

Set Instruksi

<u>Instruction</u>		<u>Comment</u>
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	$Y \leftarrow Y \div T$

(a) Three-address instructions

<u>Instruction</u>		<u>Comment</u>
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	$Y \leftarrow Y \div T$

(b) Two-address instructions

<u>Instruction</u>		<u>Comment</u>
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	$AC \leftarrow AC \div Y$
STOR	Y	$Y \leftarrow AC$

(c) One-address instructions

- ❖ **Karakteristik dan Fungsi**
- ❖ **Elemen Instruksi**
- ❖ **Jenis Instruksi**
- ❖ **Format Instruksi**

Karakteristik dan Fungsi

Dalam melakukan prosesnya CPU secara otomatis akan menjalankan satu atau beberapa instruksi dalam satu waktu (instruksi mesin/instruksi komputer).

Elemen-elemen Instruksi

- 1. Operation code (Op code) :** berisikan perintah yang akan dikerjakan
 - 2. Source Operand reference :** letak data yang harus dikerjakan sesuai perintah
 - 3. Result Operand reference :** Penyimpan hasil operasi
 - 4. Next Instruction Reference :** Perintah selanjutnya yang harus diselesaikan
-

Elemen-elemen Instruksi

- **Kode Operasi (OpCode)**
Menspesifikasikan operasi yang akan dilakukan.
 - **Referensi Operand Sumber**
Operasi dapat mencakup satu atau lebih operand sumber (input bagi operasi).
 - **Referensi Operand Hasil**
Operasi dapat menghasilkan output
 - **Referensi Operand Selanjutnya**
Memberitahukan CPU posisi instruksi berikutnya yang harus diambil setelah menyelesaikan eksekusi suatu instruksi
-

Elemen-elemen Instruksi

Dalam sebuah instruksi, tidak harus semua elemen ini dicantumkan, tergantung kebutuhan dan jenis instruksinya. Semua instruksi dijalankan didalam CPU dan umumnya menggunakan register sebagai tempat membaca/menyimpan operand meskipun tidak selamanya register berisikan operand melainkan menunjuk kepada salah satu media penyimpan lainnya missal: memori, cache, modul I/O.

Elemen-elemen Instruksi

Source and result operand dari suatu operasi dapat berada pada tiga tempat berikut:

1. Memori utama atau memori virtual
 2. CPU register
 3. Perangkat I/O
-

Representasi Instruksi

Pada bahasa mesin, setiap instruksi berbentuk pola bit biner yang unik untuk itu agar dapat dimengerti oleh pengguna maka dibuatlah berbagai instruksi menggunakan singkatan sebagai simbolik yang disebut sebagai *mnemonic*. Missal: ADD, SUB, MPY, DIV, LOAD, STOR. Operand dapat direpresentasikan dengan menggunakan simbolik missal: ADD A,B

Representasi Instruksi

Tambahkan nilai yang berada pada B ke dalam register A dan simpan hasilnya di register A. seorang programmer diperkenankan menggunakan symbol dan menetapkan lokasi untuk merepresentasikan bahasa mesin dimana setiap opcode symbol direpresentasikan dengan angka biner yang selalu tetap.

Jenis Instruksi

- Instruksi Operate
Menentukan operand untuk operasi mesin.
 - Instruksi Arimetik, Logika, dan Shift.
 - Instruksi Akses Memory
 - Instruksi Kontrol
 - Instruksi Khusus
-

Jenis Instruksi

- Instruksi Operate
Menentukan operand untuk operasi mesin.

Operand dibagi menjadi 2 jenis :

1. Implisit
 2. Eksplisit
-

Jenis Instruksi

1. Implisit

Operand yang ditentukan oleh opcode dan tidak memiliki spesifikasi yang lain.

Contoh :

Instruksi CLEAR CARRY

Jenis Instruksi

2. Eksplisit

Operand yang ditentukan oleh instruksi dalam field selain field opcode.

Contoh :

Instruksi ADD

Jenis Instruksi

- Instruksi Arimetik, Logika, dan Shift.

Aritmetik : Menyusun flag status prosesor guna menunjukkan hasil operasi.

Jenis Instruksi

Logika : Menyusun kode kondisi hanya untuk result 0 dan 1 (false atau true).

Shift : Untuk memperluas jangkauan presisi bilangan bulat (integer).

Jenis Instruksi

➤ Instruksi Akses Memory

LOAD dan STORE : Untuk memuat dan menyimpan isi register ke dalam memory.

LOAD ADDRESS : Untuk mendapatkan alamat data yang efektif atau alamat instruksi itu sendiri.

Sinkronisasi Proses : Untuk mensinkronisasikan eksekusi antara proses active dan proses inactive.

Jenis Instruksi

➤ Instruksi Kontrol

Branch Test : Menentukan apakah sebuah proses membutuhkan percabangan.

Branch Kondisional : Menjalankan awal branch hanya pada saat kondisi test tertentu telah memadai.

Jenis Instruksi

Branch Tak-kondisional : Semua operasi akan selalu dijalankan dengan menggunakan pencabangan.

Hubungan Subroutine : Digunakan untuk menghubungkan antar prosedur dalam sebuah proses.

Jenis Instruksi

➤ Instruksi Khusus

Input/Output : Digunakan untuk mengontrol peralatan I/O yang digunakan dalam suatu proses.

Privileged : Digunakan untuk menjaga sistem dari manipulasi software pemakai.

Jenis Instruksi

1. Data processing
 - Aritmetik misal : ADD, SUB dll.
 - Logic missal : AND, OR, NOT, SHR, dll.
 - Konversi data
 2. Data storage (memori)
 - Transfer data missal : STOR, LOAD, MOVE, dll.
 3. Data movement: Input dan Output ke modul I/O
 4. Program flow control: JUMP, HALT, dll
-

Format Instruksi

- ❖ Format Instruksi 3 Alamat
 - ❖ Format Instruksi 2 Alamat
 - ❖ Format Instruksi 1 Alamat
-

Format Instruksi

❖ Format Instruksi 3 Alamat :

Bentuk umum: [OPCODE] [AH], [AO1], [AO2] merupakan suatu instruksi dengan satu alamat hasil dan dua alamat operand.

Format Instruksi

❖ Format Instruksi 3 Alamat :

Misal: SUB Y,A,B bentuk algoritmik $Y \leftarrow A - B$ yang mengandung arti kurangkan isi register A dengan isi register B kemudian simpan hasilnya di register Y. Dengan format seperti ini program lebih pendek dan mengoperasikan banyak register sekaligus dan bentuk ini tidak umum digunakan di komputer.

Format Instruksi

Contoh Format Instruksi 3 Alamat

A, B, C, D, E, T, Y adalah register

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

SUB Y, A, B	bentuk algoritmik $Y \leftarrow A - B$
MPY T, D, E	bentuk algoitmik $T \leftarrow D \times E$
ADD T, T, C	bentuk algoritmik $T \leftarrow T + C$
DIV Y, Y, T	bentuk algoritmik $Y \leftarrow Y / T$

Jadi Memerlukan = 4 operasi

Format Instruksi

❖ Format Instruksi 2 Alamat :

Bentuk umum: [OPCODE] [AH], [AO]
merupakan instruksi dengan satu alamat operand dan satu alamat hasil merangkap salah satu operand lainnya.

Format Instruksi

❖ Format Instruksi 2 Alamat :

Misal: **SUB Y, B** bentuk algoritmik $Y \leftarrow Y - B$ yang mengandung arti kurangkan isi register Y dengan isi register B kemudian simpan hasilnya di register Y.

Dengan format instruksi seperti ini panjang program tidak bertambah terlalu banyak tetapi mengoperasikan lebih sedikit register dan bentuk instruksi seperti ini masih digunakan di komputer.

Format Instruksi

Contoh Format Instruksi 2 Alamat :

A, B, C, D, E, T, Y adalah register

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

MOVE Y, A	bentuk algoritmik $Y \leftarrow A$
SUB Y, B	bentuk algoritmik $Y \leftarrow Y - B$
MOVE T, D	bentuk algoritmik $T \leftarrow D$
MPY T, E	bentuk algoritmik $T \leftarrow T \times E$
ADD T, C	bentuk algoritmik $T \leftarrow T + C$
DIV Y, T	bentuk algoritmik $Y \leftarrow Y / T$

Jadi Memerlukan = 6 operasi

Format Instruksi

❖ Format Instruksi 1 Alamat :

Bentuk umum: [OPCODE] [AO], merupakan instruksi dengan satu alamat operand dan hasil disimpan di accumulator.

Misal: **SUB B** bentuk algoritmik $AC \leftarrow AC - B$ mengandung arti kurangkan isi Acc dengan isi register B kemudian simpan hasilnya di Acc.

Format Instruksi

❖ Format Instruksi 1 Alamat :

Dengan format instruksi seperti ini program menjadi bertambah panjang sementara register yang digunakan hanya satu dan bentuk ini digunakan pada komputer jaman dahulu.

Misal: **SUB**

bentuk algoritmik: $S[\text{top}] \leftarrow S[\text{top}-1] - S[\text{top}]$
mengandung arti kurangkan isi stack no.2 dari atas dengan isi stack paling atas kemudian simpan isi hasilnya di stack paling atas. Untuk instruksi PUSH dan POP mempunyai cara kerja yang berbeda.

Format Instruksi

Contoh Format Instruksi 1 Alamat :

A, B, C, D, E, Y adalah register

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

LOAD D	bentuk algoritmik	$AC \leftarrow D$
MPY E	bentuk algoritmik	$AC \leftarrow AC \times E$
ADD C	bentuk algoritmik	$AC \leftarrow AC + C$
STOR Y	bentuk algoritmik	$Y \leftarrow AC$
LOAD A	bentuk algoritmik	$AC \leftarrow A$
SUB B	bentuk algoritmik	$AC \leftarrow AC - B$
DIV Y	bentuk algoritmik	$AC \leftarrow AC / Y$
STOR Y	bentuk algoritmik	$Y \leftarrow AC$

Jadi Memerlukan = 8 operasi

Format Instruksi

Contoh Format Instruksi 0 Alamat :

A, B, C, D, E, Y adalah register

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

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

Memerlukan 10 operasi

Format Instruksi

Latihan soal :

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

Desain Instruksi

Design dari sebuah intruksi sangat kompleks karena mempengaruhi banyak aspek dalam sistem komputer. Beberapa hal yang paling fundamental dalam prose perancangan antara lain meliputi:

- ✓ Operation repertoire berkenaan dengan seberapa banyak dan jenis operasi apa yang yang harus tersedia dan sekomplek apakah operasi itu seharusnya.
-

Desain Instruksi

- ✓ Data types berkenaan dengan jenis data yang dapat disediakan.
 - ✓ Instruction format berkenaan dengan panjang instruksi (bits), nomor alamat, ukuran field dan sebagainya.
 - ✓ Register berkenaan dengan jumlah register yang dapat digunakan oleh instruksi dan masing-masing fungsinya.
 - ✓ Addressing digunakan untuk menspesifikasikan alamat operand.
-

Set Instruksi

TERIMAKASIH

Kuis Set Instruksi – Pertemuan ke-11

1. Jelaskan elemen-elemen Set Instruksi?
2. Uraikan perbedaan format instruksi 2 Alamat dan 3 Alamat?
3. Buatlah program format instruksi $Y = (A + B \times C) / (D - E \times F)$?

Note: Kerjakan di selembar kertas/buku catatan mata kuliah Arsikom kalian!