

IDENTIFIKASI JENIS KODE DALAM INSTRUCTION SET INTEL DAN ARM

Disusun Guna Memenuhi Tugas Organisasi dan Arsitektur
Komputer Semester V
Pengampu: Bana Handaga, Dr. Ir, M.T



Oleh:

Widhar Dwiatmoko
L200160179

**PROGAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH
SURAKARTA 2017**

1. INSTRUCTION SET PROCESSOR INTEL

1. DATA TRANSFER INSTRUKSI

Nama	Fungsi
MOV Rd, Rs	Mengcopy nilai dari Rs ke Rd
MOV Rd, M	Mengcopy nilai dari M ke Rd
MOV M, Rs	Mengcopy nilai dari M ke Rs
MVI Rd, d8	Memindahkan nilai register d8 ke register d8
MVI M, d8	Memindahkan nilai register d8 ke register M
LDA addr16	Menyalin data memori pada alamat yang spesifik addr16
LDAX rp	Mengcopy data pada register pair (rp)
LXI rp, d16	Mengisi register pair (rp) dari nilai data d16 (alamat 16 bit)
LHLD addr16	Menyalin data memori pada alamat yang spesifik addr16
STA addr16	Menyimpan nilai data langsung dalam memori addr16
STAX	Menyimpan nilai data pada alamat register pair (rp)
SHLD addr16	Menyimpan data register H & L langsung dalam memori alamat addr16
SPHL	Memindahkan isi dari H & L ke pointer stack
XCHG	Menukar register H & L dengan register D & E
XHTL	Menukar stack tertinggi dengan register H & L
PUSH rp	Push 2 byte data ke stack pada register pair (rp)
PUSH PSW	Push 2 byte data ke stack pada processor status word (8-bit)
POP rp	Pop Two Bytes of Data off the Stack

2. ARITHMETIC INSTRUKSI

Nama	Fungsi
ADD reg	Instruksi penambahan pada register reg
ADD M	Instruksi penambahan pada register M
ADI d8	Instruksi penambahan data secara immediate pada register d8
ADC reg	Instruksi penambahan menggunakan carry flag pada register reg
ADC M	Instruksi penambahan menggunakan carry flag pada register M

ACI d8	Instruksi penambahan data d8 secara immediate menggunakan carry
DAA	Instruksi untuk mengatur bentuk desimal
DAD rp	Penambahan register pair ganda ke H & L register pair (rp)
SUB reg	Instruksi pengurangan pada register reg
SUB M	Instruksi pengurangan pada register M
SUI d8	Instruksi pengurangan data pada d8 secara immediate
SBB reg	Instruksi pengurangan menggunakan carry flag pada register reg
SBB M	Instruksi pengurangan menggunakan carry flag pada register M
SBI d8	Instruksi pengurangan secara immediate menggunakan carry flag pada register d8
INR reg	Instruksi kenaikan data reg setiap 1 byte
INR M	Instruksi kenaikan data M setiap 1 byte
INX rp	Instruksi kenaikan 1 data register pair (rp)
DCR reg	Instruksi penurunan data reg setiap 1 byte

3. LOGIKA INSTRUKSI

Nama	Fungsi
ANA reg	Menggunakan logika AND dengan logika accumulator pada data reg
ANA M	Menggunakan logika AND dengan logika accumulator pada data M
ANI d8	Menggunakan logika AND dengan logika accumulator immediate d8
ORA reg	Menggunakan logika OR dengan logika accumulator OR pada reg
ORA M	Menggunakan logika OR dengan logika accumulator OR pada M
ORI d8	Menggunakan logika OR dengan logika accumulator OR immediate register d8
XRA reg	Menggunakan logika eksklusif OR dengan logika accumulator eksklusif OR reg
XRA M	Menggunakan logika eksklusif OR dengan logika accumulator eksklusif OR reg M
XRI d8	Menggunakan logika eksklusif OR dengan data immediate pada register d8
CMP reg	Membandingkan data pada reg
CMP M	Membandingkan data pada register M
CPI d8	Membandingkan data secara immediate pada d8
CMA	Pelengkap accumulator data pada prosesor 8085
CMC	Pelengkap carry flag pada prosesor 8085

STC	Pengatur/set/setting carry flag
RLC	Pengatur rotasi/putaran accumulator pada bagian kiri

4. BRANCHING INSTRUKSI

Nama	Fungsi
JMP addr16	Membuat program beralih/loncat ke addr16
CALL addr16	Memanggil data pada addr16
RET	Kembali pada instruksi awal
RST n	Instruksi restart secara khusus
PCHL	Memindahkan H & L pada program counter

5. MACHINE CONTROL INSTRUKSI

Nama	Fungsi
SIM	Membuat settingan mask interrupt pada mesin prosesor 8085
RIM	Membaca mask interrupt pada mesin prosesor 8085
DI	Mengnonaktifkan system interrupt pada mesin prosesor 8085
EI	Mengaktifkan system interrupt pada mesin prosesor 8085
HLT	Memberhentikan mesin
NOP	Tidak ada operasi apapun pada kontrol mesin

2. Set Instruksi pada Mikroprocessor 8086 dan 8088 (ARM)

Berikut ini penjelasan tentang instruksi yang digunakan pada bahasa rakitan 8086 dan 8088 atau sejenisnya :

No	Nama	Fungsi
1.	AAA (ASCII adjust for edition)	Pengaturan ASCII bagi penambahan
2.	AAD (ASCII adjust for division)	Pengaturan ASCII bagi pembagian
3.	AAM (ASCII adjust for multipty)	Pengaturan ASCII bagi perkalian
4.	AAS (ASCII adjust for subtraction)	Pengaturan ASCII bagi pengurangan
5.	ADC (add with carry)	Tambahkan dengan carry
6.	ADD (addition)	Penambahan

7.	AND (logic AND)	Logik AND
8	CALL (CALL subroutine)	Subrutin panggil
9.	CBW (Convert byte to word)	Konversikan byte ke kata
10.	CLC (Clear Carry)	Kosongkan carry flag
11.	CLD (Clear direction flag)	Kosongkan flag arah
12.	CLI (Clear interrupt enable)	Kosongkan flag penggerak interupsi
13.	DAS (decimal adjust for subtraction)	Pengaturan decimal bagi pengurangan
14.	DEC (decrement)	Penurunan operand tujuan dengan 1
15.	DIV (divide)	Pembagian tak bertanda
16.	ESC (escape)	Sehubungan dengan suatu co-procesor external
17.	HLT (halt)	Menghentikan prosesor sampai saluran reset diaktifkan
18.	IDIV (Integer division)	Pembagian kilat
19.	IMUL (Integer multiply)	Perkalian kilat
20.	IN (Input) IN (masukan)	Mentransfer data dari port yang dispesifikasikan ke dalam register AC atau AX
21.	INC (Increment)	Menaikkan operand tujuan dalam 1
22.	INT (Interrupt) INT (Interupsi)	Mengawali suatu prosedur interupsi dengan jenis yang dispesifikasikan oleh instruksi yang bersangkutan
23.	INTO	Interupsi bila ada overflow INTO digunakan untuk membangkitkan suatu interupsi perangkat lunak yang bergantung pada status flag OF
24.	IRET (Interrupt return)	Kembali dari suatu interupsi dan mendapatkan kembali IP,CS dan flag-flag dari stack
25.	JAE (Jump on above)	Jika operand pertama lebih besar daripada operand kedua

26.	JNBE (Jump)	Digunakan jika pada saat pembandingan operand 1, tidak lebih kecil atausama dengan operand 2.
27.	JA E (Jump on above or equal)	Digunakan pada saat pembandingan operand 1 lebih besar atau sama dengan operand 2.
28.	JNB (Jump not below)	Digunakan pada saat pembandingan operand 1 tidak lebih kecil dari operand 2.
29.	JB (Jump on below)	Digunakan pada saat operand 1 lebih kecil dari operand 2.
30.	JNA E (Jump on not above or equal)	Pada saat operand 1 tidak lebih besar atau sama dengan operand 2.
31.	JBE (Jump on below or equal)	Digunakan pada saat operand 1 lebih kecil atau sama dengan operand 2.
32.	JNA (Jump on not above)	Digunakan pada saat operand 1 tidak lebih besar dari operand 2
33.	JC (Jump on carry)	Digunakan pada saat akan diprogram CF=1.
34.	JCXZ (Jump if CX=0)	Digunakan bila isi register CX=0.
35.	JE (Jump on equal)	Digunakan pada saat pembandingan kedua operand sama.
36.	JZ (Jump on zero)	Digunakan pada saat akan diproses ZF=1
37.	JG (Jump on greather than)	Digunakan pada saat pembandingan operand 1 menunjukkan lebih besar dari operand 2
38	JNLE (Jump on greather or equal)	Digunakan pada saat operand 1 tidak lebih kecil atau sama dengan operand 2
39.	JGE (Jump on greather or equal)	Digunakan pada saat operand 1 lebih besar atau sama dengan operand 2.

40.	JNL (Jump on less)	Digunakan pada saat operand 1 tidak lebih kecil dari operand 2.
41.	JL (Jump on less)	Digunakan pada saat operand 1 lebih kecil dari operand 2.
42.	JNGE (Jump on not greather or equal)	Digunakan pada saat operand 1 tidak lebih besar atau sama dengan operand 2.
43.	JLE (Jump on less or equal)	Digunakan pada saat operand 1 lebih kecil atau sama dengan operand 2.
44.	JNG (Jump on not greather than)	Digunakan pada saat operand 1 tidak lebih besar daripada operand 2
45.	JMP (Unconditional jump)	Lompatan tidak bersyarat
46.	JNC (Jump on not carry)	Digunakan pada saat diproses CF=0
47.	JNE (Jump on not equal)	Digunakan pada saat operand 1 tidak sama dari operand 2.
48.	JNO (Jump on not zero)	Digunakan pada saat diproses ZF=0
49.	JNO (Jump on not overflow)	Digunakan bila tidak ada overflow (OF=0).
50.	JNS (Jump on not sign)	Digunakan pada saat diproses SF=0
51.	ZNP (Jump on not pority)	Digunakan bila tidak ada pointer
52.	JPO (Jump on poity odd)	Digunakan bila pointer ganjil
53.	O (Jump on overflow)	Digunakan pada saat ada overflow
54.	JP (Jump on Pority equal)	Digunakan bila pointer genap
55.	JS (Jump on sign)	Digunakan pada saat diproses ZF=1
56.	LAHF (Load AH from flag)	LAHF me-load bit 7,6,4,2 & 0 pada register AH masing-masing dengan isi flag SF,ZF,AF & CF.
57.	LDS (Load pointer using DS)	Muatkan peminjak dengan menggunakan DS.
58.	LEA (Load effective address)	LEA mentransfer opsan sumber 16 bit dalam memori ke tujuan 16 bit.
59.	LES (Load pointer using ES)	Muatkan pointer dengan menggunakan ES
60.	LOCK (Lock bas	Digunakan dalam penerapan pemakaian sumber bersama, untuk memastikan bahwa

		memori tidak diakses secara serentak oleh lebih dari satu proses.
61.	LODS (Load string)	Muatkan string byte atau kata
62.	LOOP	Loop jika EX bukan
63.	LOOPE dan LOOPZ -	-Loop bila sama
64.	Loop bila 0	bila 0
65.	LOOPNE/LOOPNZ	Loop bila tidak sama (ZF=0)
66.	Loop	bila tidak nol (ZF=O)
67.	MOVE	Pindahkan byte atau kata memindahkan data 8 bit atau 16 bit.
68.	MOUS (Move string)	Pindahkan data string 8 bit atau 16 bit
69.	MUL (Multifly)	Digunakan untuk mengalihkan isi bertandapada akumulator dengan suatu operand sumber yang dispesifikasikan
70.	NEG (Negate)	Mengurangkan suatu operand tujuan dari 0 & menyimpan hasil komplemen keduanya dalam tujuan
71.	NOP (Logic NOT)	NOP tidak membe rikan dampak tertentu (tidak ada operasi)
72.	NOT (Logic OR)	Operasi logic OR
73.	OR (Logic OR)	Operasi logic OR
74.	OUT (Output)	Digunakan untuk mentransfer data dari AL atau AX ke point (pangkalan)
75.	POP (POP from stack)	Keluaran data dari stack dari AL atau AX ke point (pangkalan)
76.	POPF (POP flag)	Keluarkan ke flag dan stack
77.	POSH (Push to stack)	Dorong sumber ke stack
78.	PUSHF (Push flag)	Dorong flag ke stack
79.	RCL (Rotate left with carry)	Putar ke kirin dengan carry 1
80.	RCR (Rotate right with carry)	Putar ke kanan dengan carry 1
81.	REP	Ulangi
82.	REE	Ulangi jika sama
83.	REPZ	Ulangi jika nol

84.	REPNE	Ulangi jika tidak sama
85.	REPNZ	Ulangi jika tidak nol
86.	RET (Return from subroutine)	Digunakan untuk kembali ke subrutin
87.	ROL (Rotate left)	Rotasi (putar) ke kiri 1
89.	SAHF (store AH in flags register)	Simpan AH dalam reegister flag
90.	SAL/SHL (Shift left arithmetic/logical)	Geser aritmatik/logika ke kiri 1
91.	SAR/SHR (Shift right arithmetic/logical)	Geser aritmatik/logika ke kanan 1
92.	SBB (Substract with borrow)	Kurangkan dengan borrow
93.	SCAS (Scan string)	Digunakan untuk mengurangi string tujuan yang dialamatkan a/n regiser DI,dari AX atau AL
94.	STC (Set carry flag)	Menset carry flag (CF)
95.	STD (Set direction flag)	Menset flag arah (DF)
96.	STI (Set interrupt enable flag)	Menset flag penggerak interupsi (IF)
97.	STOS (Substract)	Memindahkan operand sumber yang terkandung dalam AX atau AL ke suatu tujuan yang dialamatkan oleh DI
98.	SUB (Substract)	Mengurangkan sumber dari operand tujuan dan hasilnya ditepatkan dalam tujuan
99.	TEST (Logical comporison)	Menguji operand operand atau logic perbandingan
100	WAIT	Menunggu sampai saluran test aktif
101	XCHG (Exchange)	Pertukaran dan sumber
102	XLAT (Translate)	Digunakan untuk menerjemahkan se-byte dalam register AL ke dalam suatu byte yang diambil dari tabel terjemahan
103	XOR (Logic exclusive OR)	Operasi exclusive OR