**Команди асемблера.**

Є три групи команд.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перша група: ADD, SUB, CMP, AND, TEST, OR, XOR

Ці типи операндів підтримуються:

REG, memory

memory, REG

REG, REG

memory, immediate

REG, immediate

REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.

memory: [BX], [BX + SI + 7], змінна, і т.п ...

immediate: 5, -24, 3Fh, 10001101b, і т.п ...

Після операції між операндами результат завжди записується в перший операнд. Команди CMP і TEST впливають тільки на прапори і не записують результат (ця команда використовується для прийняття рішення під час виконання програми).

Ці команди впливають тільки на прапори:

       CF, ZF, SF, OF, PF, AF.

• ADD - Додати другий операнд до першого.

• SUB - Відняти другий операнд з першого.

• CMP - Відняти другий операнд з першого тільки для прапорів.

• AND - логічне І між усіма бітами двох операндів. При цьому дотримуються правила:

1 AND 1 = 1

1 AND 0 = 0

0 AND 1 = 0

0 AND 0 = 0

Як бачите, ми отримуємо 1 тільки в тому випадку, якщо обидва біти рівні 1.

• TEST - Те ж саме, що AND, але тільки для прапорів.

• OR - логічне АБО між усіма бітами двох операндів. При цьому дотримуються правила:

1 OR 1 = 1

1 OR 0 = 1

0 OR 1 = 1

0 OR 0 = 0

Як бачите, ми отримуємо 1 кожен раз, коли хоча б один біт дорівнює 1.

• XOR - логічне XOR (виключає АБО) між усіма бітами двох операндів. При цьому дотримуються правила:

1 XOR 1 = 0

1 XOR 0 = 1

0 XOR 1 = 1

0 XOR 0 = 0

Як бачите, ми отримуємо 1 кожен раз, коли біти мають різне значення.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Друга група: MUL, IMUL, DIV, IDIV

Ці типи операндів підтримуються:

REG

memory

REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.

memory: [BX], [BX + SI + 7], змінна, і т.п ...

Команди MUL і IMUL впливають тільки на ці прапори:

       CF, OF

Якщо результат перевищує розмір операнда, то ці прапори встановлені в 1, якщо результат уміщається в розмір операнда, то ці прапори встановлені в 0.

Для команд DIV і IDIV прапори не визначені.

• MUL - беззнаковое множення:

якщо операнд - це байт:

AX = AL \* операнд.

якщо операнд - це слово:

(DX AX) = AX \* операнд.

• IMUL - множення зі знаком:

якщо операнд - це байт:

AX = AL \* операнд.

якщо операнд - це слово:

(DX AX) = AX \* операнд.

• DIV - беззнаковое розподіл:

якщо операнд - це байт:

AL = AX / операнд

AH = залишок (модуль). .

якщо операнд - це слово:

AX = (DX AX) / операнд

DX = залишок (модуль). .

• IDIV - поділ зі знаком:

якщо операнд - це байт:

AL = AX / операнд

AH = залишок (модуль). .

якщо операнд - це слово:

AX = (DX AX) / операнд

DX = залишок (модуль).

Третя група: INC, DEC, NOT, NEG

Ці типи операндів підтримуються:

REG

memory

REG: AX, BX, CX, DX, AH, AL, BL, BH, CH, CL, DH, DL, DI, SI, BP, SP.

memory: [BX], [BX + SI + 7], змінна, і т.п ...

Команди INC і DEC впливають тільки на ці прапори:

       ZF, SF, OF, PF, AF.

Команда NOT не впливає ні на які прапори!

Команда NEG впливає тільки на ці прапори:

       CF, ZF, SF, OF, PF, AF.

• NOT - інвертування кожного біта операнда.

• NEG - Змінює знак операнда (додаток до двох). Зазвичай вона інвертує кожен біт операнда, а потім додає до нього одиницю. Наприклад, 5 перетвориться в -5, а -2 перетворюється в 2.