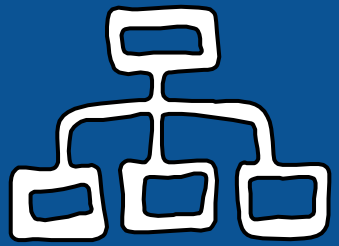
Архитектура ЭВМ и язык ассемблера

Семинар #29:

1. Регистры в `IA-32`, пересылка данных.
2. Арифметические операции: `ADD`, `SUB`, `MUL`, `DIV`.
3. Побитовые сдвиги, расширение типов.
4. Задачи на расширение типов.

23.01.2025

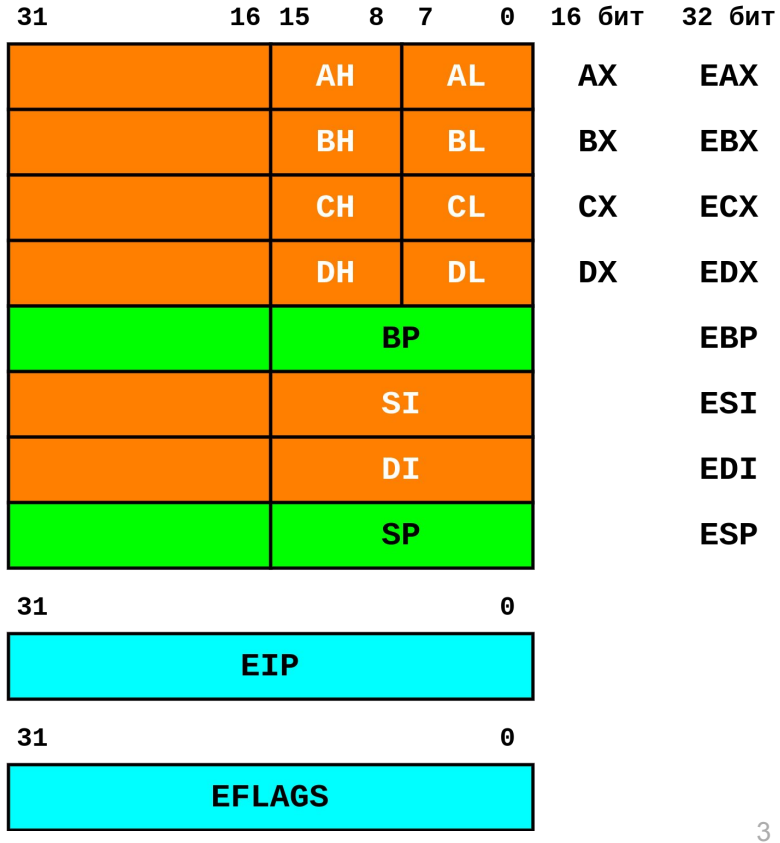
Регистры в IA-32, пересылка данных



Регистры в IA-32

EAX – Accumulator
EBX – Base register
ECX – Count register (счётчик цикла)
EDX – Data register
EBP – Base **P**ointer (стековый фрейм)
ESI – Source **I**ndex
EDI – Destination **I**ndex
ESP – Stack **P**ointer (вершина стека)

EIP – Instruction **P**ointer
(следующая инструкция)
EFLAGS – флаги для ветвлений



Пересылка данных

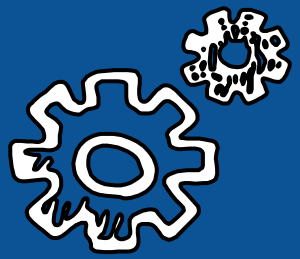
Формат пересылки

Операнд №1 (DST)	Операнд №2 (SRC)	Пример	Эквивалент на C
память	регистр	`mov byte [addr8], al`	`*addr8 = al`
регистр	память	`mov ax, word [addr16]`	`ax = *addr16`
регистр	константа	`mov ebx, DEADBEEFh`	`ebx = 0xDEADBEEF`
память	константа	`mov byte [addr8], 08h`	`*addr8 = 0x08`

Размер пересылки

1 байт	`mov byte [addr8], al`	`*addr8 = al`
2 байта (слово)	`mov ax, word [addr16]`	`ax = *addr16`
4 байта (двойное слово)	`mov edx, 08h`	`edx = 0x08`

Арифметические операции: ADD, SUB, MUL, DIV



Сложение и вычитание (ADD/SUB)

Формат операции ADD (SUB – аналогично)

Операнд №1 (DST)	Операнд №2 (SRC)	Пример	Эквивалент на C
регистр	регистр	`add eax, ebx`	`eax += ebx`
память	регистр	`add byte [addr8], al`	`*addr8 += al`
регистр	память	`add ax, word [addr16]`	`ax += *addr16`
регистр	константа	`add ebx, DEADBEEFh`	`ebx += 0xDEADBEEF`
память	константа	`add byte [addr8], 08h`	`*addr8 += 0x08`

Переполнение

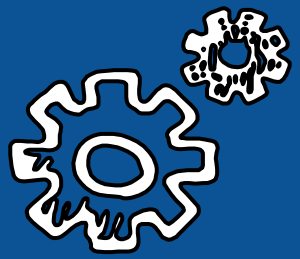
Происходит в рамках заданного регистра (ячейки памяти).

Умножение и деление (MUL/IMUL/DIV/IDIV)

Формат операнда	Пример	Вычисление
1 байт	`mul bl`	`ax = al * bl`
2 байта	`mul word [addr16]`	`dx:ax = ax * (*addr16)`
4 байта	`mul ecx`	`edx:eax = eax * ecx`
1 байт	`div bl`	`al = ax / bl` `ah = ax % bl`
2 байта	`div word [addr16]`	`ax = (dx:ax) / *addr16` `dx = (dx:ax) % *addr16`
4 байта	`div ecx`	`eax = (edx:eax) / ecx` `edx = (edx:eax) % ecx`

Операции: **знаковые** (IMUL/IDIV) и **беззнаковые** (MUL/DIV).

Побитовые сдвиги, расширение типов



Побитовые сдвиги, расширение типов

Тип	Направление	Пример	Примеры работы
Логический	Влево	`shl al, 3d`	`10000111 → 00111000`
Логический	Вправо	`shr al, 3d`	`10000111 → 00010000`
Арифметический	Влево	`sal al, 3d`	`10000111 → 00111000`
Арифметический	Вправо	`sar al, 3d`	`01110001 → 00001110` `11110001 → 11111110`

Операция	Пример	Примеры работы

Беззнаковое расширение	`movzx ax, byte [addr16]`	`*addr16 = 0xABCD`
		`eax = 0x01234567`
		`eax' = 0x0000ABCD`

Знаковое расширение	`movsx ax, byte [addr16]`	`*addr16 = 0xABCD`
		`eax = 0x01234567`
		`eax' = 0xFFFFABCD`

Задачи на расширение типов

Вопросы?

Красивые иконки взяты с сайта handdrawngoods.com