

William Stallings

Computer Organization and Architecture

Chapter 9
Set Instruksi:
Karakteristik
dan Fungsi

Set instruksi ?

- ⌘ Kumpulan instruksi lengkap yang dimengerti oleh CPU
- ⌘ Kode mesin
- ⌘ Biner
- ⌘ Kode assembly

Elemen Instruksi

⌘ Operation code (Op code)

☑ Kerjakan

⌘ Source Operand reference

☑ Dengan data ini

⌘ Result Operand reference

☑ Simpan hasilnya kesini

⌘ Next Instruction Reference

☑ Setelah selesai, kerjakan ini ...

Operands ?

- ⌘ Main memory (or virtual memory or cache)
- ⌘ CPU register
- ⌘ I/O device

Penyajian Instruksi

- ⌘ Dlm kode mesin setiap instruksi memiliki pola-bit tertentu
- ⌘ Untuk konsumsi programmers disediakan penyajian simbolik
 - ☑ Contoh: ADD, SUB, LOAD
- ⌘ Operand juga disajikan secara simbolik
 - ☑ Contoh: ADD A,B

Jenis Instruksi

- ⌘ Data processing
- ⌘ Data storage (main memory)
- ⌘ Data movement (I/O)
- ⌘ Program flow control

Jumlah address (a)

⌘ 3 address

- ☒ Operand 1, Operand 2, Result
- ☒ Contoh: ADD A,B,C
- ☒ $a = b + c;$
- ☒ Jarang digunakan
- ☒ Perlu word yang panjang

Jumlah adres (b)

⌘ 2 adres

- ☑ Salah satu sebagai operand dan result
- ☑ Contoh: ADD A,B
- ☑ $a = a + b$
- ☑ Instruksi lebih pendek
- ☑ Diperlukan kerja ekstra
 - ☒ Temporary storage untuk menyimpan beberapa hasil operasi

Jumlah adres (c)

⌘ 1 adres

☑ Adres kedua Implicit

☑ Biasanya register (accumulator)

Jumlah address (d)

⌘ 0 (zero) address

- ☑ semua address implicit

- ☑ menggunakan stack

- ☑ Contoh:

- ☑ push a

- ☑ push b

- ☑ add

- ☑ pop c

- ☑ Berarti:

- ☑ $c = a + b$

Address banyak ? Sedikit ?

⌘ Address banyak

- ☑ Instruksi semakin kompleks
- ☑ perlu register banyak
- ☑ Program lebih pendek
- ☑ Lebih cepat ?

⌘ Address sedikit

- ☑ Instruksi lebih sederhana
- ☑ Eksekusi lebih cepat

Pertimbangan Perancangan (1)

⌘ Operation

- ☑ Berapa banyak operand?
- ☑ Apa saja operasi yang dikerjakan?
- ☑ Seberapa kompleks?

⌘ Jenis data

⌘ Format instruksi

- ☑ Panjang opcode
- ☑ Jumlah address

Pertimbangan Perancangan (2)

⌘ Register

- ☑ Jumlah register yang tersedia dlm CPU
- ☑ Operasi apa yg dpt dikerjakan oleh masing-masing registers?

⌘ Addressing modes (...)

⌘ RISC v CISC

Jenis Operand

⌘ Address

⌘ Number

☑ Integer/floating point

⌘ Character

☑ ASCII etc.

⌘ Logical Data

☑ Bits or flags

⌘ (Apa ada bedanya antara number dg character?)

Tipe Data pada Pentium

⌘ 8 bit Byte

⌘ 16 bit word

⌘ 32 bit double word

⌘ 64 bit quad word

⌘ Addressing menggunakan 8 bit unit

⌘ 32 bit double word dibaca pada adres yg habis dibagi dg 4

Tipe Data Spesifik

- ⌘ General – sembarang isi biner
- ⌘ Integer - single binary value
- ⌘ Ordinal - unsigned integer
- ⌘ Unpacked BCD - One digit per byte
- ⌘ Packed BCD - 2 BCD digits per byte
- ⌘ Near Pointer - 32 bit offset within segment
- ⌘ Bit field
- ⌘ Byte String
- ⌘ Floating Point

Tipe Data Floating Point pada Pentium

⌘ Stallings hal:324

Jenis Operasi

- ⌘ Data Transfer
- ⌘ Arithmetic
- ⌘ Logical
- ⌘ Conversion
- ⌘ I/O
- ⌘ System Control
- ⌘ Transfer of Control

Data Transfer

⌘ menentukan

☑ Source

☑ Destination

☑ Jumlah data

⌘ Beda instruksi untuk setiap data movement yang beda

☑ pada IBM 370

⌘ Atau satu instruction dengan address yg beda

☑ pada VAX

Arithmetic

⌘ Add, Subtract, Multiply, Divide

⌘ Signed Integer

⌘ Floating point ?

⌘ Termasuk operasi:

☒ Increment ($a++$)

☒ Decrement ($a--$)

☒ Negate ($-a$)

Logical

- ⌘ Bitwise operations

- ⌘ AND, OR, NOT

Conversion

⌘ Contoh: Biner ke Decimal

Input/Output

- ⌘ Tersedia instruksi khusus
- ⌘ Atau digunakan instruksi data movement (memory mapped)
- ⌘ Atau dikerjakan oleh controller (DMA)

Systems Control

- ⌘ Privileged instructions
- ⌘ CPU harus berada pada state tertentu
 - ☑ Ring 0 pada 80386+
 - ☑ Kernel mode
- ⌘ Digunakan oleh operating systems

Transfer of Control

⌘ Branch

- ☒ Contoh: branch to x if result is zero

⌘ Skip

- ☒ Contoh: increment and skip if zero

- ☒ ISZ Register1

- ☒ Branch xxxx

- ☒ ADD A

⌘ Subroutine call

- ☒ interrupt call

Baca sendiri

- ⌘ Jenis operasi pada Pentium and PowerPC
- ⌘ Stallings hal:338

Byte Order

(bagian dari chips?)

- ⌘ Bagaimana urutan bilangan yang lebih dari 1 byte disimpan/dibaca
- ⌘ contoh (dalam hex)
- ⌘ Bilangan 12345678 dapat disimpan pada lokasi 4x8bit sbb:
 - ⌘

Contoh Byte Order

⌘ Address	Value (1)	Value(2)
⌘ 184	12	78
⌘ 185	34	56
⌘ 186	56	34
⌘ 186	78	12

⌘ top down atau bottom up?

Penamaan Byte Order

- ⌘ Problem ordering dinamakan Endian
- ⌘ LSB pada posisi paling kiri (pada adres terkecil)
- ⌘ Dinamakan big-endian
- ⌘ LSB pada posisi paling kanan (pada adres terbesar)
- ⌘ Dinamakan little-endian

Standard...?

- ⌘ Pentium (80x86), VAX adalah little-endian
- ⌘ IBM 370, Motorola 680x0 (Mac), dan RISC yang lain, adalah big-endian
- ⌘ Internet menggunakan big-endian
 - ☑ Membuat program untuk Internet pada PC lebih susah.
 - ☑ WinSock menyediakan htoi (Host to Internet) dan itoh (Internet to Host) untuk keperluan konversi