

# BAB 2

---

## ARSITEKTUR MIKROPROSESOR

Pemahaman yang baik terhadap arsitektur mikroprosesor sangat membantu kemampuan pengembangan program sistem mikroprosesor. Arsitektur mikroprosesor berkaitan dengan rancangan software dan hardware internal sebuah mikroprosesor. Ada tiga jenis software arsitektur mikroprosesor yaitu: (1) *Complex Instruction Set Computer (CISC)*, (2) *Reduce Instruction Set Computer (RISC)*, dan (3) Mikroprosesor Superskalar. Dan ada tiga jenis hardware arsitektur mikroprosesor yaitu: (1) Arsitektur I/O terisolasi, (2) Arsitektur I/O terpetakan dalam Memori, dan (3) Arsitektur Harvard.

Kata Kunci: arsitektur, set instruksi, *software*, *hardware*

### 1. Jenis Arsitektur Mikroprosesor

Arsitektur mikroprosesor biasanya berkaitan dengan bangunan, rancangan atau desain sebuah mikroprosesor. Desain sebuah mikroprosesor dengan ciri-ciri pokok yang sering disebut dengan *features* sebuah mikroprosesor dapat dipelajari dengan baik melalui ***Internal Software-Hardware Design***. Pemahaman dan pengkajian mendalam terhadap rancangan software dan hardware yang disebut juga dengan istilah arsitektur akan sangat membantu dalam pemrograman mikroprosesor.

Arsitektur sebuah mikroprosesor menunjukkan rancangan tentang perangkat lunak dan perangkat keras yang terpadu menjadi satu. Rancangan perangkat lunak dan perangkat keras sebuah mikroprosesor dikembangkan secara simultan sebelum

sebuah mikroprosesor diproduksi. Arsitektur perangkat lunak mikroprosesor disebut juga dengan set instruksi. Setiap mikroprosesor memiliki set instruksi tersendiri yang terdiri dari sejumlah instruksi yang dapat bekerja di dalam perangkat keras mikroprosesor.

*Internal software design* berkaitan dengan bentuk atau rancangan set instruksi (*instruction set*) yang digunakan. Set instruksi sebuah mikroprosesor dibangun dan dikembangkan bersamaan dengan pengembangan rancangan perangkat keras mikroprosesornya. Setiap perintah dalam set instruksi harus bekerja pada saat proses *decoding* yang dilakukan oleh perangkat keras mikroprosesor. Disebut *internal software* karena set instruksi berkaitan langsung dengan perangkat keras yang ada di dalam mikroprosesor. Setiap perintah dalam set instruksi dikodekan dalam heksa desimal.

## 1.1. Internal Software Design

Ada tiga model arsitektur mikroprosesor dilihat dari perangkat lunak dalam bentuk set instruksi sebagai *software design*. Ketiga jenis arsitektur mikroprosesor yaitu: (1) *Complex Instruction Set Computer* (CISC), (2) *Reduce Instruction Set Computer* (RISC), dan (3) Mikroprosesor Superskalar.

### 1.1.1. Complex Instruction Set Computer (CISC)

Pada mulanya dalam industri komputer, pemrograman dilakukan menggunakan bahasa *assembly* atau kode-kode bahasa mesin. Pemrograman semacam ini sangat *powerful* dan mudah menggunakan instruksi. Perancang CPU mencoba membuat instruksi yang dapat melakukan berbagai perintah kerja. CISC adalah jenis arsitektur mikroprosesor yang menggunakan banyak jenis dan ragam instruksi. CISC menyediakan kemampuan setiap instruksi dapat mengeksekusi operasi *low-level*, seperti *men-load* data dari memori, operasi aritmetika, dan melakukan prosedur penyimpanan ke memori. Mikroprosesor jenis ini memiliki kemampuan eksekusi cepat. Contoh mikroprosesor dengan arsitektur CISC adalah Intel 8088, 8085, 8086, Zilog Z-80 CPU, NS 32016, MC6800. Karena jumlah instruksi lebih banyak jenis dan ragamnya maka kelemahan CISC terletak pada sulitnya mengembangkan interpreter dan kompiler.

### 1.1.2. Reduce Instruction Set Computer (RISC)

RISC merupakan arsitektur *instruction set* yang menekankan kepada kesederhanaan instruksi “bekerja sedikit” tetapi tetap memberikan hasil performansi yang tinggi. Hal ini bisa terjadi karena proses eksekusi instruksinya sangat cepat. Arsitektur ini lebih baru dibandingkan dengan arsitektur CISC. Arsitektur RISC memiliki sedikit instruksi banyak register. Contoh mikroprosesor dengan arsitektur RISC adalah AMD 2900, MIPS R2000, SUN SPARC, MC 8800, ATMET 90S1200, 90S2313, 90S2323, 90S2343, 90S4434, 90S8515. Ciri-ciri RISC :

- Instruksi bersifat tunggal
- Ukuran instruksi umumnya 4 byte
- Jumlah mode pengalamatan (Addressing mode) lebih sedikit dibawah lima,
- Tidak ada mode pengalamatan tidak langsung (*indirect addressing mode*),
- Tidak ada operasi yang menggabungkan operasi *Load/Store* dengan operasi aritmetika,
- Setiap instruksi dalam satu lokasi memori memiliki lebih dari satu operand.
- Tidak mendukung sembarang peralatan
- Satu instruksi satu alamat data,
- Minimal 32 register interger dapat dirujuk secara eksplisit,
- Minimal 16 register *floating point* direferensikan secara eksplisit.

### 1.1.3. Mikroprosesor Superskalar

Mikroprosesor dengan arsitektur superskalar adalah mikroprosesor yang menggunakan instruksi-instruksi biasa (aritmetika, *floating point*, *store*, *branch*) tetapi bisa diinisialisasi secara simultan dan dapat dieksekusi secara independen. Contoh mikroprosesor dengan arsitektur superskalar antara lain: IBM RS 6000, Pentium (CISC dengan konsep superskalar).

## 1.2. Internal Hardware Design

*Internal hardware design* berkaitan dengan masalah-masalah jenis, jumlah, dan ukuran register serta komponen lainnya. Untuk dapat menginstalasikan sebuah mikroprosesor dengan komponen lainnya seperti RWM, ROM, dan I/O sebagai komponen utama dan rangkaian Clock, Reset, Buffer, dan lain-lain sebagai komponen pendukung diperlukan pemahaman sistem bus yang dimiliki oleh setiap mikroprosesor.

Ada tiga jenis arsitektur mikroprosesor berdasarkan *internal hardware design* yaitu:

- ❑ Arsitektur I/O terisolasi
- ❑ Arsitektur I/O terpetakan dalam memori
- ❑ Arsitektur Harvard

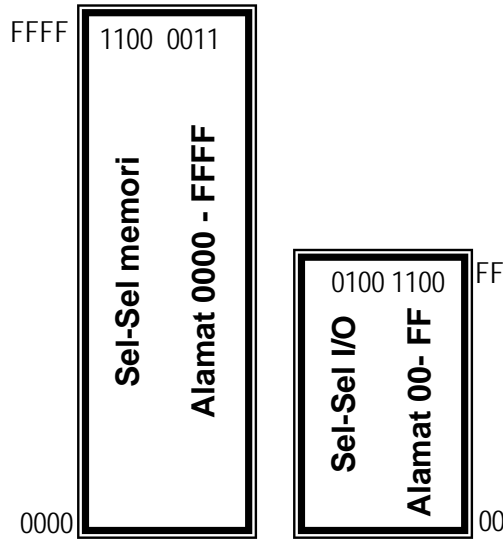
### 1.2.1. Arsitektur I/O Terisolasi

Mikroprosesor dengan arsitektur I/O terisolasi menggunakan disain pengalamatan atau pemetaan I/O terpisah atau terisolasi dengan pengalamatan atau pemetaan memori. Pengalamatan I/O menggunakan sebagian dari jumlah saluran alamat (*address bus*) sedangkan pengalamatan memori menggunakan semua saluran alamat (*address bus*). Ini merupakan ciri pokok dari mikroprosesor dengan arsitektur I/O terisolasi. Ada pengendalian yang terpisah dan bergantian. Pada saat mikroprosesor mengakses memori maka I/O harus off. Sebaliknya pada saat mikroprosesor mengakses I/O memori harus off.

Untuk memudahkan memahami kita gunakan kasus sebuah mikroprosesor dengan arsitektur I/O terisolasi memiliki saluran alamat 16 bit. Jumlah lokasi memori maksimum yang dapat dialamati oleh mikroprosesor ini adalah  $2^{16}$  atau 64 Kilo byte dan jumlah lokasi I/O yang dapat dialamati adalah  $2^8$  yaitu sama dengan 256 byte. Jadi pengalamatan memori menggunakan seluruh saluran alamat dalam hal ini 16 bit sedangkan pengalaman I/O menggunakan sebagian saluran alamat dalam hal ini 8 bit.

Jenis arsitektur I/O terisolasi menyediakan akses memori dan I/O secara terpisah. Artinya pada saat mengakses

memori, perangkat I/O harus off. Sebaliknya pada saat mengakses I/O bagian memori harus off. Model arsitektur I/O terisolasi dapat digambarkan seperti Gambar 2.1.



Gambar 2.1. Model Arsitektur I/O Terisolasi

Dari Gambar 2.1. terlihat jelas peta sel-sel memori terpisah atau terisolasi dengan peta sel-sel I/O. Untuk mikroprosesor dengan bus alamat 16 bit yakni dari A0 sampai dengan A15 sel memori berada pada alamat 0000H sampai dengan FFFFH. Sedangkan sel I/O berada pada alamat terpisah diantara 00H sampai dengan FFH.

Metoda I/O terisolasi menggunakan akumulator pada CPU untuk menerima data dari I/O atau mengeluarkan data ke bus I/O selama operasi input output. Tidak ada register lain selain akumulator yang terpakai untuk akses I/O. Dengan demikian arsitektur

I/O terisolasi disebut juga dengan I/O akumulator.

Konsep arsitektur I/O terisolasi memiliki pengaruh penting pada program komputer yaitu :

- Instruksi yang digunakan untuk mengakses I/O hanya dua kode operasi yaitu IN dan OUT.
- Informasi/data yang ada pada akumulator harus dialihkan pada suatu lokasi penyimpanan sementara sebelum ada operasi I/O berikutnya.
- Perlu ada tambahan instruksi pada program pengalihan data/informasi pada akumulator.

#### Keuntungan metoda I/O terisolasi :

- Komputer dapat mengalihkan informasi/data ke atau dari CPU tanpa menggunakan memori. Alamat atau lokasi memori sepenuhnya digunakan untuk operasi memori bukan untuk operasi I/O.
- Lokasi memori tidak berkurang oleh sel-sel I/O
- Instruksi I/O lebih pendek sehingga dapat dengan mudah dibedakan dari instruksi memori.
- Pengalamatan I/O menjadi lebih pendek dan perangkat keras untuk pengkodean alamat lebih sederhana.

Sedangkan kerugian arsitektur I/O terisolasi lebih banyak menggunakan saluran pin pengendalian pada bus kendali dari mikroprosesornya.

Mikroprosesor buatan perusahaan Intel dan mikroprosesor buatan Zilog menggunakan arsitektur I/O terisolasi.

### 1.2.2. Arsitektur I/O Terpetakan dalam Memori

Mikroprosesor dengan arsitektur I/O terpetakan dalam memori menyatukan sel-sel I/O dalam pengalamatan bersama dengan sel-sel memori. Mikroprosesor dengan arsitektur I/O terpetakan dalam memori dapat diilustrasi seperti Gambar 2.2.



Gambar 2.2. Model Arsitektur I/O Terpetakan dalam Memori

Dari Gambar 2.2. nampak bahwa sel-sel I/O menjadi satu dengan sel-sel memori. Arsitektur I/O terpetakan dalam memori menunjukkan penggunaan instruksi tipe memori untuk mengakses alat-alat I/O. I/O yang dipetakan dalam memori memungkinkan CPU menggunakan instruksi yang sama untuk alih data ke memori seperti yang digunakan untuk alih data ke I/O.

Sebuah pintu I/O diperlakukan seperti sebuah lokasi memori. Keuntungan sistem ini adalah instruksi yang dipakai untuk pembacaan dan penulisan memori dapat digunakan untuk memasukkan dan mengeluarkan data pada I/O.

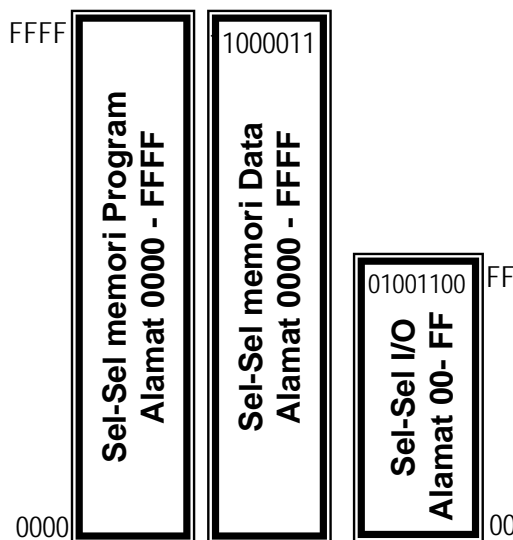
Kerugiannya pertama tiap satu pintu I/O mengurangi satu lokasi memori yang tersedia. Kedua alamat lokasi I/O memerlukan 16 bit saluran. Ketiga instruksi I/O yang dipetakan dalam memori lebih lama dari instruksi I/O terisolasi. Gambar 2.2 menunjukkan bentuk pengendalian I/O terpetakan dalam Memori.

### 1.2.3. Arsitektur Harvard

Arsitektur Harvard menggunakan disain yang hampir sama dengan arsitektur I/O terisolasi. Perbedaannya pada arsitektur Harvard antara memori program dan memori data dipisahkan atau diisolasi. Pemisahan antara memori program dan memori data

menggunakan perintah akses memori yang berbeda.

Arsitektur Harvard ditinjau dari kemampuan jumlah memori lebih menguntungkan. Terpisahnya memori program dengan memori data menyebabkan arsitektur Harvard berkemampuan memori dua kali lipat kemampuan memori arsitektur I/O terisolasi. Model pemetaan arsitektur Harvard digambarkan pada Gambar 2.3.



Gambar 2.3. Model Arsitektur Harvard

Selanjutnya untuk mempelajari lebih mendalam arsitektur sebuah mikroprosesor secara nyata digunakan mikroprosesor Z-80 CPU sebagai studi kasus. Untuk kasus-kasus mikroprosesor jenis lainnya dapat dengan mudah dipahami karena merupakan replika yang struktur konsepnya hampir sama.

## 2. Feature Mikroprosesor Kasus pada Zilog Z-80 CPU

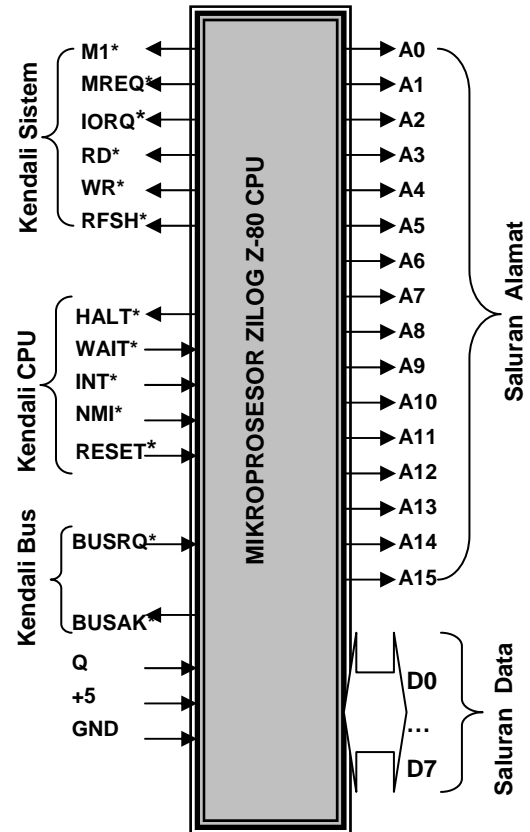
Mikroprosesor Zilog Z-80 CPU adalah mikroprosesor 8 bit buatan Zilog dengan gambaran kemampuan sebagai berikut::

- Mikroprosesor 8 bit dengan arsitektur I/O terisolasi.
- Bus alamat 16 bit dan bus data 8 bit dengan kemampuan:
  - ✓ Pengalamatan memori 64 Kilo byte.
  - ✓ Pengalamatan I/O 256 byte.
- CISC dengan 148 instruksi.
- 8 buah register 8 bit sebagai register utama.
- 8 buah register 8 bit sebagai register alternatif.
- 4 buah register 16 bit.
- 2 buah register 8 bit fungsi khusus.
- Frekuensi Clock 2,5 MHz - 4 Mhz
- Komsumsi daya: Aktif 150 mA
- Kemasan *plastic dual inline package* (PDIP)

Mikroprosesor Z-80 CPU merupakan mikroprosesor 8 bit yang sangat andal dan populer digunakan pada zamannya. Komputer Apple II dan IBM pertamakali menggunakan mikroprosesor Z-80 CPU. Dengan 40 pin mikroprosesor Z-80 CPU susunan kaki dan fungsi masing-masing pin dapat digambarkan seperti Gambar 2.4. Pada Gambar 2.4. terlihat ada lima kelompok pin saluran yaitu: kelompok kendali sistem, kendali CPU, kendali bus, saluran data, dan saluran alamat.

Kendali CPU menjalankan fungsi-fungsi sebagai berikut :

- **M1\*** (*Machine Cycle One*= satu siklus mesin): merupakan pin keluaran aktif rendah jika CPU sedang mengambil sandi operasi instruksi dari memori. Pada saat ini bus alamat berisi alamat memori seperti data yang ada pada register PC, dan data bus mengarah masuk.
- **MREQ\*** (*Memory Request* = permintaan memori): merupakan pin keluaran aktif rendah pada waktu saluran alamat A0 s/d A15 berisi alamat memori.
- **IORQ\*** (*Input Output Request*= permintaan Input Output): pin keluaran aktif rendah pada waktu saluran alamat A0 s/d A7 berisi alamat I/O.
- **RD\*** (*Read*= Baca): pin keluaran aktif rendah pada waktu CPU melakukan operasi baca/memasukkan data.
- **WR\*** (*Write* = Tulis) Keluaran aktif rendah pada waktu CPU melakukan operasi tulis/mengeluarkan data.
- **RFSH\*** (*Refresh* = penyegaran): pin keluaran aktif rendah jika CPU mengeluarkan alamat memori untuk menyegarkan memori dinamik.



Gambar 2.4 Susunan dan Konfigurasi Pin Z-80 CPU

- **HALT\***: pin keluaran aktif rendah pada saat CPU melaksanakan instruksi halt/berhenti.
- **WAIT\***: adalah pin masukan dibuat aktif rendah oleh alat luar yang menyela kerja CPU.
- **INT\*** (*Interrupt* = interupsi): pin masukan aktif rendah jika ada luar yang meminta layanan interupsi.

- **NMI\*** (*Non Mascable Interrupt/* interupsi yang tidak bisa dihalangi): masukan aktif rendah jika ada selaan yang yang tak dapat dihalangi.
- **RESET\***: masukan dibuat aktif rendah oleh alat luar untuk membuat CPU ada dalam keadaan awal.
- **BUSRQ\*** (*Buss Request =* permintaan bus): sinyal masukan yang dibuat aktif rendah jika ada alat luar yang meminjam bus sistem.
- **BUSAK\*** (*Bus Acknowledge*): keluaran aktif rendah yang menandakan CPU mengijinkan peminjaman bus sistem.

*Catatan : tanda \* berarti aktif Low= 0*

Z-80 CPU dalam mengendalikan sistem menggunakan enam pin kendali dan empat diantaranya digunakan untuk berkomunikasi dengan memori dan I/O. Cara berkomunikasi menggunakan status bit seperti Tabel 2.1. berikut :

Tabel 2.1. Operasi Komunikasi Baca dan Tulis Memori atau I/O

Pin Kendali				Operasi
MREQ*	IORQ*	RD*	WR*	
0	1	0	1	Baca Data dari Memori
0	1	1	0	Tulis Data ke Memori
1	0	0	1	Baca Data dari I/O
1	0	1	0	Tulis Data ke I/O

Operasi komunikasi memori dan I/O menunjukkan data dapat mengalir keluar atau masuk CPU. Pada operasi pembacaan (*READ*) data mengalir masuk ke CPU melalui delapan bit saluran data bus. Pada operasi penulisan (*WRITE*) data mengalir keluar dari CPU. Dalam hal ini satu siklus proses hanya ada satu operasi yaitu operasi baca atau operasi tulis. Cukup nalar bahwa tidak akan pernah terjadi operasi tulis dan operasi baca berlangsung bersamaan. Namun demikian karena siklus clock kerja CPU sangat tinggi maka kedua operasi ini akan dapat berlangsung secara cepat dan seakan-akan terjadi bersamaan.

Ada dua sasaran terpisah dalam pengalihan data yaitu memori dan I/O. Pengendalian ini dilakukan oleh dua bit kontrol yaitu IORQ\* dan MREQ\*. Mode operasi akses memori dan I/O digambarkan seperti tabel 2.2.

Tabel 2.2. Operasi komunikasi Memori atau I/O

Pin Kendali		Operasi
IORQ*	MREQ*	
0	1	Akses I/O
1	0	Akses Memori
1	1	Stand by



### 3. Rancangan *Hardware* Internal Mikroprosesor Z-80 CPU

Arsitektur mikroprosesor Z-80 CPU dapat digambarkan seperti Gambar 2.5. Dari Gambar 2.5. terlihat ada lima blok komponen utama yaitu: ALU, Instruction register, Instruction decoder CPU control & timing, Data bus control, CPU register, Address bus control.

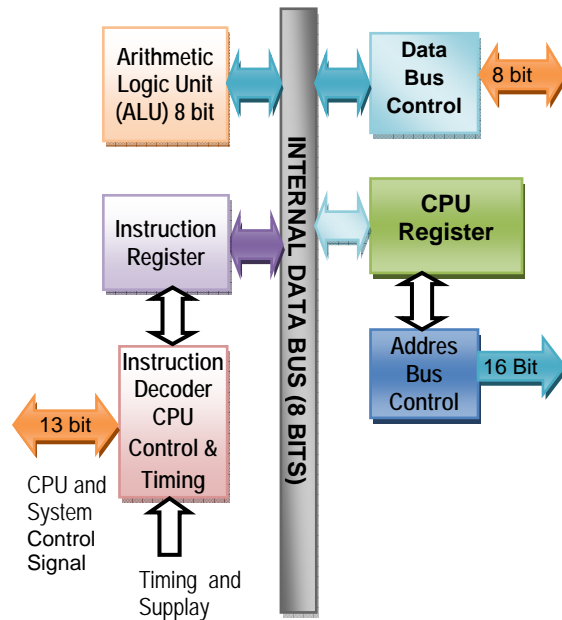
#### 3.1. Arithmetic Logic Unit (ALU)

ALU 8 bit adalah untai gerbang-gerbang logika yang membentuk suatu fungsi esensial yaitu operasi aritmetika (ADD, SUB, dan turunannya) dan operasi Logika (AND, OR, XOR, INC, DEC dan turunannya) dalam kapasitas 8 bit. Dapat membentuk operasi aritmetika 16 bit penjumlahan dan pengurangan dengan cara operand ditempatkan pada dua buah register 8 bit (Register HL, IX, dan IY).

#### □ Fungsi Aritmetika pada ALU

- Penjumlahan (ADD = add, ADC=Add With Carry)
- Pengurangan (SUB=subtract, SBC= Subtract With Carry)
- Penambahan dengan satu (+1) (INC= Increment)
- Pengurangan dengan satu (-1) (DEC= Decrement)
- Perbandingan (CP= Compare)

- Koreksi aritmetika desimal (DAA= Decimal Adjust Accumulator)



Gambar 2.5. Blok diagram Arsitektur Z-80 CPU

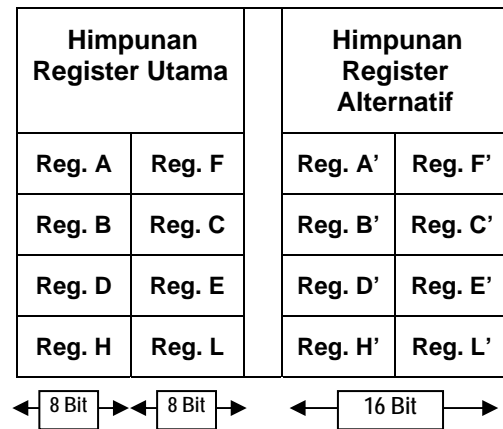
#### □ Fungsi Logika pada ALU

- Fungsi AND
- Fungsi OR
- Fungsi XOR (Exclusive OR)
- Putar Kanan (RRA= Rotate Right Accumulator, RRCA= Rotate Right Circular Accumulator)
- Putar Kiri (RLA= Rotate Left Accumulator, RRCA= Rotate Left Circular Accumulator)
- Geser Kiri (SLA= Shift Left Arithmetic)
- Geser Kanan (SRA= Shift Right Arithmetic)
- Manipulasi bit (SET, RESET, dan Test)

- *Address Bus Control* bekerja mengendalikan pengalamatan memori dan I/O. Pengalamatan memori dan I/O menggunakan satu register 16 bit yang disebut dengan register *Program Counter* (PC).
- *Data Bus Control* bekerja mengatur arah aliran data pada saat operasi pembacaan dan penulisan. *Data bus control* bekerja menggunakan *tri state buffer*.
- Bagian *Instruction Register* dan *Instruction Decoder* bekerja sebagai penerima kode bahasa mesin atau *object code* pada proses *fetching*, dan mendekode *object code* pada saat *decoding* kode-kode yang diambil oleh CPU, serta melakukan eksekusi .

### ➤ Register Internal Z-80 CPU

- ✦ Terdiri dari 18 buah register 8 bit dan 4 buah register 16 bit
- ✦ 16 buah dari 18 buah register 8 bit di bagi menjadi dua himpunan yaitu himpunan register utama dan himpunan register alternatif (pengganti) seperti Gambar 2.6.
- ✦ Dua buah register 8 bit khusus yaitu register I dan Register R.



Gambar 2.6 Susunan Register 8 bit Z-80 CPU

- ✦ Register A disebut juga dengan *Accumulator* yaitu register penampung hasil operasi ALU.
- ✦ Register F (*Flag*) disebut sebagai register status yang berfungsi untuk mencatat status hasil sebuah operasi dalam ALU
- ✦ Register B, C, D, E, H, dan L adalah register serbaguna 8 bit yang dapat dipasangkan menjadi register 16 bit dengan pasangan : BC, DE, dan HL yang dapat digunakan secara mandiri.
- ✦ Register A', F', B', C', D', E', H', L' digunakan sebagai alternatif penyimpanan sementara pada saat mengamankan isi register utama.

### ➤ Register Utama 8 bit

- ✦ Register utama adalah register 8 bit.
- ✦ Sebagai tempat simpan data 1 byte.
- ✦ Isinya dapat dikutipkan dari satu register ke register lainnya.
- ✦ Dapat dioperasikan aritmetik atau logik terhadap data pada akumulator.

#### ✦ Contoh :

1. LD B, 1Fh : Register B diisi dengan data 1Fh
2. LD C,B : Isi Register B dikutipkan ke register C; C = 1Fh
3. LD A, 01h : Akumulator diisi 01h
4. ADD A,B : Isi Reg. B ditambahkan ke A ; A = 20h
5. AND 0Fh : Data di A = 20 di AND kan dengan 0Fh; A = 00h
6. INC B : Isi B ditambahkan 1 ; B = 20h
7. DEC C : Isi C dikurangi 1 ; C = 1Eh
8. LD L, C : Isi Reg. C dikutipkan ke L ; L = 1Eh

### ➤ Register Utama 16 Bit

- ✦ Dapat dibangun menjadi register 16 bit dengan menggabungkan dua buah register 8 bit pasangan BC, DE, HL.
- ✦ Tempat simpan 2 byte data
- ✦ Sebagai pencatat alamat memori
- ✦ Register HL, mempunyai sifat utama sebagai akumulator

- ✦ Register BC, DE sebagai penyimpan penyimpan angka untuk cacahan/ hitungan
- ✦ Dapat dioperasikan aritmetik terhadap data pada akumulator HL.

#### ✦ Contoh:

1. LD DE, 1900h : Register DE diisi dengan data 1900h
2. LD A, 1Fh : Register A diisi data 1Fh ; A = 1Fh
3. LD (DE), A : Data Reg. A dicopy ke alamat 1900 ; (1900) = 1Fh
4. LD HL,1900h : Register HL diisi data 1900h
5. LD B, (HL) : Copy data dari memori yang alamatnya dicatat oleh HL= 1900h ke Reg. B; B= 1Fh
6. LD A,(DE) : Copy data dari memori yang alamatnya dicatat oleh DE= 1900h ke Reg. A; A= 1Fh
7. ADD A, B : Data di A= 1Fh ditambahkan dengan data di B=1Fh; A= 3Eh
8. LD (HL), A : Isi A dicopykan ke alamat HL= 1900h; Alamat 1900 berisi data 3Eh
9. ADD HL,DE : Operasi penjumlahan 16 bit; HL= 3200h

### ➤ Register 16 Bit Khusus

- ✦ Z-80 CPU mempunyai 4 buah register 16 bit yaitu Program Counter (PC), Stack Pointer (SP), Index Register X, Index Register Y.

<b>PROGRAM COUNTER (PC)</b>
<b>STACK POINTER (SP)</b>
<b>INDEX REGISTER X (IX)</b>
<b>INDEX REGISTER Y (IY)</b>

Gambar 2.7. Susunan Register 16 bit mikroprosesor Z-80 CPU

### ➤ PROGRAM COUNTER (PC)

- ✦ Program counter adalah register 16 bit yang sering juga disebut dengan *Instruction Pointer*.
- ✦ *Instruction Pointer* atau PC adalah penunjuk instruksi dalam hal ini pemegang alamat memori lokasi instruksi yang akan dieksekusi oleh CPU.
- ✦ PC secara logika sebagai penunjuk bit dari bus alamat.
- ✦ Misalnya PC = 1800h = 0001 1000 0000 0000b berarti kondisi biner masing-masing saluran dari bus alamat adalah seperti Gambar 2.8.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Gambar 2.8. Kondisi biner saluran alamat

- ✦ Sebelum sebuah program dieksekusi PC harus mencatat alamat awal program.
- ✦ Setelah dieksekusi PC secara otomatis naik satu bit (*Increment*) sampai dinyatakan berhenti atau berubah nilainya sesuai dengan sasaran percabangan Jump dan Branch.
- ✦ PC juga sebagai pemegang alamat I/O menggunakan separo jumlah bit address bus yaitu dari A0 – A7.

### ➤ STACK POINTER (SP).

- ✦ Stack pointer adalah register 16 bit sebagai pencatat atau penunjuk/pointer alamat stack atau penggalan memori pada daerah RWM.
- ✦ Berhubungan dengan perintah atau operasi pembentukan stack dari perintah PUSH dan POP.
- ✦ Isi SP berubah secara otomatis pada setiap operasi PUSH dan POP.
- ✦ Operasi PUSH dan POP adalah operasi transfer data khusus antara register dengan memori khusus jenis RWM.
- ✦ Proses operasi PUSH dan POP menggunakan pola LIFO (*Last In First Out*) atau FILO (*First In Last Out*). Data yang masuk terakhir keluar pertama kali.
- ✦ Perintah PUSH bekerja menumpuk data dari suatu register ke memori.
- ✦ Perintah POP bekerja mengambil data dari memori ke register.

- ✦ Penggunaan PUSH – POP dibatasi oleh luasan memori yang ada pada sistem mikroprosesor.

✦ **Contoh:**

1. LD SP, 1B00h : Register SP mencatat alamat 1B00h
2. LD BC, BBCCh : Register B = BBh dan C = CCh
3. LD DE, DDEEh : Register D = DDh dan E = EEh
4. PUSH DE : Simpan data DDEEh ke stack; Alamat 1AFF= DDh ; alamat 1AFE= EEh ; SP= 1AFE
5. PUSH BC : Simpan data BBCCh ke stack; Alamat 1AFD= BBh ; alamat 1AFC= CCh; SP = 1AFC
6. POP IX : Isi stack dimasukkan ke register IX; Register IX= BBCCh; SP= 1AFE
7. POP IY : Isi stack dimasukkan ke register IY  
Register IY = DDEEh  
SP = 1B00H

➤ **REGISTER INDEKS (IX dan IY)**

- ✦ Register IX dan IY adalah register 16 bit yang independen satu sama lain.
- ✦ Digunakan untuk menyediakan alamat awal 16 bit pada pengalamatan berindeks.
- ✦ Memiliki kemampuan untuk menunjuk alamat memori menggunakan angka

indeks berdasarkan alamat awal yang tercatat pada register IX atau IY.

- ✦ Keuntungan riil dari register ini adalah memperpendek waktu eksekusi dan lebih menyingkat program.

✦ **Contoh :**

1. LD IX, 1900h : Register IX diisi data 2 byte 1900h
2. LD IY, 2000h : Register IY diisi data 2 byte 2000h
3. LD (IX+00), 19h : Memori alamat 1900 diisi data 19h
4. LD (IY+00), 20h : Memori alamat 2000 diisi data 20h
5. LD (IX+05), 19h : Memori alamat 1905 diisi data 19h
6. LD (IY+05), 20h : Memori alamat 2005 diisi data 20h.

➤ **REGISTER R (Refresh)**

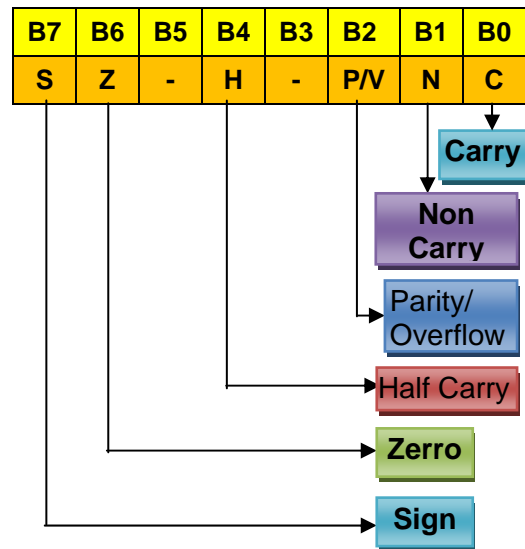
- ✦ Digunakan untuk menyediakan 7 bit (A0 – A6) alamat lokasi memori yang akan disegarkan.
- ✦ Berfungsi untuk memelihara kesegaran data pada memori dinamik jenis RWM.
- ✦ Secara otomatis setiap 2 mili detik disegarkan.
- ✦ Tidak termanfaatkan jika menggunakan memori static. Memori dinamik adalah kapasitor dimana isi data akan bertahan dalam batas waktu tertentu sehingga perlu disegarkan kembali. Memori static adalah sebuah Flip Flop dimana data tetap mantap dan tidak perlu penyegaran.

### ➤ REGISTER INTERUPSI ( I )

- Register 8 bit yang menyediakan byte alamat orde tertinggi bila CPU memasuki subroutin interupsi.
- Alamat interupsi orde rendah diberikan oleh program melalui perangkat interupsi.

### ➤ REGISTER FLAG ( F )

- Register F adalah register 8 bit pencatat status operasi ALU yang sangat penting dalam setiap operasi hitung dan logika sebuah mikroprosesor.
- Status akhir dari sebuah step proses program atau instruksi sangat dibutuhkan dalam membangun keputusan.
- Keputusan untuk mencabang atau melompat dapat dikontrol menggunakan status yang tercatat di Register F.
- Bila ALU telah menyelesaikan operasi aritmetika atau logika, hasilnya akan disimpan di register A, dan bersamaan dengan itu status operasi akan dicatat kondisinya bit demi bit di register F.
- Ada tujuh jenis status pada Mikroprosesor Z-80 CPU antara lain: Carry, Non carry, Parity/Overflow, Half carry, Zerro,dan Sign. Susunan dan makna masing status flag digambarkan seperti Gambar 2.9.
- Gambar 2.9. menunjukkan susunan bit dari register F mulai dari bit B0, B1, B2, B3, B4, B5, B6, B7.



Gambar 2.9. Susunan Register Flag mikroprosesor Z-80 CPU

Berdasarkan Gambar 2.9. terlihat fungsi masing-masing bit seperti Gambar 2.10.

Bit	Logika	Makna
0 (Carry)	0	Operasi Aritmetika/Logika tidak ada Carry atau Borrow
	1	Operasi Aritmetika/Logika ada/terjadi Carry atau Borrow
1 (Negate)	0	Operasi yang terjadi bukan Subtract
	1	Operasi yang terjadi adalah Subtract
2 (V/P)	0	Paritas ganjil atau tidak terjadi OVERFLOW
	1	Paritas genap atau terjadi OVERFLOW
3 (X)	0	TIDAK DIGUNAKAN
	1	

Bit	Logika	Makna
4 (HC)	0	Tidak ada Carry dari Bit 3 ke Bit 4
	1	Ada Carry dari Bit 3 ke Bit 4
5 (X)	0	TIDAK DIGUNAKAN
	1	
6 (Z)	0	Hasil Operasi ALU tidak sama dengan NOL
	1	Hasil Operasi ALU sama dengan NOL
7 (S)	0	Hasil Operasi ALU PLUS
	1	Hasil Operasi ALU MINUS

Gambar 2.10. Susunan dan fungsi logika register F mikroprosesor Z-80 CPU.

### ➤ Carry dan Half Carry

- Carry dicatat pada bit B0 sedangkan Half Carry dicatat pada bit B4. Carry adalah limpahan yang terjadi dari bit B7 ke bit B8 untuk operasi 8 bit dan limpahan dari bit B15 ke bit B16 untuk operasi 16 bit.
- Bit carry bernilai 1 jika sebuah operasi penjumlahan 8 bit melebihi FFh = 255d dan untuk operasi 16 bit melebihi nilai FFFFh = 65535d.
- Half Carry adalah limpahan yang terjadi dari bit B3 ke bit B4 untuk operasi 8 bit dan limpahan dari bit B7 ke bit B8 untuk operasi 16 bit.
- Untuk lebih memahami carry dan half carry perhatikan contoh-contoh berikut.

Contoh :

Penjumlahan data F3H dengan 24H

$$\begin{array}{r}
 1\ 1\ 1\ 0\ 0\ 0\ 0 \\
 \text{F3} = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\
 \text{24} = 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0 \\
 \hline
 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1
 \end{array}$$

Half Carry ; HC = 0

Ada Carry ; C = 1

Penjumlahan data 0BH dengan 2CH

$$\begin{array}{r}
 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \\
 \text{0B} = 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1 \\
 \text{2C} = 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0 \\
 \hline
 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 1
 \end{array}$$

Half Carry ; HC = 1

Tidak Ada Carry ; C = 0

- Pada operasi pengurangan SUB Bit Carry pada Flag dapat bermakna sebagai borrow. Dalam hal ini nilai Flag N = 1.
- Bit carry flag digunakan sebagai pendeteksi status dalam operasi JP C, JP NC, JR C, JR NC, CALL C, CALL NC, RET C, RET NC.

### ➤ Parity dan Overflow

- Digunakan untuk dua fungsi berbeda dalam satu bit.
- Bit B2 dinyatakan sebagai pencatat Paritas jika operasi sebelumnya adalah operasi logika dan B2 sebagai pencatat Overflow jika operasi sebelumnya adalah operasi aritmetika.
- Jika operasi logika menghasilkan bit "1" dalam jumlah yang genap maka P=1 dan jika operasi logika menghasilkan bit "1" dalam jumlah yang ganjil maka P=0.
- Overflow dapat diartikan sebagai suatu keadaan melimpah atau luber yaitu suatu keadaan pada operasi bilangan biner bertanda komplemen 2 melebihi batas maksimum range (-128 sampai dengan +127). Secara hukum matematis overflow menandakan suatu keadaan yang salah. Yaitu positif tambah positif hasilnya negatif atau negatif tambah negatif hasilnya positif.

### ➤ Zerro

- Sebagai penunjuk apakah hasil operasi ALU bernilai nol atau tidak.
- Sangat efektif digunakan untuk pendeteksian pencabangan dalam perintah JP Z, JP NZ, JR Z, JR NZ, DJNZ, CALL Z, CALL NZ, RET Z, RET NZ.
- Sering membingungkan bagi pemula karena jika hasil operasi sama dengan nol Z=1, dan jika hasil operasi tidak sama nol Z=0.

### ➤ Sign

- Bit penanda bilangan ini memberikan tanda apakah nilai hasil operasi ALU positif atau negatif. Positif atau negatifnya hasil ALU ditentukan oleh nilai bit B7 (MSB). Jika bit B7=1 maka nilai bilangan tersebut adalah negatif dan jika bit B7=0 maka nilai bilangan tersebut adalah positif.
- Bit Sign diperhatikan jika bekerja dalam format bilangan bertanda (signed bit), sedangkan jika bekerja dalam format bilangan tidak bertanda (*unsigned bit*) maka bit sign diabaikan.

## 3.2. MODE PENGALAMATAN (ADDRESSING MODE)

### 3.2.1. Immediate Addressing Mode

Immediate addressing mode disebut juga dengan istilah pengalamatan segera merupakan cara yang paling sederhana untuk membangkitkan data pada destinasi dengan cara membuat data menjadi bagian dari opcode. Sumber data secara langsung dinyatakan sebagai bagian dari perintahnya. Pada saat Z-80 CPU mengeksekusi perintah ini, program counter secara otomatis naik satu digit untuk mengambil data secara langsung dari memori.



Pola *immediate addressing mode* dan contoh perintahnya adalah sebagai berikut:

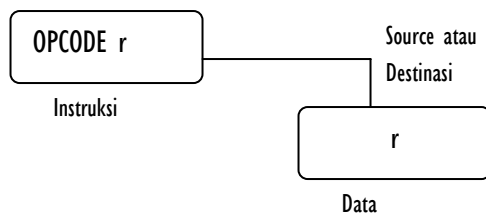
Mnemonic	Operand	Op code	Data
<b>LD</b>	<b>A, FE</b>	<b>3E</b>	<b>FE</b>

Contoh:

- o Assembly                      Operasi  
LD A, 01h                      copy data 01h ke Register A
- LD B, 1Ch                      copy data 1Ch ke Register B
- LD HL, ABCDh                copy data ABCDh ke Register HL

### 3.2.2. Register Addressing Mode

Register addressing mode adalah model pengalamatan alih data dimana nama register A, B, C, D, E, H, L, IX, IY, dan SP digunakan sebagai bagian dari opcode mnemonic baik sebagai source atau sebagai destinasi. Model dan contoh register addressing mode sebagai berikut:



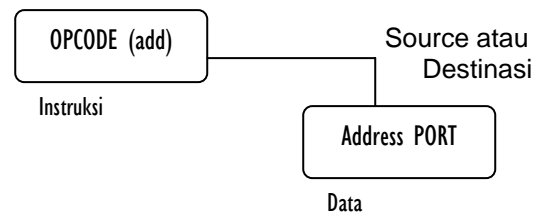
Contoh:

- o Assembly                      Operasi  
LD A, B                      copy data pada register B ke register A
- LD B, C                      copy data pada register C ke register B

- LD SP, HL                    copy data pada register HL ke register SP
- ADD A, B                    jumlahkan data register B ke register A
- AND C                      operasikan AND data register A dengan data Register C

### 3.2.3. Direct Addressing Mode (Pengalamatan Langsung)

Direct addressing mode menggunakan pengalamatan dengan menunjukan alamat secara langsung salah satu dari 256 byte alamat I/O. Model dan contoh direct addressing mode sebagai berikut:



Contoh:

- o Assembly                      Operasi  
IN A, PORTA                copy data dari Port A ke register A
- IN A, PORTB                copy data dari Port B ke register A
- OUT PORTC, A              copy data dari register A ke Port C
- OUT PORTB, A              copy data dari register A ke Port B

### 3.2.4. Indirect Addressing Mode

- o Menggunakan register sebagai pencatat atau pemegang alamat aktual yang akan digunakan untuk memindahkan data
- o Register itu sendiri bukan alamat

- o Menggunakan Register BC, DE, HL, dan SP sebagai Pointer data

Contoh:

- | o Assembly | Operasi   |
|------------|---|
| LD A, (HL) | copy data dari alamat yang dicatat oleh HL ke register A  |
| LD B, (DE) | copy data dari alamat yang dicatat oleh DE ke register B  |
| LD (HL),A  | copy data dari register A ke alamat yang tercatat oleh HL |

### 3.2.5. Indexed Addressing Mode

- o Menggunakan register sebagai pencatat atau pemegang alamat aktual yang akan digunakan untuk memindahkan data
- o Register itu sendiri bukan alamat
- o Menggunakan Register IX dan IY sebagai Pointer data

Contoh:

- | o Assembly   | Operasi   |
|--------------|---|
| LD A, (IX+d) | copy data dari alamat yang dicatat oleh IX+d ke register A  |
| LD B, (IY+d) | copy data dari alamat yang dicatat oleh IY+d ke register B  |
| LD (IX+d),A  | copy data dari register A ke alamat yang tercatat oleh IX+d |

### 3.2.6. Extended Addressing Mode

- o Menggunakan data immediate 16 bit sebagai pencatat atau pemegang alamat aktual yang akan digunakan untuk memindahkan data

Contoh:

- | o Assembly   | Operasi                                  |
|--------------|--|
| LD A, (1800) | copy data dari alamat 1800 ke register A |
| LD B, (1900) | copy data dari alamat 1900 ke register B |
| LD (1902),A  | copy data dari register A ke alamat 1902 |

### 3.2.7. Relatif Addressing Mode

- o Mode pengalamatan dengan nilai offset diantara  $-120$  s/d  $+127$ .
- o Bergerak mundur dari posisi program counter bernilai negatif
- o Nilai offset :
  - $e = (\text{alamat sumber} + 2) - \text{Alamat tujuan}$  ; dimana hasil e di komplemen duakan
- o Bergerak maju dari posisi program counter bernilai positif
- o Nilai offset:
  - $e = \text{Alamat tujuan} - (\text{alamat sumber} - 2)$

## 4. Kemasan Mikroprosesor

Ada empat jenis bentuk kemasan mikroprosesor yaitu:

- ✦ PDIP : *Plastic Dual Inline Package*
- ✦ PLCC : *Plastic J-Lieded Chip Carrier*
- ✦ TQFP : *Plastic Gull Wing Quad Flat Package*
- ✦ SOIC : *Plastic Gull-wing Small Outline*