**Intruksi set 8085**

Untuk dapat menggunakan suatu mikroprosesor, maka harus mengetahui set instruksi dari mikroprosesor yang bersangkutan. Set instruksi dari suatu mikroprosesor berisikan daftar seluruh instruksi-instruksi yang dapat dilakukan oleh mikroprosesor tersebut. Pada mikroprosesor 8085 terdapat lebih dari 70 instruksi yang terbagi menjadi beberapa kelompok instruksi, yaitu :

1.      Kelompok instruksi transfer data

2.      Kelompok instruksi aritmatika

3.      Kelompok instruksi geser (“shift” dan “rotate”)

4.      Kelompok instruksi logika

5.      Kelompok instruksi pencabangan (“branch)

6.      Kelompok instruksi input/output dan control

Kelompok Instruksi :

**1. Transfer Data**

Di antara semua kelompok instruksi pada mikroprosesor 8085, kelompok instruksi transfer data menunjukkan bagaimana cara memindahkan data 8 bit dari suatu register ke register lain atau antara memori dan register.  Kelompok instruksi transfer data terdiri dari : MOV,  MVI,  STA,  LDA, STAX,  LDAX,  LHLD,  XCHG,  XTHL,  SHLD, dan LXI. Instruksi-instruksi tersebut tidak mengubah isi register flag.

**A. Instruksi MOV (Move)**

* Instruksi **MOV r1, r2** adalah instruksi untuk mentransfer data dari r2(register sumber) ke r1 (register tujuan). Pengertian kata ‘mentransfer´ di sini adalah ‘menggandakan´ atau ‘mengkopi´, karena pada akhir pelaksanaaninstruksi, isi register tujuan jadi sama dengan isi register sumber.
* Instruksi **MOV r, M** berfungsi untuk mentransfer data dari lokasi memori yang alamatnya berada pada pasangan register HL ke suatu register 8 bit. Alamat memori M ditunjukkan oleh isi pasanganregister HL.
* Instruksi **MOV M,** r merupakan kebalikan dari instruksi MOV r, M.Instruksi MOV M, r ini akan memindahkan data 8 bit yang berada pada suaturegister ke suatu lokasi memori yang alamatnya ditunjukkan oleh pasanganregister HL.

**B. Instruksi  MVI (Move Immediate)**

Instruksi  MVI  r, data 8 bit´ dan instruksi  MVI M, data 8 bit.

* Instruksi MVI M, 8 bit, digunakan untuk mengisi data 8 bit pada suatu lokasi memori yang alamatnya ada pada register HL.

**C. Instruksi STA (Store Accumulator Directly in Memory) dan LDA (Load Accumulator Directly from Memory)**

* Instruksi ‘ STA’ dipergunakan untuk menyimpan data yang berada pada akumulator (register A) ke dalam unit memori pada lokasi alamat (4 digit heksadesimal dari belakang mnemonik STA).
* Instruksi  LDA  digunakan untuk memindahkan data 8 bit dari unit memori lokasi alamat tertentu ke akumulator (register A).

**D. Instruksi LHLD (Load H & L Registers Directly from Memory)**

Berfungsi menyalin data memori pada alamat yang spesifik.

**E. Instruksi SHLD (Store H & L Registers Directly in Memory)**

Berfungsi menyimpan data register H & L langsung dalam memori.

‘X’ dalam nama instruksi transfer data menunjukkan bahwa itu berhubungan dengan sepasang  register (16 bit).

**F. Instruksi STAX (Store Accumulator in Address in Register Pair) dan LDAX (Load Accumulator from Address in Register Pair)**

* Instruksi ‘STAX rp’ digunakan untuk menyimpan isi akumulator kelokasi memori yang alamatnya ditunjukkan oleh suatu pasangan register (rp).
* Instruksi ‘LDAX rp’ berfungsi mengisi akumulator dengan data 8 bit dari lokasi memori yang alamatnya ditunjukkan oleh suatu pasangan register.

**G. Instruksi LXI (Load Register Pair with Immediate data)**

Instruksi LXI rp, d16 ; berfungsi mengisi register pair (rp) dari nilai data d16 (alamat 16 bit).

**H. Instruksi XCHG (Exchange H & L with D & E)**

Berfungsi menukar register H & L dengan register D & E.

**I. Instruksi XTHL ( Exchange Top of Stack with H & L)**

**2. Aritmatika**

Instruksi aritmatika contohnya adalah penjumlahan, penguranagan,penambahan dengan 1 (increment) dan pengurangan dengan 1 (decrement).

Pada bagian ini terdapat berbagai intruksi aritmatika diantaranya : ADD, ADI,ADC, ACI, SUB, SUI, SBB, SBI, INR, DCR, INX, DCX, DAD dan DAA.

**A. Instruksi ADD dan SUB**

* Intruksi ADD dipergunakan untuk melakukan operasi penjumlahan 8 bit.Pada operasi tersebut , isi register A (akumulator) akan ditambahkan dengan isi register 8 bit atau isi lokasi memori yang alamatnya ditunjukan oleh pasangan register HL.
* Intruksi SUB dipergunakan untuk melakukan operasi pengurangan 8bit.Pada operasi tersebut, isi register A (akumulator) akan dikurangi dengan isi register 8 bit, atau isi lokasi memori yang alamatnya ditunjukan oleh pasanganregister HL.

**B. Instruksi ADI dan SUI**

* Instruksi ADI dipergunakan untuk menambahkan isi register A (akumulator)dengan data 8 bit secara langsung (immediate data).
* Intruksi SUI dipergunakan dipergunakan untuk mengurangkan isi register A(akumulator) dengan data 8 bit secara langsung (immediate data).

**C. Instruksi ADC dan SBB**

* Instruksi ADC dipergunakan untuk menambah isi register dengan data 8 bitdan mengikutsertakan carry BIT (Cy) dari operasi sebelumnya. Data 8 bit tersebutdapat diambil dari suatu register atau suatu lokasi memori yang alamatnyaditunjukan oleh register HL.
* Instruksi SBB berfungsi untuk mengurangi isi register A dengandata 8 bitdan mengikutsertakan Borrow bit (Cy) dari operasi sebelumnya. Data 8 bittersebut dapat diambil dari suatu register atau suatu lokasi memori yangalametnya ditunjukan oleh register HL.

**D. Instruksi ACI dan SBI**

* Instruksi ACI berfungsi untuk menambah isi register A dengan data 8 bitsecara langsung (immediate data) dan mengikutsertakan carry bit (Cy) darioperasi sebelumnya.
* Instruksi SBI berfungsi untuk mengurangi isi register A dengan data 8 bitlangsung (immediate data) dan mengikutsertakan Borrow bit (Cy) dari operasisebelumnya.

Instruksi ADD, ADI, ADC, SUB, SBI, SUI dan SBB akan mempengaruhi regidter flag (S, Z, AC, P, dan Cy).

**E. Instruksi INR dan INX**

Perintah penambahan 1 sebenarnya  hanya menambahkan isi suatu lokasi (register atau memori) dengan 1. Untuk 8085 ada 2 perintah yaitu INR dan INX. Semua flag kecuali Cy (sisa) dipengaruhi oleh perintah INR dan tidak ada bendera yang dipengaruhi oleh perintah INX.

* Instruksi INR dipergunakan untuk menambah (menaikan)  dengan 1 isi dari suaturegister 8 bit atau lokasi memori yang alamatnya ditunjukan oleh regidter HL.
* INX  dipergunakan untuk menambah dengan 1 isi suatu pasangan register (16 bit register).

**F. Instruksi DCR dan DCX**

* Instruksi DCR dipergunakan untuk mengurangi dengan 1 isi dari suatu register 8 bitatau suatu lokasi memori yang alamatnya ditunjukan oleh register HL.
* Instruksi DCX  dipergunakan untuk mengurangi dengan 1 isi suatu pasangan register (16 bit register).

Instruksi INR dan DCR akan mempengaruhi atau mengubah bit ± bit dari register flag seperti S, Z dan AC tetapi tidak mempengaruhi bit Cy. Instruksi INX danDCX tidak mempengaruhi bit ± bit dari register flag.

**G. Instruksi DAD**

Instruksi DAD dipergunakan untuk menambahkan isi pasangan register HLdengan isi suatu pasangan register pada operasi penjumlahan16 bit.

**H. Instruksi DAA**

Instruksi untuk mengatur bentuk desimal.

**3. Geser**

Instruksi-instruksi geser digunakan untuk menggeser isi register A.Instruksi-instruksi geser hanya mempengaruhi Carry flag (CY).

**A. Instruksi RAR ( Rotate A Right through carry )**

RAR adalah instruksi yang digunakan untuk menggeser isi register A ke kanan melalui carry flag. LSB dari data yang berada pada register A akan mengisi carry flag, sedangkan isi carry flag sebelumnya pindah menjadi MSB.

**B. Instruksi RAL ( Rotate A Left through carry )**

RAL adalah instruksi yang digunakan untuk menggeser isi register A ke kiri melalui carry flag. MSB dari data yng berada pada register A akan mengisi carryflag, sedangkan isi carry flag sebelumnya pindah menjadi LSB.

**C. Instruksi RRC ( Rotate A Right )**

RRC adalah instruksi yang digunakan untuk menggeser isi register A kekanan. LSB akan mengisi MSB dan juga akan mengisi carry flag, sedangkan datase belumnya yang ada di carry flag akan hilang.

**D. Instruksi RLC ( Rotate A Left )**

RLC adalah instruksi yang digunakan untuk menggeser isi register A ke kiriMSB akan mengisi LSB dan juga akan mengisi carry flag, sedangkan datase belumnya yang ada di carry flag akan hilang.

**4. Logika**

**A. Instruksi logika AND**

* ANA M ( And Memory with A )

ANA M adalah instruksi yang digunakan untuk melakukan operasi logika and antara isi register A dengan memori. Kemudian hasil operasi disimpan secara otomatis di register A.

* ANA R ( And Register with A )

ANA R adalah instruksi yang digunakan untuk melakukan operasi logika and antara isi register A dengan isi suatu register lain. Kemudian hasil operasidisimpan secara otomatis di register A.

* ANI ( And Immediate with A )

ANI adalah instruksi yang digunakan untuk melakukan operasi logika andantara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika AND  mempengaruhi isi register flag, yaitu Zero flag (Z), Sign flag (S),  Parity flag (P),  Carry flag selalu dalam kondisi reset (CY = 0) dan Auxiliary Carry selalu dalam kondisi set (AC = 1).

**B. Instruksi OR**

* ORA M ( Or Memory with A )

ORA M adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan memori. Kemudian hasil operasi disimpan secara otomatis di register A.

* ORA R ( Or Register with A )

ORA R adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan isi suatu register lain.  Kemudian hasil operasi disimpan secara otomatis di register A.

* ORI ( Or Immediate with A )

ORI adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika OR mempengaruhi isi register flag, yaitu Zero flag (Z), Signflag (S),  Parity flag (P),  Carry flag selalu dalam kondisi reset (CY = 0) dan Auxiliary Carry selalu dalam kondisi reset (AC = 0).

**C. Instruksi XOR**

* XRA M ( Exclusive Or Memory with A )

XRA M adalah instruksi yang digunakan untuk melakukan operasi logikaexclusive or antara isi register A dengan memori. Kemudian hasil operasidisimpan secara otomatis di register A.

* XRA R ( Exclusive Or Register with A )

XRA R adalah instruksi yang digunakan untuk melakukan operasi logika exclusive or antara isi register A dengan isi suatu register lain. Kemudian hasil operasi disimpan secara otomatis di register A.

* XRI ( Exclusive Or Immediate with A )

XRI adalah instruksi yang digunakan untuk melakukan operasi logika exclusive or antara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika Exclusive Or mempengaruhi isi register flag, yaitu Zero flag(Z), Sign flag (S), Parity flag (P), Carry flag selalu dalam kondisi reset (CY = 0)dan Auxiliary Carry selalu dalam kondisi reset (AC = 0).

**5. Pencabangan**

Instruksi pencabangan dibagi menjadi dua, yaitu :

**A. Pencabangan bersyarat (conditional)** : JC, JNC, JZ, JNZ, JP,JM, JPE, JPO.

* JC  (J ump if  Carry Set)

Perintah ini digunakan apabila kita ingin menguji carry bit. Jika carry bit berlogika ‘1’, maka program akan melompat ke alamat yang ditulis pada perintah ’JC addr’.

* JNC  (Jump if Carry Not Set )

Perintah ini digunakan apabila kita ingin menguji carry bit. Jika carry bitberlogika ‘0’, maka program akan melompat ke alamat yang ditulis pada perintah ’JNC addr’.

* JM  (J ump if Minus)

Perintah ini digunakan apabila kita ingin menguji sign bit. Program akan melompat pada alamat tertentu apabila sign bit berlogika ‘1’.

* JP  (J ump if plus)

Perintah ini digunakan apabila kita ingin menguji sign bit. Jika sign bit berlogika ‘0’, maka program akan melompat ke alamat yang ditulis setelah perintah JP.

* JZ  (Jump if Zero)

Perintah ini digunakan apabila kita ingin menguji zero bit. Jika zero bi tberlogika ‘1’ maka program akan melompat ke alamat yang ditulis setelah perintah JZ.

* JNZ (J ump if not zero)

Perintah ini digunakan apabila kita ingin menguji zero bit. Jika zero bit berlogika ‘0’ maka program akan melompat ke alamat yang ditulis setelah perintah JNZ.

* JPE (J ump if Even Parity)

Perintah ini digunakan apabila kita ingin menguji parity bit. Jika parity bit berlogika ‘1’, artinya jumlah logika satu dari suatu data hasil operasi yang mempengaruhi register flag berjumlah genap, maka program akan melompat ke alamat yang ditulis setelah perintah JPE.

* JPO (J ump if Parity Odd)

Perintah ini digunakan apabila kita ingin menguji parity bit. Jika parity bit berlogika ‘0’, artinya jumlah logika satu dari suatu data hasil operasi yang mempengaruhi register flag berjumlah ganjil, maka program akan melompat ke alamat yang ditulis setelah perintah JPO.

**B. Pencabangan tidak bersyarat** : JMP (jump) dan CALL.

* JMP  (Jump)

Perintah ini digunakan apabila kita ingin melompat ke alamat tertentu, akan tetapi tanpa melibatkan proses pengujian terhadap bit-bit yang terdapat pada register flag. Oleh karena itu, perintah jump dimasukkan ke dalam perintah pencabangan tak bersyarat.

* CALL

Perintah ini digunakan apabila kita ingin menggunakan suatu subrutin.

**6. IN/OUT dan Control**

Instruksi IN dan OUT berfungsi untuk mengatur lalu lintas dengan peralatan I/O. Dalam instruksi OUT, sumber selalu accumulator. Dalam instruksi IN, tujuan adalah accumulator. Di belakang operasi IN dan OUT kita hanya perlu mencantumkan nomor pintu ( gate number ) saja, yaitu alamatnya peralatan I/O.

Instruksi Machine Control :

* Instruksi SIM

Berfungsi membuat settingan mask interrupt pada mesin prosesor 8085.

* Instruksi RIM

Berfungsi  membaca mask interrupt pada mesin prosesor 8085.

* Instruksi DI

Berfungsi mengnonaktifkan system interrupt pada mesin prosesor 8085.

* Instruksi EI

Berfungsi mengaktifkan system interrupt pada mesin prosesor 8085

* Instruksi HLT

Berfungsi memberhentikan mesin

* Instruksi NOP

Berfungsi tidak ada operasi apapun pada kontrol mesin