

# ORGANISASI KOMPUTER DASAR

## A. KOMPONEN SISTEM

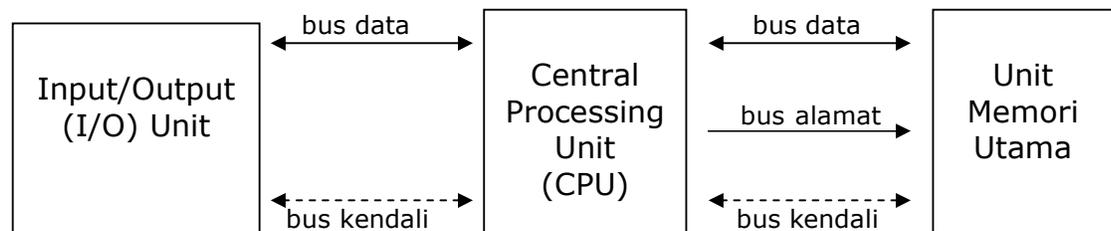
Sebuah komputer moderen/digital dengan program yang tersimpan di dalamnya merupakan sebuah system yang memanipulasi dan memproses informasi menurut kumpulan instruksi yang diberikan. Sistem tersebut dirancang dari modul-modul hardware seperti :

1. Register
2. Elemen aritmatika dan logika
3. Unit pengendali
4. Unit memori
5. Unit masukan/keluaran (I/O)

Komputer dapat dibagi menjadi 3 bagian utama, yaitu :

1. Unit pengolahan pusat (CPU)
2. Unit masukan/keluaran (I/O)
3. Unit memori

Organisasi dasar dari sebuah komputer dapat ditunjukkan pada blok diagram pada gambar di bawah ini :



Keterangan :

CPU mengendalikan urutan dari semua pertukaran informasi dalam komputer dan dengan dunia luar melalui unit I/O. Sedangkan unit memori terdiri dari sejumlah besar lokasi yang menyimpan program dan data yang sedang aktif digunakan CPU. Ketiga unit tersebut dihubungkan dengan berbagai macam bus.

- Bus adalah sekelompok kawat atau sebuah jalur fisik yang berfungsi menghubungkan register-register dengan unit-unit fungsional yang berhubungan dengan tiap-tiap modul. Informasi saling dipertukarkan di antara modul dengan melalui bus.

## B. OPERASI MIKRO

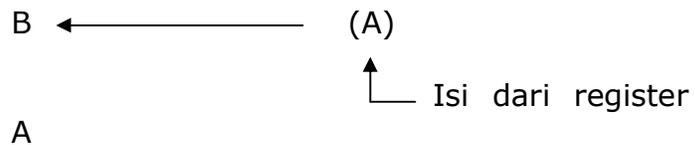
Adalah operasi tingkat rendah yang dapat dilakukan oleh komputer atau CPU sehingga fungsi-fungsi operasi akan dihasilkan untuk memindahkan data antar register.

Salah satu cara dalam melakukan operasi mikro tersebut dengan menggunakan bahasa transfer register / Register Transfer Language (RTL).

RTL adalah sebuah bahasa yang digunakan untuk menjabarkan atau melaksanakan operasi mikro.

Untuk mengungkapkan bahasa RTL ini dapat digunakan notasi RTL yang merupakan aturan penulisan pemberian instruksi RTL. Contoh notasi tersebut antara lain :

- Notasi RTL untuk mentransfer isi register A ke B



- Notasi RTL untuk mentransfer bagian-bagian dari register (field). Sebuah field pada sebuah register dinotasikan dengan menggunakan tanda kurung. Field AD di register IR ditransfer ke register PC



- Notasi RTL untuk mentransfer field AD dari register IR ke register PC



Isi register X ditransfer ke bit 0 sampai 3 pada register R1, yang berarti X mempunyai panjang bit adalah 4. Selain itu, dapat juga dipakai konstanta pada sisi sebelah kanan tanda panah.

$$L \longleftarrow 5$$

Artinya simpan nilai 5 pada register L

- Notasi RTL untuk menggambarkan berbagai macam operasi-mikro Aritmatika.

$$A3 \longleftarrow (A1) + (A2)$$

Artinya isi register A1 dan A2 dijumlahkan dengan menggunakan sirkuit adder biner dan hasil jumlahnya ditransfer ke register A3.

Namun apabila dilakukan pengulangan penjumlahan akan menyebabkan *overflow* dan untuk menampung overflow tersebut digunakan register 1-bit yaitu V sebagai register overflow serta pelengkap A3.

$$VA3 \longleftarrow (A1) + (A2)$$

Contoh operasi aritmatika lainnya :

$$\begin{array}{ll}
 A \longleftarrow (A) + 1 & ; \text{ increment isi A oleh 1} \\
 A \longleftarrow (A) - 1 & ; \text{ decrement isi A oleh 1} \\
 A \longleftarrow (A) & ; \\
 \text{komplemen A} & \\
 A \longleftarrow (A) + B + 1 & ; \text{ lakukan } A - B \text{ dengan} \\
 & \text{menambahkan} \\
 & \text{komplemen 2's ke A}
 \end{array}$$

- Notasi RTL untuk menggambarkan berbagai macam operasi-mikro Logika.

$$C \longleftarrow (A) \text{ OR } (B)$$

Artinya bahwa logika OR dari sisi register A dan B ditransfer ke register C. Begitu juga dengan operasi AND

$C \longleftarrow (A) \text{ AND } (B)$

- Notasi RTL untuk menggambarkan transfer data ke dan dari word memori.  
Dalam RTL, unit memori utama pada komputer dianggap sebagai M dan menulis word ke-i dalam memori menjadi M[i].

Proses pembacaan memori (*memory read*) adalah :

$B \longleftarrow (M[A])$

Proses penulisan memori (*memory write*) adalah :

$(M[A]) \longleftarrow B$

artinya word memori yang alamatnya ditunjukkan oleh register A ditransfer ke atau dari register B dalam CPU.

- Notasi RTL digunakan untuk transfer register hanya pada kondisi tertentu, hal tersebut dilakukan dengan 2 cara :
  1. Menggunakan pernyataan kondisi logika (logical condition)

$\text{IF } (V) > (W) \text{ THEN } Q \longleftarrow 0$

Men-set 0 ke register Q hanya jika nilai register V lebih besar dari nilai register W.

Note :

Pernyataan kondisi logikal hanya didefinisikan untuk IF - THEN dan tidak untuk ELSE.

2. Menggunakan pernyataan kondisi pengendalian (control condition)

$t_0 (c_1 + c_2) : X \longleftarrow (Y)$

dengan metode ini, kondisinya merupakan fungsi logikal dari variabel biner yang mengatur input register.

Fungsi-fungsi ini dijabarkan disebelah kiri dari operasi transfer register dan diikuti oleh tanda titik dua.

Keterangan contoh di atas :

Isi Y dipindahkan ke X hanya jika  $t_0$  bernilai 1 dan salah satu  $c_1$  atau  $c_2$  juga bernilai 1  
 Namun jika kondisi tertentu adalah 0, simbol utama ( $\leftarrow$ ) harus digunakan sehingga pernyataan RTL - nya adalah :

$$t'_0 (c_1 + c_2) : X \leftarrow (Y)$$

maka transfer hanya akan terjadi jika  $t_0$  bernilai 0 dan salah satu  $c_1$  atau  $c_2$  juga bernilai 1.

### **SIC (SIMPLIFIED INSTRUCTIONAL COMPUTER)**

Komputer yang didasarkan pada SIC ini merupakan komputer yang termasuk dalam perancangan arsitektur yang sangat sederhana dan komputer ini dipersembahkan oleh BECK (1985).

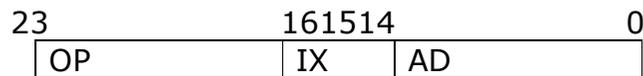
Struktur Mesin SIC terdiri dari :

1. CPU
2. Unit memori
3. Minimal satu unit prinati I/O

Untuk CPU yang digunakan terdiri dari 13 register khusus, seperti yang ada pada table di bawah ini.

NO	REGISTER	UKURAN (bit)	NAMA
1	A	24	Accumulator
2	X	15	Register Index
3	L	15	Register Linkage
4	PC	15	Program Counter
5	IR	24	Instruction Register
6	MBR	24	Memori Buffer Register
7	MAR	15	Memori Address Register
8	SW	11	Status Word
9	C	2	Counter
10	INT	1	Interrupt Flag
11	F	1	Fetch Cycle Flag
12	E	1	Execute Cycle Flag
13	S	1	Start / Stop Flag

Format instruksi pada mesin SIC :



Keterangan :

OP = OPCODE 8 bit yang menerangkan operasi-mikro yang akan dijalankan

IX = flag indeks yang menunjukkan mode pengalamatan yang harus digunakan

AD = alamat untuk memori operand 15 bit

- Pengalamatan langsung (direct addressing) yaitu operand disimpan di dalam M[AD]
- Pengalamatan berindeks (index addressing) yaitu operand disimpan di dalam M[AD = (X)] dengan bit IX bernilai 1

### Penggunaan register-register pada SIC

1. Register A = register yang digunakan untuk proses perhitungan
2. Register X = register yang digunakan untuk mode pengalamatan berindex
3. Register PC = register yang menyimpan alamat instruksi berikutnya
4. Register L = register yang menyimpan alamat asal sebelum melakukan subroutines
5. Register IR = register yang menyimpan instruksi yang sedang dikerjakan
6. Register MBR = register yang digunakan untuk proses masukan atau keluaran data dari memori
7. Register MAR = register yang menyimpan alamat memori untuk proses pembacaan atau penulisan
8. SW = register yang berisi informasi status relatif terhadap instruksi sebelumnya
9. C = register yang membangkitkan signal waktu t<sub>0</sub>, t<sub>1</sub>, t<sub>2</sub>, t<sub>3</sub>
10. INT = register yang menentukan apakah signal interrupt telah diterima
11. F = register yang digunakan dalam proses "siklus fetch"
12. E = register khusus yang digunakan dalam proses "siklus eksekusi"
13. S = register yang akan mengaktifkan register C

## Kumpulan Instruksi SIC

Ada 21 instruksi SIC yang digunakan, dimana pada instruksi ini  $m$  menunjukkan address memori dari operand dan  $(m)$  menunjukkan nilai yang disimpan pada address memori tersebut. Opcode instruksinya ditulis dalam notasi heksadesimal.

- JSUB dan RSUB merupakan dua instruksi yang berhubungan dengan subrutin. JSUB menyimpan PC saat ini ke L dan kemudian melompat ke subrutin dengan menyimpan operand ke PC. RSUB kembali dari subrutin dengan melompat ke lokasi yang dinyatakan oleh L.
- Instruksi TD digunakan untuk menguji piranti I/O sebelum berusaha untuk membaca dari atau menulis ke piranti tersebut. Hasil pengujian tersebut disimpan di dalam kode kondisi (*condition code*), field CC, pada SW. Panjang field ini 2 bit dan digunakan untuk mewakili salah satu dari tiga nilai  $<$ ,  $=$ ,  $>$   
Jika instruksi TD dijalankan, nilai field CC aka di-set menurut kode berikut :

$<$  menunjukkan bahwa piranti telah siap

$=$  menunjukkan bahwa piranti sedang sibuk dan tidak dapat digunakan pada saat itu

$>$  menunjukkan bahwa piranti tidak beroperasi

- Instruksi COMP digunakan juga untuk men-set field CC. Nilai yang disimpan field CC setelah sebuah instruksi COMP setelah sebuah instruksi COMP menggambarkan hubungan antara A dan operand instruksi
- Instruksi IRT digunakan oleh interrupt handler agar menyebabkan lompatan kembali ke tempat dimana CPU berada sebelum intrupsi terjadi.  
Jika interupsi terjadi, CPU akan menyimpan PC saat ini ke dalam memori pada address 0.  
Untuk kembali dari sebuah interupsi , isi dari alamat memori ini harus di-load kembali ke dalam PC.
- Instruksi-instruksi lainnya adalah operasi aritmatika dan logika, transfer dari pengendalian(jump), loading register, storing register atau membaca dan menulis ke piranti I/O.