INSTRUKSI DAN BAHASA PEMOGRAMAN MIKROKONTROLER

Yoyo Somantri dan Erik Haritman
Dosen Jurusan Pendidikan Teknik Elektro
FPTK Universitas Pendidikan Indonesia

Pendahuluan

Dalam bab ini akan dibahas tujuan perkuliahan, tentang instruksi-instruksi pada mikrokontroler AT 89S51 dan bahasa pemograman mikrokontroler .

Tujuan Perkuliahan

Setelah mempelajari bab ini, diharapkan mahasiswa mampu untuk :

1. Memahami instruksi-instruksi mikrokontroler
2. Memahami bahasa pemograman

Instruksi – instruksi dan bahasa pemograman

Instruksi-instruksi pada keluarga MCS-51 dibagi dalam 5 kelompok, yaitu :

(Sumber dari : Atmel, (2005**)** *Data Book Microcontroller dan* Intel, (1994). *MCS'51 Microcontroller Family User Manual.)*

1. Instruksi operasi *Arithmetic*

- Operasi penjumlahan tanpa *Carry*

  Sintaks : ADD A, (src-byte)

  Operasi : A ← A + (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
ADD	A, Rn	1	1	0010 1rrr
ADD	A, *Direct*	2	1	0010 0101 (*direct address*)
ADD	A, @ Ri	1	1	0010 011I
ADD	A, #data	2	1	0010 0100 (*immediate data*)

- Operasi Penjumlahan dengan *Carry*

  Sintaks : ADDC A, (src-byte)

  Operasi : A ◄——— A + C + (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
ADDC	A, Rn	1	1	0011 1rrr
ADDC	A, *Direct*	2	1	0011 0101 (*direct address*)
ADDC	A, @ Ri	1	1	0011 011I
ADDC	A, #data	2	1	0011 0100 (*immediate data*)

- Operasi Pengurangan dengan *Borrow*

  Sintaks : SUBB A, (src-byte)

  Operasi : A ◄——— A - C - (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
SUBB	A, Rn	1	1	1001 1rrr
SUBB	A, *Direct*	2	1	1001 0101 (*direct address*)
SUBB	A, @ Ri	1	1	1001 011i
SUBB	A, #data	2	1	1001 0100 (*immediate data*)

- Operasi *Increment*

  Sintaks : INC (byte)

  Operasi : (byte) ◄—— (byte) + 1

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
INC	A	1	1	0000 0100
INC	Rn	2	1	0000 1rrr
INC	direct	1	1	0011 011I (direct address)
INC	@Ri	2	1	0011 0100
INC	DPTR	1	2	1010 0011

- Operasi *Decrement*

  Sintaks : DEC (byte)

  Operasi : (byte) ◄—— (byte) - 1

Anggotanya :

OpCode			*Byte*	*Cycles*	*Encoding*
DEC	A	1	1		0001 0100
DEC	Rn		1	1	0001 1rrr
DEC	direct		2	1	0001 010I (direct address)
DEC	@Ri		1	1	0001 011i

- Operasi Perkalian

  Sintaks : MUL AB

  Operasi : A_{7} -0 ← A x B

  $B_{15 - 8}$

OpCode		*Byte*	*Cycles*	*Encoding*
MUL	AB	1	4	1010 0100

- Operasi Pembagian

  Sintaks : DIV AB

  Operasi : $A_{15 - 8}$ ← A / B

  $B_{7 - 0}$

OpCode		*Byte*	*Cycles*	*Encoding*
DIV	AB	1	1	1101 0100

2. Instruksi Operasi Logical

- Operasi AND

  Sintaks : ANL (dest-byte) (src-byte)

  Operasi : (dest – byte) ← (dest – byte) (src-byte)

  Anggotanya :

OpCode		Byte	Cycles	Encoding
ANL	A, Rn	1	1	0101 1rrr
ANL	A, *Direct*	2	1	0101 0101 (*direct address*)
ANL	A, @ Ri	1	1	0101 011i
ANL	A, #data	2	1	0101 0100 (*immediate data*)
ANL	*direct*, A	2	1	0101 0010 (*direct address*)
ANL	direct, #data	3	2	0101 0011 (*direct addres*) (immediate data)

- Operasi OR

  Sintaks : ORL (dest-byte) (src-byte)

  Operasi : (dest – byte) ⟵ (dest – byte) (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
ORL	A, Rn	1	1	0100 1rrr
ORL	A, *Direct*	2	1	0100 0101 (*direct address*)
ORL	A, @ Ri	1	1	0100 011i
ORL	A, #data	2	1	0100 0100 (*immediate data*)
ORL	*direct*, A	2	1	0100 0010 (*direct address*)
ORL	*direct*, #data	3	2	01000011(*direct address*) (*immediate data*)

- Operasi XOR

  Sintaks : ORL (dest-byte) (src-byte)

  Operasi : (dest – byte) ⟵ (dest – byte) (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
XRL	A, Rn	1	1	0110 1rrr
XRL	A, *Direct*	2	1	0110 0101 (*direct address*)
XRL	A, @ Ri	1	1	0110 011i
XRL	A, #data	2	1	0110 0100 (*immediate data*)
XRL	*direct*, A	2	1	0110 0010 (*direct address*)
XRL	*direct*, #data	3	2	01100011(*direct addres*) (*immediate data*)

- Operasi CLEAR

  Sintaks : CLR A

  Operasi : A ⟵ 0

CLR	A	1	1	1110 0100

- Operasi *Complement*

  Sintaks : CPL A

  Operasi : A ⟵ A'

OpCode			*Byte*	*Cycles*	*Encoding*
CPL	A		1	1	1111 0100

- Operasi Rotasi tanpa *Carry*

  PUTAR KIRI

  Sintaks : RL A

  Operasi : An + 1 ⟵ An n = 0 – 6

  A0 ⟵ A7

OpCode			*Byte*	*Cycles*	*Encoding*
RL	A		1	1	0010 0011

  PUTAR KANAN

  Sintaks : RR A

  Operasi : An + 1 ⟵ An n = 0 – 6

  A7 ⟵ A0

OpCode			*Byte*	*Cycles*	*Encoding*
RR	A		1	1	0000 0011

- Operasi Rotasi dengan Carry

  PUTAR KIRI

  Sintaks : RL A

  Operasi : An + 1 ⟵ An n = 0 – 6

  A0 ⟵ C

  C ⟵ A7

OpCode			*Byte*	*Cycles*	*Encoding*
RLC	A		1	1	0011 0011

PUTAR KANAN

Sintaks : RRC A

Operasi : An + 1 ⟵ An n = 0 – 6

A7 ⟵ C

C ⟵ A7

OpCode		*Byte*	*Cycles*	*Encoding*
RLC	A	1	1	0001 0011

- Operasi PERTUKARAN NIBLE (4bit)

Sintaks : SWAP A

Operasi : A3-0 ⟷ A7-4

OpCode		Byte	Cycles	Encoding
SWAP	A	1	1	1100 0100

1. Instruksi Transfer Data

- Instruksi Transfer Data pada Memori Data Internal

Sintaks : MOV (dest-byte) (src-byte)

Operasi : (dest – byte) ⟵ (src-byte)

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
MOV	A, Rn	1	1	1110 1rrr
MOV	A, *Direct*	2	1	1110 0101 (*direct address*)
MOV	A, @ Ri	1	1	1110 011i
MOV	A, #data	2	1	0111 0100 (*immediate data*)
MOV	Rn, A	1	1	1111 1rrr
MOV	Rn, Direct	2	2	1010 1rrr (*direct address*)
MOV	Rn, #data	2	1	0111 1rrr (*immediate data*)
MOV	Direct, A	2	1	1111 0101 (*direct address*)
MOV	Direct, Rn	2	2	1000 1rrr (*direct address*)
MOV	Direct, Direct	3	2	1000 0101 (dir.addr.(src)) (dir.addr.(dest))
MOV	Direct, @Ri	2	2	1000 011I (*direct address*)

MOV	Direct, #data	3	2	01110101 (*direct address*)	(*immediate data*)
MOV	@Ri, A	1	1	1111 011I	
MOV	@Ri, Direct	2	2	1010 011I	
MOV	@Ri, #data	2	1	0111 011I (*immediate data*)	
MOV	DPTR, data16	3	2	1001 0000 (immd.data 15-8)	(immd.data 7-0)

- Operasi Transfer Data pada Memori Data external

  Sintaks : MOVX (dest-byte) (src-byte)

  Operasi : (dest – byte) ⟵ (src-byte)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
MOVX	A, @Ri	1	2	1110 001I
MOVX	A, @DPTR	1	2	1110 0000
MOVX	@Ri, A	1	2	1111 001I
MOVX	@DPTR, A	1	2	1111 0000

- Operasi Transfer Data pada Memori Program

  Sintaks : MOVC A, @A + (base-reg)

  Operasi : A ⟵ (A + (base-reg)

  Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
MOVC	A, @Ri	1	2	1001 0011
MOVC	A, @DPTR	1	2	1000 0011

- Operasi Transfer data pada *stack*
  - Simpan ke *stack*

    Sintaks : PUSH *Direct*

    Operasi : SP ⟵ SP + 1

    (SP) ⟵ (direct)

OpCode	*Byte*	*Cycles*	*Encoding*

PUSH *direct address*	2	2	1100 0000 (direct address)

- Ambil dari *stack*

  Sintaks : POP *Direct*

  Operasi : *Direct* ← (SP)

  SP ← SP – 1

OpCode	*Byte*	*Cycles*	*Encoding*
PUSH *direct address*	2	2	1100 0000 (*direct address*)

- Operasi Pertukaran Data
  - Pertukaran Byte

    Sintaks : XCH A, (byte)

    Operasi : A ⟷ (byte)

    Anggotanya :

OpCode	*Byte*	*Cycles*	*Encoding*
XCH A, Rn	1	1	1100 1rrr
XCH A, *direct*	2	1	1100 0101 (*direct address*)
XCH A, @Ri	1	1	1100 011I

  - Pertukaran Nible

    Sintaks : XCHD A, A, @Ri

    Operasi : A_{3-0} ⟷ (Ri_{3-0})

OpCode	*Byte*	*Cycles*	*Encoding*
XCHD A, @Ri	1	1	1101 011I

2. Instruksi Manipulasi Variabel Boolean

- Operasi *Clear Bit*

  Sintaks : CLR Bit

Operasi : Bit ⟵ 0

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
CLR	C	1	1	1100 001I
CLR	Bit	2	1	1100 0010 (bit address)

- Operasi *Set Bit*

Sintaks : SETB Bit

Operasi : Bit ⟵ 1

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
SETB	C	1	1	1101 001I
SETB	Bit	2	1	1101 0010 (*bit address*)

- Operasi *Complement Bit*

Sintaks : CPL Bit

Operasi : Bit ⟵ Bit'

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
CPL	C	1	1	1011 001I
CPL	Bit	2	1	1011 0010 (*bit address*)

- Operasi AND bit

Sintaks : ANL C, (src-bit)

Operasi : C ⟵ C ^ bit atau C ⟵ C ^ bit'

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
ANL	C, bit	2	2	1000 0010 (*bit address*)
ANL	C,/bit	2	2	1011 0000 (*bit address*)

- Operasi OR bit

Sintaks : ORL C, (src-bit)

Operasi : C ⟵ C V bit atau C ⟵ C V bit'

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*

ORL	C, bit	2	2	0111 0010 (*bit address*)
ORL	C,/bit	2	2	1010 0000 (*bit address*)

- Operasi Transfer Bit

Sintaks : MOV (dest-bit) , (src-bit)

Operasi : (dest-bit) ⟵ (src-bit)

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
MOV	C, bit	2	1	1010 0010 (*bit address*)
MOV	bit, C	2	2	1001 0010 (*bit address*)

- Operasi JUMP oleh Bit
  - JUMP Jika *Bit Set*

Sintaks : JB C, rel

Operasi : Jika bit = 1, maka PC ⟵ PC + rel

OpCode		*Byte*	*Cycles*	*Encoding*
JB	bit, rel	3	2	0010 0000 (*bit address*)(rel.*address*)

  - JUMP Jika Bit Set dan Bit tersebut di – *Clear* - kan

Sintaks : JBC Bit, rel

Operasi : Jika bit = 1, maka Bit = 0 dan PC⟵ PC + rel

OpCode		*Byte*	*Cycles*	*Encoding*
JBC	bit, rel	3	2	0001 0000 (*bit address*)(*rel.address*)

  - JUMP Jika *Carry Set*

Sintaks : JC rel

Operasi : Jika C = 1, maka PC⟵ PC + rel

OpCode		*Byte*	*Cycles*	*Encoding*
JC	rel	2	2	0010 0000 (rel.*address*)

  - JUMP Jika Bit Not Set

Sintaks : JNB Bit, rel

Operasi : Jika bit = 0, maka PC ⟵ PC + rel

OpCode		*Byte*	*Cycles*	*Encoding*
JNB	bit, rel	3	2	0011 0000 (*bit address*)(*rel.address*)

- JUMP Jika *Carry Not Set*

Sintaks : JNC rel

Operasi : Jika C = 0, maka PC ⟵ PC + rel

Anggotanya :

OpCode		*Byte*	*Cycles*	*Encoding*
JNB	rel	2	2	0101 0000 (*rel.address*)

3. Instruksi Percabangan (*Program Branching*)

- Operasi Pemanggilan *Sub Routine*
  - Absolute Call

Sintaks : ACALL addr11

Operasi : PC ⟵ PC + 2

SP ⟵ SP + 1

(SP) ⟵ PC_{7-0}

SP ⟵ SP + 1

(SP) ⟵ PC_{15-8}

PC_{10-0} ⟵ *Page Address*

OpCode		Byte	Cycles	Encoding
ACALL	addr11	2	2	a_{10} a_9 $a_8$1 0001 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0

  - Long Call

Sintaks : LCALL addr 16

Operasi : PC ⟵ PC +3

SP ⟵ SP + 1

(SP) ⟵ PC_{7-0}

SP ⟵ SP + 1

(SP) ⟵ PC_{15-8}

PC_{10-0} ⟵ Addr $_{15-0}$

OpCode	*Byte*	*Cycles*	*Encoding*

ACALL addr11 3 2 0001 0010 a_{15-8} a_{7-0}

- Operasi Kembali *Sub routine*
  - Return dari Subroutine

    Sintaks : RET

    Operasi : $PC_{15-8} \leftarrow (SP)$ $SP \leftarrow SP - 1$

    PC_{15-8} (SP) SP SP – 1

Opcode	*Byte*	*Cycle*	*Encoding*
RET	1	2	0010 0010

  - Return dari Interupt (Subroutine)

    Sintaks : RETI

    Operasi : $PC_{15-8} \leftarrow (SP)$ $SP \leftarrow SP - 1$

    PC_{15-8} (SP) SP SP – 1

Opcode	*Byte*	*Cycle*	*Encoding*
RETI	1	2	0011 0010

- Operasi JUMP tanpa Kondisi
  - *Absolute Jump*

    Sintaks : AJMP

    Operasi : PC $\leftarrow$ PC + 2

    $PC_{10-0} \leftarrow$ ~~page~~ address

Opcode	*Byte*	*Cycle*	*Encoding*
AJMP addr16	3	2	a_{10} a_9 a_8 00001 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0

  - *Long Jump*

    Sintaks : LJMP

    Operasi : PC $\leftarrow$ PC + 2

    $PC_{15-0} \leftarrow$ ~~addr~~$_{15-0}$

Opcode	*Byte*	*Cycle*	*Encoding*
LJMP addr16	3	2	0000 0001 a_{15-8} a_{7-0}

- *Short Jump*

  Sintaks : SJMP relative addr.

  Operasi : PC ← PC + 2

  PC ← PC + rel

Opcode	*Byte*	*Cycle*	*Encoding*
SJMP rel	2	2	1000 0000 (*rel address*)

- *Relative DPTR Jump*

  Sintaks : JMP @A + DPTR

  Operasi : PC ← A + DPTR

  PC ← A + DPTR

Opcode	*Byte*	*Cycle*	*Encoding*
JMP @A + DPTR	1	2	0111 0011

- Operasi *Jump* dengan Kondisi
  - Jump jika ACC sama dengan nol

    Sintaks : JZ *relative*

    Operasi : PC ← PC + 2

    jika A = 0, maka PC ← PC + rel

Opcode	*Byte*	*Cycle*	*Encoding*
JZ rel	2	2	0110 0000 (*relative address*)

  - Jump jika ACC tidak sama dengan nol

    Sintaks : JNZ relative

    Operasi : PC ← PC + 2

    jika A = 0, maka PC ← PC + rel

Opcode	*Byte*	*Cycle*	*Encoding*
JNZ rel	2	2	0111 0000 (*relative address*)

- Jump jika perbandingan dest – byte tidak sama dengan src – byte

  Sintaks : CJNE (dest-byte), (src-byte), rel

Operasi : PC ← PC + 3

jika (dest-byte) ≠ (src-byte) maka PC ← PC + rel

jika (dest-byte) = (src-byte) maka C ← 1, selain itu C ← 0

Anggotanya :

Opcode	*Byte*	*Cycle*	*Encoding*
CJNE A, direct, rel	3	2	10110101 (*directa.addr)(rel addr)*
CJNE A, # data, rel	3	2	10110100 (*immediate data)(rel addr*)
CJNE Rn, # data, rel	3	2	10111rrr (*immediate data)(rel addr*)
CJNE @Ri, #data, rel	3	2	1011011r (*immediate)(rel addr*)

- Jump jika hasil DEC byte data tidak sama dengan nol

Sintaks : DJNZ (byte), rel

Operasi : PC ← PC + 2

(byte ← Byte - 1

jika (byte) > 0, atau (byte) < 0, maka PC ← PC + rel

Anggotanya :

Opcode	*Byte*	*Cycle*	*Encoding*
DJNZ Rn, rel	2	2	1101 1rrr (*relative address*)
DJNZ *direct*, rel	3	2	1101 0101 (*direct addr) (rel. addr*)

4. Tanpa Fungsi

Sintaks : NOP

Operasi : PC ← PC + 1

Opcode	*Byte*	*Cycle*	*Encoding*
NOP	1	1	0000 0000

Referensi :

1. Myke Predko, (1995). *Programming and Custumizing The 8051 Microcontroller.* Mc Graw Hill.
2. Atmel, (2005**).** *Data Book Microcontroller.*
3. Sencer, (1997). *Programming Interfacing 8051 Microcontroller*. Mc Graw Hill.
4. Intel, (1994). *MCS'51 Microcontroller Family User Manual.*