

Set Instruksi

Definisi

Set intruksi adalah sekumpulan lengkap intruksi yang dimengerti oleh CPU, set intruksi disebut bahasa mesin karena aslinya biner kemudian direpresentasikan sebagai bahasa assembly agar bisa lebih mudah dimengerti oleh manusia.

Set instruksi ?

- ⌘ Operasi dari CPU ditentukan oleh instruksi-instruksi yang dilaksanakan atau dijalankannya. Instruksi ini sering disebut sebagai instruksi mesin (*machine instructions*) atau instruksi komputer (*computer instructions*).
- ⌘ Kumpulan dari instruksi-instruksi yang berbeda yang dapat dijalankan oleh CPU disebut set Instruksi (*Instruction Set*).

Elemen Instruksi

⌘ Operation code (Op code)

☒ Kerjakan, **menentukan operasi yang akan dilaksanakan**

⌘ Source Operand reference

☒ Dengan data ini, **merupakan input bagi operasi yang akan dilaksanakan**

⌘ Result Operand reference

☒ Simpan hasilnya kesini, **merupakan hasil dari operasi yang dilaksanakan**

⌘ Next Instruction Reference

☒ Setelah selesai, kerjakan ini ... , **memberitahu CPU untuk mengambil (*fetch*) instruksi berikutnya setelah instruksi yang dijalankan selesai.**

Operands ?

- ⌘ Ingat: Semua instruksi dijalankan dalam CPU
- ⌘ Rata-rata operasi hanya membutuhkan register sebagai tempat membaca /menyimpan operand
- ⌘ Adakalanya juga register tidak berisi operand tapi menunjuk ke tempat penyimpanan lainnya (memory, cache, modul I/O)
- ⌘ Source dan results operands dapat berupa salah satu di antara tiga jenis berikut :
 - ☒ Main memory (or virtual memory or cache)
 - ☒ CPU register
 - ☒ I/O device

Format Instruksi

- ⌘ Suatu instruksi terdiri dari beberapa *field* yang sesuai dengan elemen dalam instruksi tersebut. Layout dari suatu instruksi sering disebut sebagai Format Instruksi (*Instruction Format*).

OPCODE	OPERAND REFERENCE	OPERAND REFERENCE
--------	----------------------	----------------------

Penyajian Instruksi

- ⌘ Dlm kode mesin setiap instruksi memiliki pola-bit tertentu yang unik
- ⌘ Agar dapat dimengerti manusia, dibuatlah representasi simbolik instruksi, biasanya berupa singkatan (disebut *mnemonic*)
 - ☑ Contoh: ADD, SUB, LOAD
- ⌘ Sedangkan Operand juga disajikan secara simbolik
 - ☑ Contoh: ADD **A, B**

Jenis Instruksi

- ⌘ Data processing, ***Arithmetic dan Logic Instructions***
- ⌘ Data storage (main memory), ***Memory instructions***
- ⌘ Data movement (I/O), ***I/O instructions***
- ⌘ Program flow control, ***Test and branch instructions***

Jumlah adres (a)

⌘ 3 adres

- ☒ Bentuk umum: [OPCODE] [AH], [AO1], [AO2]
- ☒ Satu alamat hasil, dua alamat operand
- ☒ Misal: **SUB Y, A, B**
 - ☒ Bentuk algoritmik: $Y \leftarrow A - B$
 - ☒ Arti: Kurangkan isi Reg A dengan isi Reg B, kemudian simpan hasilnya di Reg Y.
- ☒ Bentuk ini tidak umum digunakan di komputer
- ☒ Mengoperasikan banyak register sekaligus
- ☒ Program lebih pendek

Jumlah adres (b)

⌘ 2 adres

- ☑ Bentuk umum: [OPCODE] [AH], [AO]
- ☑ Satu alamat hasil merangkap operand, satu alamat operand
- ☑ Misal: **SUB Y, B**
 - ☒ Bentuk algoritmik: $Y \leftarrow Y - B$
 - ☒ Arti: Kurangkan isi Reg Y dengan isi Reg B, kemudian simpan hasilnya di Reg Y.
- ☑ Bentuk ini masih digunakan di komputer sekarang
- ☑ Mengoperasikan lebih sedikit register, tapi panjang program tidak bertambah terlalu banyak

Jumlah adres (c)

⌘ 1 adres

- ☒ Bentuk umum: [OPCODE] [AO]
- ☒ Satu alamat operand, hasil disimpan di accumulator
- ☒ Misal: **SUB B**
 - ☒ Bentuk algoritmik: $AC \leftarrow AC - B$
 - ☒ Arti: Kurangkan isi Acc. dengan isi Reg B, kemudian simpan hasilnya di Acc.
- ☒ Bentuk ini digunakan di komputer jaman dahulu
- ☒ Hanya mengoperasikan satu register, tapi program menjadi bertambah panjang

Jumlah adres (d)

⌘ 0 (zero) adres

- ☒ Bentuk umum: [OPCODE] [O]
- ☒ Semua alamat operand implisit, **disimpan dalam bentuk stack**. Operasi yang biasanya membutuhkan 2 operand, akan mengambil isi stack paling atas dan di bawahnya
- ☒ Misal: **SUB**
 - ☒ Bentuk algoritmik: $S[\text{top}] \leftarrow S[\text{top}-1] - S[\text{top}]$
 - ☒ Arti: Kurangkan isi Stack no.2 dari atas dengan isi Stack paling atas, kemudian simpan hasilnya di Stack paling atas
- ☒ Ada instruksi khusus Stack: **PUSH** dan **POP**
- ☒ Contoh lain:
 - push a
 - push b
 - add
 - pop c
- ☒ Berarti:
 $c = a + b$

Contoh Format Instr 3 Alamat

⌘ A, B, C, D, E, T, Y adalah register

⌘ Program: $Y = (A - B) / (C + D \times E)$

⌘ SUB Y, A, B $Y \leftarrow A - B$

⌘ MPY T, D, E $T \leftarrow D \times E$

⌘ ADD T, T, C $T \leftarrow T + C$

⌘ DIV Y, Y, T $Y \leftarrow Y / T$

⌘ Memerlukan 4 operasi

Contoh Format Instr 2 Alamat

- ⌘ A, B, C, D, E, T, Y adalah register
 - ⌘ Program: $Y = (A - B) / (C + D \times E)$
 - ⌘ MOVE Y, A $Y \leftarrow A$
 - ⌘ SUB Y, B $Y \leftarrow Y - B$
 - ⌘ MOVE T, D $T \leftarrow D$
 - ⌘ MPY T, E $T \leftarrow T \times E$
 - ⌘ ADD T, C $T \leftarrow T + C$
 - ⌘ DIV Y, T $Y \leftarrow Y / T$
-
- ⌘ Memerlukan 6 operasi

Contoh Format Instr 1 Alamat

- ⌘ A, B, C, D, E, Y adalah register
 - ⌘ Program: $Y = (A - B) / (C + D \times E)$
 - ⌘ LOAD D $AC \leftarrow D$
 - ⌘ MPY E $AC \leftarrow AC \times E$
 - ⌘ ADD C $AC \leftarrow AC + C$
 - ⌘ STOR Y $Y \leftarrow AC$
 - ⌘ LOAD A $AC \leftarrow A$
 - ⌘ SUB B $AC \leftarrow AC - B$
 - ⌘ DIV Y $AC \leftarrow AC / Y$
 - ⌘ STOR Y $Y \leftarrow AC$
-
- ⌘ Memerlukan 8 operasi

Contoh Format Instr 0 Alamat

⌘ A, B, C, D, E, Y adalah register

⌘ Program: $Y = (A - B) / (C + D \times E)$

PUSH A	$S[\text{top}] \leftarrow A$
PUSH B	$S[\text{top}] \leftarrow B$
SUB	$S[\text{top}] \leftarrow A - B$
PUSH C	$S[\text{top}] \leftarrow C$
PUSH D	$S[\text{top}] \leftarrow D$
PUSH E	$S[\text{top}] \leftarrow E$
MPY	$S[\text{top}] \leftarrow D \times E$
ADD	$S[\text{top}] \leftarrow C + S[\text{top}]$
DIV	$S[\text{top}] \leftarrow (A - B) / S[\text{top}]$
POP Y	$\text{Out} \leftarrow S[\text{top}]$

⌘ Memerlukan 10 operasi

$$\text{Kerjakan } X = (A + B \times C) / (D - E \times F)$$