$$

$$ZF = 1$$

$$a \neq b$$

$$ZF = 0$$

$$a < b$$

$$CF = 1$$

$$a > b$$

$$CF = 0$$

[checkup.5

|   | CF | DF | ZF | SF |
|---|----|----|----|----|
| 1 | 0  | 1  | 0  | 1  |
| 2 | 1  | 1  | 0  | 0  |
| 3 | 1  | 0  | 0  | 1  |
| 4 | 1  | 1  | X  | X  |
| 5 | 1  | 0  | X  | X  |
| 6 | 1  | 1  | X  | X  |
| 7 | 1  | 0  | X  | X  |
| 8 | 0  | 0  | 0  | 1  |

α  
↑  
не  
учитывается