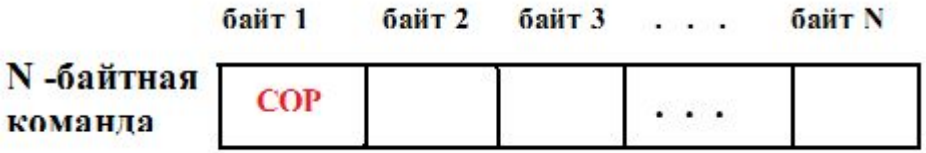
Лекция 5. Обзор системы команд процессора x386. Операнды в командах

Система команд – это аппаратно реализованный набор команд, которые может выполнять процессор

- Длина команды: от одного до нескольких байтов.
- Первый байт любой команды - код операции (Code of Operation) задает тип операции



- Последующие байты команды задают операнды для команды

Размещение многобайтной команды в памяти

- Адрес команды в памяти – это адрес ее первого байта (байт COP).
- Каждый последующий байт команды располагается в памяти по адресу, на 1 больше предыдущего

Пример: Пусть, CS:0007 - адрес 4-байтной команды (0007 - 16–разрядный внутрисегментный адрес)

Адреса байтов памяти, которые занимает команда?
Байты памяти с адреса CS:0007 по CS: 000A

Основные группы команд

- Команды пересылки данных. Перемещают (копируют) данные между регистрами процессора, памятью и портами. Не влияют на флаги.
- Команды двоичной арифметики. Устанавливают арифметические флаги.
- Команды логических операций. Выполняют операции булевой алгебры. Устанавливаются некоторые флаги.
- Команды сдвигов. Выполняют сдвиг кода в регистре или памяти на заданное количество бит. Влияют на флаг CF.
- Команды битовых операций. Манипулируют с отдельным битом регистра или памяти
- Команды передачи управления, вызова процедур, прерываний. Их назначение - нарушить линейную последовательность чтения процессором команд из памяти.

Операнды в командах

- Команды без операндов

```
CLC     ; сброс флага CF
```

- Команды с 1-м операндом

```
INC   BX   ; увеличить на 1 код в регистре BX
```

- Команды с 2-мя операндами.

Результат записывается на место 1-го операнда

```
ADD  AL, BL     ; AL ← AL + BL
ADD  AL, 3      ; AL ← AL + 3
ADD  ds:[7], BL  ; ds:[0007]←ds:[0007]+BL
```

!! Недопустимое сочетание двух операндов: “память - память“

Типы операндов в командах

Операнд	Что указывается в команде?
Регистр	→ Имя регистра
Память	→ Адрес памяти
Непосредственный операнд	→ числовой или символьный код

Размер операндов

Операнд может быть размещен в регистре, в памяти или в самой команде – «непосредственный операнд».

Если один из операндов команды - регистр, а второй - память или непосредственный операнд, то длину операндов определяет формат регистра.

Примеры:

- mov al, ds:[1] ; чтение из памяти 1 байта. Операнды: регистр-память
- mov ax, ds:[1] ; чтение 2 байтов
- mov eax, ds:[1] ; чтение 4 байтов
- mov eax, 5 ; Операнды: регистр-непосредственный операнд.

При занесении непосредственного операнда в регистр EAX процессор расширит его до 4-байтного формата (00 00 00 05)

Обозначение операндов в описании системы команд

- *r ,sr, reg* — регистр;
- *m, mem* — адрес операнда в памяти;
- *i* — непосредственный операнд;
- *r/m* — регистр или память;
- *r/m/i* — регистр, адрес памяти или непосредственный операнд;
- *r8, r16, r32* — 8-, 16- , 32-разрядный регистр;
- *m8, m16, m32* — 8-, 16- , 32- разрядный операнд из памяти;
- *i8, i16, i32*— 8-ми, 16-ти , 32-х разрядный непосредственный операнд ;

Пример описания команды процессора

Команда	Как выполняется процессором
MOV r/m, r/m/i	; пересылка r/m ← r/m/i
MOVZX r16, r/m8	; пересылка r16 ← r/m8 с беззнаковым расширением
MOVSX r32, r/m8	; пересылка r32 ← r/m8 с расширением знаковым битом

Способы задания внутрисегментного адреса операнда

1. Прямая адресация

Внутрисегментный адрес операнда в команде задаете явно:

mov bx, es:[2]

2. Косвенная адресация

Внутрисегментный адрес (или его часть) записан в регистре. Процессор будет извлекать адрес из регистра и , возможно, вычислять окончательный внутрисегментный адрес

mov bx, es:[si] - внутрисегментный адрес будет взят процессором из регистра SI

mov bx, es:[si+5] - внутрисегментный адрес будет вычислен процессором, как сумма содержимого регистра SI и 5

Варианты косвенной адресации

Для косвенной адресации используют регистры: si, di, bx, bp.
Косвенная адресация имеет несколько вариантов:

1) [si+disp], [di+disp], [bx+disp], [bp+disp]
2) [bx+si+disp], [bx+di+disp], [bp+si+disp], [bp+di+disp]

Они отличаются использованием одного или пары регистров.
disp – это знаковое число (не обязательно)

Пример: пусть, BX=0001 SI=0004

По какому адресу процессор будет обращаться в память за байтом в команде

```
mov dh, ds:[bx+si-4]      ; dh ← ds: [1]
```

Указатель сегмента данных «по умолчанию»

Если в команде процессору не задан указатель на сегмент данных, процессор по умолчанию использует регистр DS.

Если это команда с косвенной адресацией через регистр BP – тогда использует регистр SS.

Примеры: В командах есть внутрисегментный адрес, но отсутствует указатель сегмена

выполнение команды процессором

```
mov  [1], dl        ;  ds:[1] □ dl
mov  [si-2],  bh    ;  ds:[si-2] □ bh
mov  [bp+4], bh     ;  ss:[bp+4] □ bh
```