1. Karakteristik Instruksi Mesin

Menurut Kamus Besar Bahasa Indonesia, Karakteristik adalah ciri-ciri khusus atau mempunyai sifat khas sesuai dengan perwatakan tertentu. Instruksi adalah perintah atau arahan (untuk melakukan suatu pekerjaan atau melaksanakan suatu tugas). Mesin adalah perkakas untuk menggerakkan, atau membuat sesuatu yang dijalankan dengan roda-roda dan digerakkan oleh tenaga manusia atau motor penggerak yang menggunakan bahan bakar minyak atau tenaga alam.

Jadi, karakteristik-karakteristik instruksi mesin adalah ciri-ciri khusus atau sifat khas yang dimiliki oleh instruksi-instruksi atau kode operasi dalam pemrograman komputer.. Operasi CPU ditentukan oleh instruksi-instruksi yang dieksekusinya. Instruksi-instruksi ini dikenal sebagai intruksi mesin atau instruksi computer. Set fungsi dari instruksi-instruksi yang berbeda yang dapat di eksekusi oleh CPU dikenal sebagai set instruksi CPU.

A. Elemen-elemen Instuksi Mesin

Setiap instruksi harus terdiri dari informasi yang diperlukan oleh CPU untuk dieksekusi. Gambar langkah-langkah yang terdapat dalam eksekusi instruksi dan bentuk elemen-elemen instruksi mesin, adalah sebagai berikut :

• **Kode Operasi** : menentukan operasi-operasi yang akan dilakukan (misalnya: ADD,I/O). Operasi itu dispesifilan oleh sebuah kode biner, dikenal sebagai kode operasi.

• **Acuan Operand Sumber** : Operasi dapat melibatkan satu atau lebih operand sumber, dengan kata lain, operand adalah input bagi operasi.

• **Acuan Operand Hasil**: Operasi dapat menghasilkan sebuah hasil.

• Acuan Instruksi Berikutnya: Elemen ini memberitahukan CPU posisi instruksi berikutnya yang harus diambil setelah menyelesaikan eksekusi suatu instruksi. Instuksi berikutnya yang akan diambil berada di memori utama atau pada system memori virtual, akan berada baik di dalam memori utama atau memori sekunder. Umumnya, instruksi yang akan segera diambil selanjutnya, berada setelah instruksi saat itu. Ketika acuan eksplisit dibutuhkan, maka alamat memori utama atau alamat memori virtual harus disiapkan. Operand sumber dan hasil dapat berada di salah satu dari ketiga daerah berikut ini:

• **Memori Utama atau Memori Virtual**: Dengan adanya acuan instruksi berikutnya, maka alamat memori utama atau memori virtual harus diketahui.

• **Register CPU**: Dengan suatu pengecualian yang jarang terjadi, CPU terdiri dari sebuah register atau lebih yang dapat diacu oleh instruksi-instruksi mesin. Bila hanya terdapat sebuah register saja, maka acuan ke instruksi tersebut dapat berbentuk implicit. Sedangkan jika terdapat lebih dari satu register, maka setiap register diberi nomor yang unik, dan instruksi harus terdiri dari nomor register yang dimaksud.

• **Perangkat I/O**: Instruksi harus menspesifikan modul I/O dan perangkat yang diperlukan oleh operasi. Jika digunakan I/O memori terpetakan, maka perangkat ini merupakan memori utama atau memori virtual.

B. Representasi Instruksi

Di dalam computer, instruksi dipresentasikan oleh sehimpunan bit. Himpunan bit ini dibagi menjadi beberapa bidang, dengan bidang-bidang ini berkaitan dengan elemen-elemen yang akan memuat instruksi. Layout instruksi ini dikenal sebagai bentuk instruksi. Contoh yang sederhana ditunjukkan pada gambar. Pada sebagian besar set instruksi, dapat digunakan lebih dari satu bentuk. Selama berlangsungnya eksekusi instruksi, instruksi dibaca ke dalam register instruksi yang terdapat dalam CPU. Untuk melakukan operasi yang diperlukan, CPU harus dapat mengeluarkan data dari berbagai bidang instruksi. Opcode direpresentasikan dengan singkatan-singkatan, yang disebut mnemorik, yang mengindikasikan operasi, contohnya adalah:

ADD Add (Menambahkan)
SUB Substract (Pengurangan)
MPY Multiply (Perkalian)
DIV Divide (Pembagiano
LOAD Muatkan data data dari memori
STOR Simpan data ke memori

Operand-operand juga direpresentasikan secara simbolik. Misalnya instruksi ADD R,Y Berarti tambahkan niali yang terdapat pada lokasi Y ke isi register R. Dalam contoh ini, Y berkaitan dengan alamat lokasi di dalam memori, dan R berkaitan dengan register tertentu. Perlu dicatat bahwa operasi dilakukan terhadap isi alamat, bukan terhadap alamatnya.

Sehingga adalah mungkin untuk menuliskan program bahasa mesin dalam bentuk simbolik. Setiap opcode simbolik memiliki representasi biner yang tetap, dan pemrograman dapat menetapkan masing-masing operand simbolik. Misalnya, pemrograman dapat memulainya dengan definisi-definisi:
X=523
Y=514
dan seterusnya. Sebuah program yang sederhana akan menerima input simbol ini, kemudian mengkonversiakn opcode dan acuan operand menjadi bentuk biner, dan akhirnya membentuk instruksi mesin biner.

C. Jenis-Jenis Instruksi

Sebuah instuksi yang dapat diekspresikan dalam bahasa BASIC atau FORTRAN. X = X+Y Pernyataan ini menginstruksiakna komputer untuk menambahkan nilai yang tersimpan di Y ke nilai yang tersimpan di X dan menyimpan hasilnya di X. Variabel X dan Y berkorespondensi dengan lokasi 513 dan 514. Jika kita mengasumsikan set instruksi mesin yang sederhana, maka operasi ini dapat dilakukan dengan tiga buah instruksi:

1. Muatkan sebuah register dengan isi lokasi memori 513
2. Tambahkan isi lokasi memori ke register
3. Simpan isi register di lokasi memori 513

Suatu komputer harus memiliki set instruksi yang memungkinkan pengguna untuk memformulasikan pengolahan data atau dengan memperhatikan kemampuan pemrograman bahasa tingkat tinggi. Agar dapat dieksekusi, setiap program yang ditulis dalam bahasa program tingkat tinggi harus diterjemahkan ke dalam bahasa mesin. Jadi, set instruksi mesin harus dapat mengekspresikan setiap instruksi bahas atingkat tinggi.

Adapun Jenis-jenis instrusi sebagai berikut:
- Pengolahan Data : Instrusi-instruksi aritmatika dan logika
- Penyimpanan Data : Instriksi-instruksi memori
- Perpindahan Data : Instruksi I/O
- Kontrol : Instruksi pemeriksaan dan percabangan

D. Jumlah Alamat

Salah satu cara tradisional dalam menjelaskan arsitektur prosesor adalah dengan memakai jumlah alamat yang terdapat pada masing-masing instruksi. Instruksi

aritmatika dan logika memerlukan operand yang berjumlah banyak. Secara virtual, seluruh operasi eritmatika dan logika merupakan uner/unary (satu operand) atau biner (dua operand). Dengan demikian, memerlukan maksimum dua alamat untuk acuan operand. Hasil sebuah operasi akan memerlukan alamat ketiga.

Dengan demikian, instruksi perlu memiliki empat buah acuan alamat: dua buah operand, sebuah hasil operasi, dan sebuah alamat instruksi berikutnya. Sebagian besar CPU merupakan variasi satu, dua, atau tiga alamat dengan alamat instruksi berikutnya merupakan implisit (diperoleh dari pencacah program). Format tiga alamat tidak umum digunakan, karena instruksi-instruksi tersebut memerlukan bentuk instruksi yang lebih relatif lebih panjang untuk menampung acuan-acuan tiga alamat. Sedangkan bentuk dua alamat mengurangi kebuatuahan ruang akan tetapi menimbulkan kesulitan. Instruksi yang lebih sederhana adalah instruksi satu alamat. Agar alamat ini dapat berfungsi, alamat perlu diimplisitkan.

E. Rancangan Set Intruksi

Salah satu hal yang paling menarik tentang rancangan komputer adalah rancangan set instruksi. Karena rancangan ini mempengaruhi banak aspek sistem komputer, maka rancangan set instruksi sangat kompleks. Set instruksi menentukan banyak fungsi yang akan dilakukan oleh CPU dan karena itu memiliki efek yang sangat menentukan implementasi CPU. Set instruksi merupakan alat bagi pemrogram untuk mengontrol CPU. Dengan demikian, kebutuhan-kebutuhan pemrogram harus menjadi bahan pertimbangan dalam merancang set instruksi. Masalah rancangan fundamental yang paling signifikan meliputi:

1. Repertori Operasi: Berapa banyak dan opersai-operasi apa yang harus disediakan, dan sekompleks apakah operasi itu seharusnya.
2. Jenis data : berbagai jenis data pada saat operasi dijalankan
3. Bentuk instruksi : Panjang instruksi (dalam bit), jumlah alamat, ukuran bidang, dan sebagainya.
4. Register : Jumlah register CPU yang dapat diacu oleh instruksi, dan fungsinya.
5. Pengalamatan: Mode untuk menspesifikasikan alamat suatu operand.

Masalah-masalah ini saling berkaitan dan harus diperhatikan dalam merancang set instruksi.

2. Tipe — Tipe Operand

Operand adalah sebuah objek yang ada pada operasi matematika yang dapat digunakan untuk melakukan operasi. Operand atau operator dalam bahasa C berbentuk simbol bukan berbentuk keyword atau kata yang biasa ada di bahasa pemrograman lain. Simbol yang digunakan bukan karakter yang ada dalam abjad tapi ada pada keyboard kita seperti =,,* dan sebagainya.

Tipe-tipe operand diantaranya :

1. Addresses (akan dibahas pada addressing modes)

2. Numbers :
- Integer or fixed point
- Floating point
- Decimal (BCD)

3. Characters :
- ASCII
- EBCDIC

4. Logical Data : Bila data berbentuk binary: 0 dan 1
Jenis-jenis operator adalah sebagai berikut :

1. Operator Aritmetika
Operator untuk melakukan fungsi aritmetika seperti : +(penjumlahan), — (mengurangkan), * (mengalikan), / (membagi).

2. Operator relational
Operator untuk menyatakan relasi atau perbandingan antara dua operand, seperti > (lebih besr), =(lebih besar atau sama), <= (lebih kecil atau sama), == (sama), != (tidak sama).

3. Operator Logika
Operator untuk merelasikan operand secara logis seperti && (and), || (or), !(not).

3. Tipe — Tipe Operasi
Dalam perancangan arsitektur komputer, jumlah kode operasi akan sangat berbeda untuk masing-masing komputer, tetapi terdapat kemiripan dalam jenis operasinya.

Jenis operasi computer :

- Transfer data

1. Menetapkan lokasi operand sumber dan operand tujuan.

2. Lokasi-lokasi tersebut dapat berupa memori, register atau bagian paling atas daripada stack.

3. Menetapkan panjang data yang dipindahkan.

4. Menetapkan mode pengalamatan.

- Aritmatika

Tindakan CPU untuk melakukan operasi arithmetic :

1. Transfer data sebelum atau sesudah.
2. Melakukan fungsi dalam ALU.
3. Menset kode-kode kondisi dan flag.

-Logika

Tindakan CPU sama dengan arithmetic
Operasi set instruksi untuk operasi logical :

1. AND, OR, NOT, EXOR
2. COMPARE : melakukan perbandingan logika.
3. TEST : menguji kondisi tertentu.
4. SHIFT : operand menggeser ke kiri atau kanan menyebabkan

konstanta pada ujung bit.

5. ROTATE : operand menggeser ke kiri atau ke kanan dengan ujung yang terjalin.

- Konversi

Tindakan CPU sama dengan arithmetic dan logical.
Instruksi yang mengubah format instruksi yang beroperasi terhadap format data.
Misalnya pengubahan bilangan desimal menjadi bilangan biner.

Operasi set instruksi untuk conversi :

1. TRANSLATE : menterjemahkan nilai-nilai dalam suatu bagian memori berdasrkan tabel korespodensi.
2. CONVERT : mengkonversi isi suatu word dari suatu bentuk ke bentuk lainnya.

- Input / Output
Tindakan CPU untuk melakukan INPUT /OUTPUT :

1. Apabila memory mapped I/O maka menentukan alamat memory mapped.
2. Mengawali perintah ke modul I/O

Operasi set instruksi Input / Ouput :
1. INPUT : memindahkan data dari pernagkat I/O tertentu ke tujuan
2. OUTPUT : memindahkan data dari sumber tertentu ke perangkat I/O
3. START I/O : memindahkan instruksi ke prosesor I/O untuk mengawali operasi I/O
4. TEST I/O : memindahkan informasi dari sistem I/O ke tujuan

- Transfer Control
Tindakan CPU untuk transfer control :
Mengupdate program counter untuk subrutin , call / return.
Operasi set instruksi untuk transfer control :

· JUMP (cabang) : pemindahan tidak bersyarat dan memuat PC dengan alamat tertentu.

· JUMP BERSYARAT : menguji persyaratan tertentu danmemuat PC dengan alamat tertentu atau tidak melakukan apa tergantung dari persyaratan.

· JUMP SUBRUTIN : melompat ke alamat tertentu.
· RETURN : mengganti isi PC dan register lainnya yang berasal dari lokasi tertentu.
· EXECUTE : mengambil operand dari lokasi tertentu dan mengeksekusi sebagai instruksi
· SKIP : menambah PC sehingga melompati instruksi berikutnya.
· SKIP BERSYARAT : melompat atau tidak melakukan apa-apa berdasarkan pada persyaratan

· HALT : menghentikan eksekusi program.

· WAIT (HOLD) : melanjutkan eksekusi pada saat persyaratan dipenuhi.

· NO OPERATION : tidak ada operasi yang dilakukan.

- Control System
Hanya dapat dieksekusi ketika prosesor berada dalam keadaan khusus tertentu atau sedang mengeksekusi suatu program yang berada dalam area khusus, biasanya digunakan dalam sistem operasi.

Contoh : membaca atau mengubah register kontrol.

4. Pengalamatan
Metode pengalamatan adalah bagaimana cara menunjuk dan mengalamati suatu lokasi memori pada sebuah alamat di mana operand akan diambil. Mode pengalamatan diterapkan pada set instruksi, pengalamatan memberikan fleksibilitas khusus yang sangat penting.

JUMLAH ALAMAT (NUMBER OF ADDRESSES)
* Salah satu cara tradisional untuk menggambarkan arsitektur prosessor adalah dengan melihat jumlah alamat yang terkandung dalam setiap instruksinya.
* Jumlah alamat maksimum yang mungkin diperlukan dalam sebuah instruksi :

1. Empat Alamat (dua operand, satu hasil, satu untuk alamat instruksi berikutnya)
2. Tiga Alamat (dua operand, satu hasil)
3. Dua Alamat (satu operand merangkap hasil, satunya lagi operand)
4. Satu Alamat (menggunakan accumulator untuk menyimpan operand dan hasilnya)

Macam-macam instruksi menurut jumlah operasi yang dispesifikasikan
1. O — Address Instruction
2. 1 — Addreess Instruction.
3. N — Address Instruction
4. M + N — Address Instruction

Macam-macam instruksi menurut sifat akses terhadap memori atau register
1. Memori To Register Instruction
2. Memori To Memori Instruction
3. Register To Register Instruction

ADDRESSING MODES

1. Immediate addressing
Operand (data yang akan dikomputasi) berada langsung pada set instruksi.

2. Direct Addressing
Operand berada pada memori, set instruksi memegang alamat lokasi memori dimana operand tersebut berada.

3. Indirect Addresing
Operand berada pada memori, untuk mendapatkan operand ini CPU harus melakukan penelusuran dua kali yaitu dari data alamat memori yang ada pada set instruksi serta alamat yang ditunjuk oleh alamat memori yang diperoleh dari set instruksi tadi.

4. Register addressing
Operand berada pada register, cara kerjanya mirip dengan direct addressing hanya saja CPU mengakses alamat register bukan alamat memori.

5. Register Indirect Addressing
Operand berada pada memori, untuk mendapatkan operand CPU harus mengakses register terlebih dahulu karena informasi lokasi operand berada pada register.

6. Displacement
Operand berada pada memori, cara kerjanya merupakan gabungan dari teknik direct addressing dan register indirect addressing.

7. Stack
Operand berada pada stack, operand secara berkala dimasukan ke stack sehingga ketika operand dibutuhkan maka operand sudah berada pada “top of the stack”.

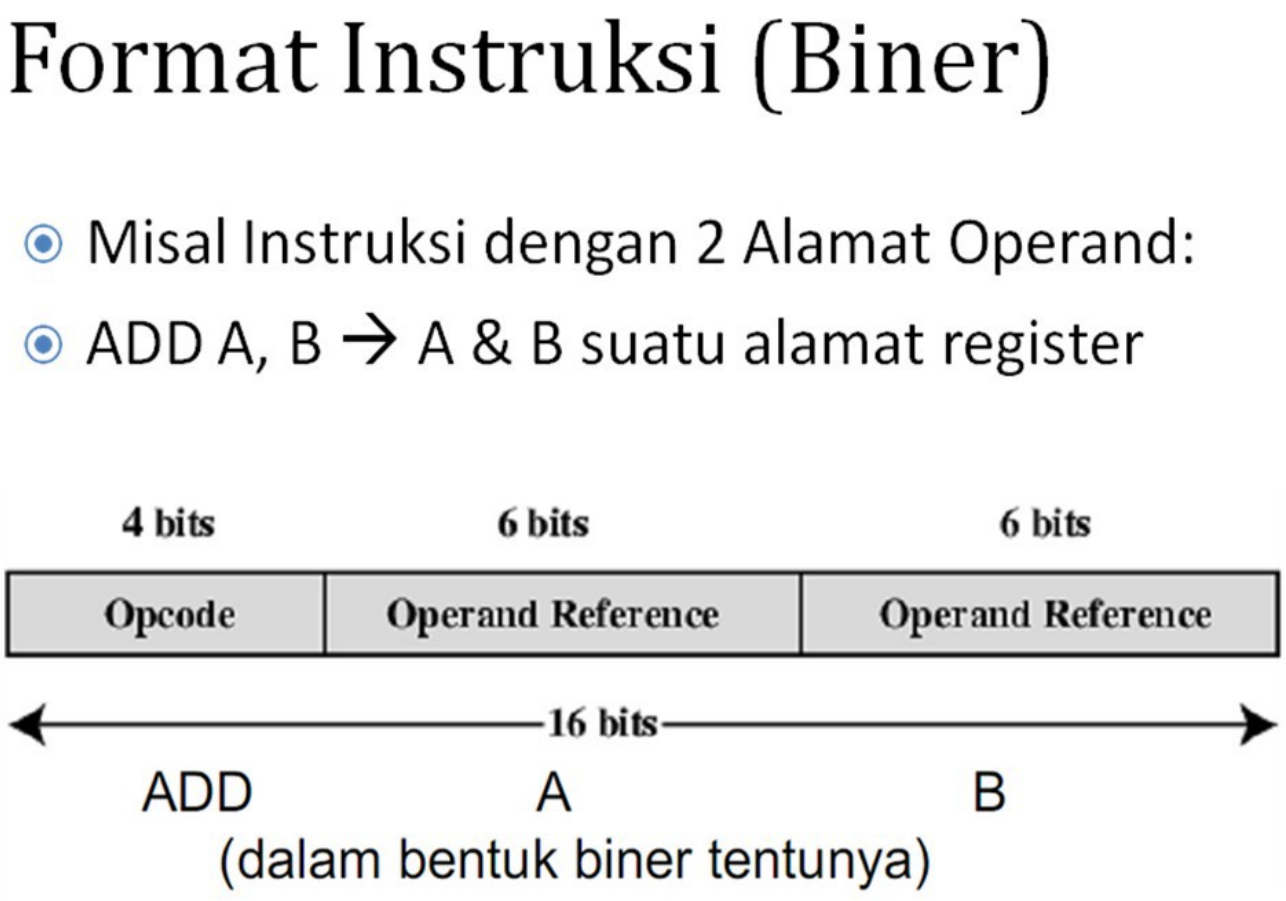
Teknik pengalamatan tersebut harus dapat memenuhi kebutuhan komputasi yang dilakukan oleh computer yang secara garis besar dapat dibagi kedalam tiga kategori yaitu:
- Operasi load (memasukan data).
- Operasi branch (percabangan).
- Operasi aritmatik dan logika.

5. Format Instruksi

Format instruksi menentukan layout bit suatu instruksi. Format instruksi harus mencakup opcode dan secara implisit atau eksplisit, nol operand atau lebih. Secara implisit atau eksplisit, format harus dapat mengindikasikan mode pengalamatan seluruh operand-nya. Pada sebagian besar set instruksi, digunakan lebih dari satu format

instruksi. Rancangan format instruksi merupakan seni yang kompleks, dan telah diimplementasikan bermacam-macam rancangan.

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



Bentuk instruksi:

- Format instruksi 3 alamat
Mempunyai bentuk umum seperti : [OPCODE][AH],[AO1],[AO2]. Terdiri dari satu alamt hasil, dan dua alamat operand, misal SUB Y,A,B Yang mempunyai arti dalam bentuk algoritmik : Y := A — B dan arti dalam bentuk penjelasan : kurangkan isi reg a dengan isi reg B, kemudian simpan hasilnya di reg Y. bentuk bentuk pada format ini tidak umum digunakan di dalam computer, tetapi tidak dimungkinkan ada pengunaanya, dalam peongoprasianya banyak register sekaligus dan program lebih pendek.

Contoh:
A, B, C, D, E, T, Y adalah register
Program: Y = (A — B) / (C + D × E)
SUB Y, A, B Y := A — B
MPY T, D, E T := D × E
ADD T, T, C T := T + C
DIV Y, Y, T Y:= Y / T
Memerlukan 4 operasi

- Format instruksi 2 alamat

Mempunyai bentuk umum : [OPCODE][AH],[AO]. Terdiri dari satu alamat hasil merangkap operand, satu alamat operand, missal : SUB Y,B yang mempunyai arti dalam algoritmik : Y:= Y — B dan arti dalam bentuk penjelasan : kurangkan isi reg Y dengan isi reg B, kemudian simpan hasillnya di reg Y. bentuk bentuk format ini masih digunakan di computer sekarang, untuk mengoprasikan lebih sedikit register, tapi panjang program tidak bertambah terlalu banyak.

Contoh :
A, B, C, D, E, T, Y adalah register
Program: Y = (A — B) / (C + D × E)
MOVE Y, A Y := A
SUB Y, B Y := Y — B
MOVE T, D T := D
MPY T, E T := T × E
ADD T, C T := T + C
DIV Y, T Y:= Y / T
Memerlukan 6 operasi

- Format instruksi 1 alamat

Mempunyai bentuk umum : [OPCODE][AO]. Terdiri dari satu alamat operand, hasil disimpan di accumulator, missal : SUB B yang mempunyai arti dalam algoritmik : AC:= AC — B dan arti dalam bentuk penjelasan : kurangkan isi Acc dengan isi reg B, kemudian simpan hasillnya di reg Acc. bentuk bentuk format ini masih digunakan di computer jaman dahulu, untuk mengoprasikan di perlukan satu register, tapi panjang program semakin bertambah.

Contoh :
A, B, C, D, E, Y adalah register
Program: Y = (A — B) / (C + D × E)
LOAD D AC := D
MPY E AC := AC × E
ADD C AC := AC + C
STOR Y Y := AC
LOAD A AC := A
SUB B AC := AC — B
DIV Y AC := AC / Y
STOR Y Y := AC
Memerlukan 8 operasi

- Format instruksi 0 alamat

Mempunyai bentuk umum : [OPCODE]. Terdiri dari semua alamat operand implicit, disimpan dalam bentuk stack. Operasi yang biasanya membutuhkan 2 operand, akan mengambil isi stack paling atas dan dibawahnya missal : SUB yang mempunyai arti dalam algoritmik : S[top]:=S[top-1]-S[top] dan arti dalam bentuk penjelasan : kurangkan isi stack no2 dari atas dengan isi stack paling atas, kemudian simpan hasilnya di stack paling atas, untuk mengoprasikan ada beberapa instruksi khusus stack PUSH dan POP.

Contoh :
A, B, C, D, E, Y adalah register
Program: Y = (A — B) / (C + D × E)
PUSH A S[top] := A
PUSH B S[top] := B
SUB S[top] := A — B
PUSH C S[top] := C
PUSH D S[top] := D
PUSH E S[top] := E
MPY S[top] := D × E
ADD S[top] := C + S[top]
DIV S[top] := (A — B) /S[top]
POP Y Out := S[top]

6. Kesimpulan

Instruction Set Architecture (ISA) didefinisikan sebagai sesuatu aspek dalam arsitektur computer yang dapat dilihat oleh para pemrogram. Disebut juga machine code (bahasa mesin), aslinya juga berbentuk biner > bahasa assembly.

Di dalam intruksi — intruksi terdiri dari operand dan operator yang nantinyaakan melakukan sebuah operasi di dalam komputer. Operasi — operasi yangberjalan di dalam CPU ditentukan oleh instruksi-instruksi yang dieksekusinya.

Di dalam set instruksi ada elemen-elemen yang akan digunakan untukdieksekusi. Namun intruksi yang dilakukan harus direpresentasikan olehsehimpunan bit agar dapat dimengerti oleh manusia / programmer. Intruksidapat berupa 3 alamat — 0 alamat.