Семинар 8: Intel x86 assembly

14 января, 2020

Ассемблер

- Процессор выполняет *инструкции*
- Выполнение инструкций — комплексный процесс
- Стадии выполнения: fetch, decode, execute, memory access, writeback
- Современные процессоры — *суперскалярные*, то есть выполняют несколько стадий одновременно
- Dynamic execution

Pipelining

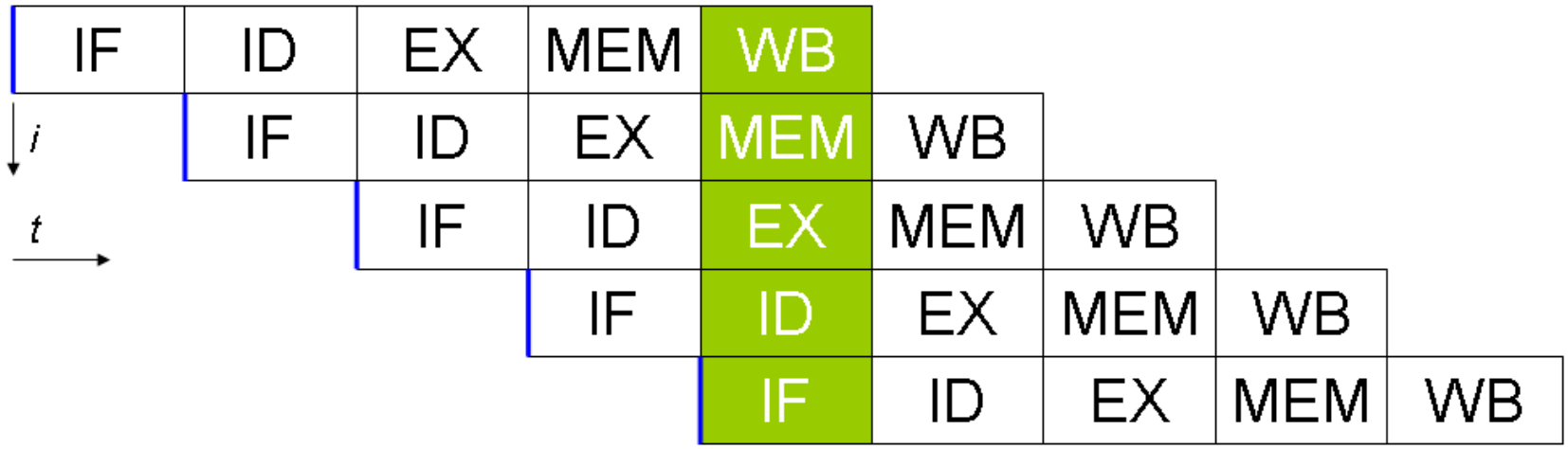


Figure: Pipeline современных процессоров (из Wikipedia)

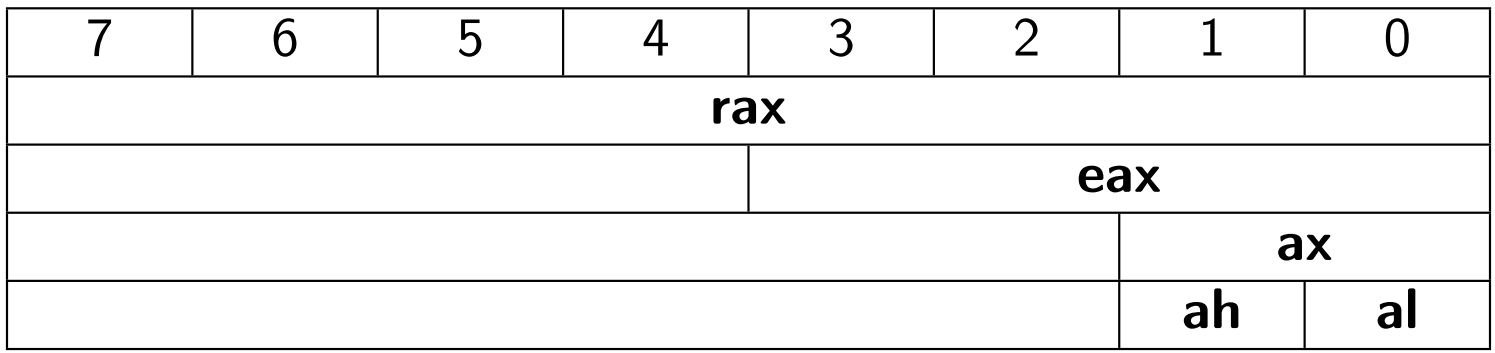
Ассемблер Intel x86

- Впервые появился в процессоре Intel 8086
- CISC-архитектура
- Два синтаксиса: Intel syntax, AT&T syntax

Регистры

- Регистры — очень быстрая, но маленькая память
- Под x86 — **32 бита**, под x86-64 — **64 бита**
- 32-bit registers: **eax**, **ebx**, **ecx**, **edx**, **esi**, **edi**
- 64-bit registers: **rax**, **rbx**, **rcx**, **rdx**, **rsi**, **rdi**, **r8-r15**
- Хранят числа в two's complement little endian представлении

Вложенность регистров



Инструкции

- Кодируются различным количеством байт: от 1 до 6
- При decode транслируются в более низкоуровневый μ-op
- Запись: LABEL: INST ARGS
- В основном все инструкции — бинарные
- Аргументом могут выступать либо регистр, либо память, либо *immediate value*
- По крайней мере один регистр должен быть среди аргументов!

Примеры инструкций

- add rax, rbx
- cmp rax, rbx
- inc r8

Различия синтаксисов

- Intel: INST DST, SRC
- AT&T: INST SRC, DST

EFLAGS

- Специальный регистр, который хранит *флаги*
- Каждому флагу соответствует определённый бит в EFLAGS
- Флаги выставляются в результате выполнения арифметических инструкций

Примеры флагов

- **ZF** выставляется, если был получен 0
- **SF** выставляется, если было получено отрицательное число
- **CF** выставляется при возникновении переноса в MSB (carry flag)
- **CF** выставляется при возникновении знакового переполнения (overflow flag)

Grazie!