

**PANDUAN DASAR
MIKROKONTROLER KELUARGA MCS-51**

PANDUAN DASAR MIKROKONTROLER KELUARGA MCS-51

Danny Christanto, S.T.

Kris Pusporini, S.T., M.T.

© 2004, Innovative Electronics

Hak Cipta dilindungi undang-undang

Sampul & Tata Letak: Gersom Sutedjo, S.T.

Diterbitkan pertama kali oleh:

Innovative Electronics, Surabaya 2004

Website: www.innovativeelectronics.com

Semua pertanyaan tentang produk ini dapat dikirimkan melalui e-mail ke:

tutorial51@innovativeelectronics.com

Dilarang mengutip, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

MCS-51 adalah merk dagang terdaftar dari Intel Corporation.

MS-DOS dan Windows adalah merk dagang terdaftar dari Microsoft Corporation.

ASM51 adalah merk dagang MetaLink Corporation.

DT-51 adalah merk dagang Innovative Electronics.

KATA PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Kuasa atas berkat yang telah diberikanNya dalam penyusunan Panduan Dasar Mikrokontroler Keluarga MCS-51 ini. Terima kasih juga kepada semua pihak yang telah membantu baik secara langsung maupun tidak.

Pada Panduan Dasar Mikrokontroler Keluarga MCS-51 ini, Pembaca akan diajak untuk mengenal mikrokontroler khususnya keluarga MCS-51.

Untuk hasil maksimal, Panduan Dasar Mikrokontroler Keluarga MCS-51 ini selayaknya digunakan berdampingan dengan Panduan Praktikum Dasar Mikrokontroler Keluarga MCS-51 yang mengandung berbagai contoh dan latihan pemrograman MCS-51. Semua pelajaran dasar yang ada dapat dipraktekkan dengan menggunakan modul tersebut.

Besar harapan penyusun agar para Pembaca dapat memberikan kritik dan saran mengenai isi ataupun penyusunan Panduan Dasar Mikrokontroler Keluarga MCS-51 ini.

Akhir kata, penyusun mengucapkan terima kasih kepada para Pembaca. Selamat belajar!

Surabaya, Oktober 2003

Tim Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN	xii
1. PENDAHULUAN MCS-51	1
1.1. SEKILAS TENTANG KELUARGA MCS-51	1
1.2. BEBERAPA VARIAN IC MCS-51 DAN FITURNYA.....	1
1.3. RUANG LINGKUP PANDUAN	2
2. <i>CLOCK & CPU TIMING</i>	4
2.1. <i>SISTEM CLOCK</i>	4
2.1.1. <i>XTAL DAN ON-CHIP OSCILLATOR</i>	4
2.1.2. <i>EXTERNAL CLOCK</i>	5
2.2. <i>PERHITUNGAN & TIMING DIAGRAM MACHINE CYCLES</i>	5
2.3. <i>PROGRAM COUNTER</i>	7
3. MEMORI.....	8
3.1. TIPE MEMORI.....	8
3.1.1. <i>DATA MEMORY</i>	8
3.1.2. <i>PROGRAM MEMORY</i>	8
3.2. <i>ORGANISASI INTERNAL DATA MEMORY</i>	9
3.2.1. <i>GENERAL PURPOSE RAM</i>	10
3.2.2. <i>BIT-ADDRESSABLE RAM</i>	10
3.2.3. <i>REGISTER BANKS</i>	10
3.3. <i>SPECIAL FUNCTION REGISTERS</i>	11
3.3.1. <i>PROGRAM STATUS WORD</i>	12
3.3.2. <i>ACCUMULATOR</i>	14
3.3.3. <i>REGISTER B</i>	14
3.3.4. <i>STACK POINTER</i>	14
3.3.5. <i>DATA POINTER</i>	14
3.3.6. <i>PORT REGISTERS</i>	14
3.3.7. <i>TIMER REGISTERS</i>	14
3.3.8. <i>SERIAL PORT REGISTERS</i>	14
3.3.9. <i>INTERRUPT REGISTERS</i>	14
3.3.10. <i>POWER CONTROL REGISTER</i>	15

3.4.	MENGAKSES <i>EXTERNAL MEMORY</i>	15
3.4.1.	MENGAKSES <i>EXTERNAL DATA MEMORY</i>	16
3.4.2.	MENGAKSES <i>EXTERNAL PROGRAM MEMORY</i>	17
3.4.3.	<i>ADDRESS DECODING</i>	17
3.4.4.	<i>DATA MEMORY</i> DAN <i>PROGRAM MEMORY</i> YANG SALING <i>OVERLAPPING</i>	18
4.	<i>PIN</i>	19
4.1.	MACAM-MACAM <i>PIN</i> DAN FUNGSINYA	19
4.1.1.	<i>PORT 0</i>	19
4.1.2.	<i>PORT 1</i>	19
4.1.3.	<i>PORT 2</i>	19
4.1.4.	<i>PORT 3</i>	19
4.1.5.	<i>PSEN</i>	20
4.1.6.	<i>ALE</i>	20
4.1.7.	<i>EA</i>	20
4.1.8.	<i>RST</i>	20
4.1.9.	<i>ON-CHIP OSCILLATOR INPUT</i>	20
4.1.10.	<i>POWER CONNECTION</i>	20
4.2.	STRUKTUR <i>I/O PORT</i>	21
5.	<i>INTERRUPT</i>	25
5.1.	ORGANISASI <i>INTERRUPT</i>	25
5.2.	MEMPROSES <i>INTERRUPT</i>	30
5.3.	<i>TIMER/COUNTER INTERRUPT</i>	31
5.4.	<i>SERIAL PORT INTERRUPT</i>	31
5.5.	<i>EXTERNAL INTERRUPT</i>	31
5.6.	<i>INTERRUPT TIMINGS</i>	32
6.	<i>TIMER/COUNTER</i>	33
6.1.	KONSEP <i>TIMER</i> DAN <i>COUNTER</i>	33
6.2.	<i>TIMER REGISTER</i>	33
6.3.	MODE OPERASI	35
6.3.1.	MODE 0	35
6.3.2.	MODE 1	35
6.3.3.	MODE 2	36
6.3.4.	MODE 3	36
6.4.	INISIALISASI SEBUAH <i>TIMER</i>	37
6.5.	MEMBACA SEBUAH <i>TIMER</i>	38
6.5.1.	MEMBACA <i>REGISTER</i>	38
6.5.2.	MENDETEKSI <i>OVERFLOW</i>	39
6.6.	PENGUKUR DURASI KEJADIAN	39
6.7.	PENGHITUNG KEJADIAN	40
7.	<i>SERIAL INTERFACE</i>	41

7.1.	<i>STANDART SERIAL INTERFACE</i>	41
7.2.	<i>SERIAL REGISTER</i>	41
7.3.	<i>MODE OPERASI</i>	43
7.3.1.	<i>MODE 0</i>	43
7.3.2.	<i>MODE 1</i>	43
7.3.3.	<i>MODE 2</i>	43
7.3.4.	<i>MODE 3</i>	43
7.4.	<i>BAUD RATE</i>	43
7.5.	<i>INISIALISASI SERIAL</i>	45
8.	<i>RESET</i>	47
8.1.	<i>POWER-ON RESET</i>	47
8.2.	<i>POWER SAVING MODES OF OPERATION</i>	49
8.3.	<i>IDLE MODE</i>	50
8.4.	<i>POWER DOWN MODE</i>	50
9.	<i>PROGRAM FLOW</i>	51
9.1.	<i>DIRECT JUMP</i>	51
9.2.	<i>CONDITIONAL BRANCHING</i>	51
9.3.	<i>DIRECT CALL</i>	51
9.4.	<i>RETURN FROM SUBROUTINE</i>	51
9.5.	<i>INTERRUPT</i>	51
10.	<i>ADDRESSING MODES</i>	52
10.1.	<i>REGISTER ADDRESSING</i>	52
10.2.	<i>DIRECT ADDRESSING</i>	52
10.3.	<i>INDIRECT ADDRESSING</i>	52
10.4.	<i>IMMEDIATE ADDRESSING</i>	52
10.5.	<i>RELATIVE ADDRESSING</i>	52
10.6.	<i>ABSOLUTE ADDRESSING</i>	53
10.7.	<i>LONG ADDRESSING</i>	53
10.8.	<i>INDEXED ADDRESSING</i>	53
11.	<i>INSTRUKSI</i>	55
11.1.	<i>ARITHMETIC INSTRUCTION</i>	55
11.2.	<i>LOGICAL INSTRUCTION</i>	55
11.3.	<i>BOOLEAN INSTRUCTION</i>	56
11.4.	<i>JUMP INSTRUCTION</i>	57
11.5.	<i>DATA TRANSFER INSTRUCTION</i>	57
12.	<i>PROSES DISAIN MINIMUM SYSTEM</i>	59
12.1.	<i>PEMILIHAN MIKROKONTROLER</i>	59
12.2.	<i>EXTERNAL ACCESS</i>	60
12.3.	<i>ADDRESS DECODING & MEMORY MAPPING</i>	61
12.4.	<i>INTERFACING</i>	63

13.	PROSES PENGEMBANGAN PROGRAM	65
13.1.	<i>DEVELOPMENT CYCLE</i>	65
13.2.	ASM51	66
13.2.1.	<i>ASSEMBLER CONTROL</i>	67
13.2.2.	SIMBOL DAN LABEL.....	67
13.2.3.	DEFINISI SIMBOL.....	68
13.2.4.	<i>SEGMENT</i> DAN LABEL.....	69
13.2.5.	<i>MNEMONIC</i> DAN <i>OPERAND</i>	70
13.2.6.	<i>COMMENT</i>	70
13.2.7.	<i>END</i>	70
	PENUTUP	71
	DAFTAR PUSTAKA	72
	LAMPIRAN	73

DAFTAR GAMBAR

Gambar 1:	Rangkaian XTAL dengan <i>On-Chip Oscillator</i>	4
Gambar 2:	MCS-51 dengan <i>External Clock Signal</i>	5
Gambar 3:	<i>Timing Diagram</i> Instruksi 1-Cycle	6
Gambar 4:	<i>Timing Diagram</i> Instruksi 2-Cycle	7
Gambar 5:	Pembagian Ruang <i>Internal Data Memory</i>	8
Gambar 6:	Konfigurasi <i>Internal</i> dan <i>External Data Memory</i>	8
Gambar 7:	Pembagian Ruang <i>Internal Program Memory</i>	9
Gambar 8:	Konfigurasi <i>Internal</i> dan <i>External Program Memory</i>	9
Gambar 9:	Organisasi <i>Internal Data Memory</i>	10
Gambar 10:	Pembagian <i>Register Banks</i>	11
Gambar 11:	Pembagian SFR.....	12
Gambar 12:	Alokasi <i>Bit PSW</i>	12
Gambar 13:	Alokasi <i>Bit PCON</i>	15
Gambar 14:	Penggunaan <i>Port</i> pada Teknik <i>Multiplex</i>	16
Gambar 15:	Koneksi MCS-51 dengan <i>External Data Memory</i>	16
Gambar 16:	Koneksi MCS-51 dengan <i>External Program Memory</i>	17
Gambar 17:	<i>Address Decoding</i> 8 KB	18
Gambar 18:	Rangkaian <i>Overlapping</i>	18
Gambar 19:	Rangkaian <i>Internal Port 0</i>	21
Gambar 20:	Rangkaian <i>Internal Port 1</i>	22
Gambar 21:	Rangkaian <i>Internal Port 2</i>	22
Gambar 22:	Rangkaian <i>Internal Port 3</i>	23

Gambar 23: Alokasi <i>Bit</i> IE	25
Gambar 24: Alokasi <i>Bit</i> IP	26
Gambar 25: Alokasi <i>Bit</i> TCON	27
Gambar 26: Alokasi <i>Bit</i> SCON	28
Gambar 27: Ringkasan Struktur <i>Interrupt</i>	29
Gambar 28: <i>Timing Diagram</i> Respon <i>Interrupt</i>	32
Gambar 29: Alokasi <i>Bit</i> TMOD	33
Gambar 30: Alokasi <i>Bit</i> TCON	34
Gambar 31: <i>Timer/Counter</i> Mode 0	35
Gambar 32: <i>Timer/Counter</i> Mode 1	36
Gambar 33: <i>Timer/Counter</i> Mode 2	36
Gambar 34: <i>Timer/Counter</i> Mode 3	37
Gambar 35: Alokasi <i>Bit</i> SCON	41
Gambar 36: Rangkaian <i>Power-On Reset</i>	47
Gambar 37: Rangkaian <i>Manual Reset</i>	48
Gambar 38: Beberapa Macam Kondisi <i>Reset</i>	49
Gambar 39: Rangkaian Penghemat Daya	49
Gambar 40: Rangkaian Dasar Mikrokontroler	59
Gambar 41: Rangkaian <i>External Access</i>	60
Gambar 42: <i>Address Decoding</i> 8 Bagian	61
Gambar 43: Penambahan EEPROM 28C64B dan PPI 82C55	62
Gambar 44: RS-232 – TTL <i>Converter</i>	63

Gambar 45: <i>Interface I/O</i>	64
Gambar 46: <i>Flowchart Pengembangan Program</i>	65

DAFTAR TABEL

Tabel 1:	<i>Program Status Word</i>	13
Tabel 2:	Pemilihan <i>Register Bank</i>	13
Tabel 3:	<i>Power Control Register</i>	15
Tabel 4:	Fungsi Lain dari <i>Port 3</i>	19
Tabel 5:	<i>Interrupt Enable</i>	25
Tabel 6:	<i>Interrupt Priority</i>	26
Tabel 7:	<i>Timer Control</i>	27
Tabel 8:	<i>Serial Port Control</i>	28
Tabel 9:	<i>Interrupt Vector Address</i>	29
Tabel 10:	<i>Timer Mode</i>	33
Tabel 11:	Mode Operasi <i>Timer/Counter</i>	34
Tabel 12:	<i>Timer Control</i>	34
Tabel 13:	<i>Serial Port Control</i>	41
Tabel 14:	Mode Komunikasi Serial.....	42
Tabel 15:	Mode Komunikasi Multiprosesor	42
Tabel 16:	Nilai dan Konfigurasi <i>Baud Rate</i>	45
Tabel 17:	Daftar Instruksi Aritmetik	55
Tabel 18:	Daftar Instruksi Logika	55
Tabel 19:	Daftar Instruksi <i>Boolean</i>	56
Tabel 20:	Daftar Instruksi Percabangan	57
Tabel 21:	Daftar Instruksi Pemindahan Data	57
Tabel 22:	<i>Memory Mapping</i> Gambar 43	62

DAFTAR LAMPIRAN

Lampiran 1: Legenda	73
Lampiran 2: Daftar Instruksi MCS-51	74
Lampiran 3: Kondisi SFR Setelah <i>Reset</i>	123

7. SERIAL INTERFACE

MCS-51 memiliki kemampuan untuk berkomunikasi secara serial melalui *pin* RXD dan TXD. Satu hal yang perlu diingat adalah tingkat tegangan komunikasi kedua *pin* serial menggunakan tingkat tegangan TTL.

7.1. STANDART SERIAL INTERFACE

Pada prinsipnya, komunikasi serial adalah komunikasi dimana transmisi data dilakukan per *bit*. *Interface* serial hanya membutuhkan jalur yang sedikit (umumnya hanya 2 jalur) sehingga lebih menghemat *pin* jika dibandingkan dengan *interface* paralel.

Komunikasi serial ada dua macam, *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan *clock* dan mengirimkan *clock* tersebut bersama-sama dengan data. Contoh penggunaan *synchronous serial* terdapat pada transmisi data *keyboard*.

Asynchronous serial adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan *clock* namun hanya data yang ditransmisikan, tanpa *clock*. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi *clock* harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi *clock* pengirim dan penerima akan membaca data sesuai dengan frekuensi *clock* penerima. Contoh penggunaan *asynchronous serial* adalah pada *Universal Asynchronous Receiver Transmitter* (UART) yang digunakan pada *serial port* (COM) komputer.

MCS-51 mendukung komunikasi secara asinkron, bahkan tiga dari empat *serial mode* yang dimiliki MCS-51 kompatibel dengan UART.

7.2. SERIAL REGISTER

Register yang digunakan untuk mengatur komunikasi serial terdapat pada *Serial Control* (SCON).

MSB						LSB	
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Gambar 35. Alokasi *Bit* SCON

Tabel 13. *Serial Port Control*

<i>Bit</i>	<i>Alamat Bit</i>	<i>Simbol</i>	<i>Deskripsi</i>
SCON.7	9FH	SM0	Pemilih Mode Komunikasi Serial
SCON.6	9EH	SM1	Pemilih Mode Komunikasi Serial
SCON.5	9DH	SM2	Pemilih Mode Komunikasi Multiprosesor
SCON.4	9CH	REN	<i>Reception Enable</i>
SCON.3	9BH	TB8	<i>Bit ke-9 yang Dikirim</i>

Tabel 13. *Serial Port Control* (sambungan)

Bit	Alamat Bit	Simbol	Deskripsi
SCON.2	9AH	RB8	<i>Bit ke-9 yang Diterima</i>
SCON.1	99H	TI	<i>Transmit Interrupt Flag</i>
SCON.0	98H	RI	<i>Receive Interrupt Flag</i>

Berikut ini adalah penjelasan masing-masing *bit* SCON yang berkaitan dengan *serial port*:

- SM0 & SM1
Pemilih mode komunikasi serial.

Tabel 14. Mode Komunikasi Serial

SM0	SM1	Mode	Deskripsi	Baud Rate
0	0	0	<i>8-bit Shift Register</i>	Frek. Osilator/12
0	1	1	<i>8-bit UART</i>	Variabel
1	0	2	<i>9-bit UART</i>	Frek Osilator/64
1	1	3	<i>9-bit UART</i>	Variabel

Baud rate pada mode 1, 2, dan 3 dapat dilipatgandakan dengan memberi nilai ‘1’ pada SMOD (dalam SFR PCON). *Baud rate* variabel adalah *baud rate* yang dihasilkan oleh *Timer 1* (*baud rate* akan dibahas pada bagian 7.4).

- SM2
Jika SM2 bernilai ‘1’ maka komunikasi multiprosesor diaktifkan dengan kondisi terdapat pada tabel 15.

Tabel 15. Mode Komunikasi Multiprosesor

Mode	Jika SM2 = 1
2 atau 3	RI tidak akan diaktifkan jika <i>bit ke-9</i> yang diterima bernilai ‘0’
1	RI tidak akan diaktifkan jika <i>stop bit</i> yang <i>valid</i> (bernilai ‘1’) tidak diterima

Pada mode 0, nilai SM2 harus ‘0’.

- REN
REN harus diberi nilai ‘1’ untuk mengaktifkan penerimaan data. Jika REN diberi nilai ‘0’, maka tidak akan ada penerimaan data.
- TB8
TB8 adalah *bit ke-9* yang dikirimkan dalam mode 2 atau 3. Nilai *bit* ini diatur oleh program *user*.
- RB8
RB8 adalah *bit ke-9* yang diterima dalam mode 2 atau 3. Pada mode 1, RB8 adalah *stop bit* yang diterima. Pada mode 0, RB8 tidak digunakan.

7.3. MODE OPERASI

MCS-51 memiliki 4 mode komunikasi serial. Mode 0 berupa *synchronous serial (shift register)*, sedangkan tiga mode yang lain berupa *asynchronous serial (UART)*. Pada semua mode, pengiriman dilakukan jika ada instruksi yang mengisi nilai *register SBUF*. Sedangkan pada saat penerimaan, data yang diterima akan disimpan pada *register SBUF*.

7.3.1. MODE 0

Mode 0 adalah *8-bit shift register* dimana data dikirimkan dan diterima melalui *pin RXD* sedangkan *clock* dikirimkan dan diterima melalui *pin TXD*. Pengiriman data *8 bit* dilakukan dengan mengirimkan *Least Significant Bit (LSB)* terlebih dahulu.

Pada mode 0, *baud rate* yang digunakan adalah sebesar $1/12$ dari frekuensi osilator.

7.3.2. MODE 1

Pada mode 1, jumlah data yang dikirimkan sebanyak *10 bit* yang terdiri dari *start bit*, *8 bit* data (LSB terlebih dahulu), dan *stop bit*. Pada proses penerimaan, nilai *stop bit* akan dimasukkan ke RB8 secara otomatis. Pada proses pengiriman, *stop bit* akan diberi nilai '1' secara otomatis.

Pada mode 1, *baud rate* yang digunakan dapat diatur melalui *Timer 1*.

7.3.3. MODE 2

Pada mode 2, jumlah data yang dikirimkan sebanyak *11 bit* yang terdiri dari *start bit*, *8 bit* data (LSB terlebih dahulu), *bit ke-9*, dan *stop bit*. Pada proses pengiriman, nilai *bit ke-9* dapat diatur dengan mengisi nilai TB8. Pada proses penerimaan, *bit ke-9* akan dimasukkan ke RB8 secara otomatis.

Pada mode 2, *baud rate* yang dapat digunakan adalah sebesar $1/64$ frekuensi osilator atau $1/32$ frekuensi osilator jika SMOD bernilai '1'.

7.3.4. MODE 3

Mode 3 hampir sama dengan mode 2. Perbedaannya terdapat pada *baud rate* yang digunakan. Jika mode 2 menggunakan *baud rate* yang pasti, mode 3 menggunakan *baud rate* yang dihasilkan oleh *Timer 1*.

7.4. BAUD RATE

Baud rate adalah frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data. Satuan *baud rate* pada umumnya adalah bps (*bit per second*), yaitu jumlah *bit* yang dapat ditransmisikan per detik.

Baud rate untuk mode 0 bernilai tetap dengan rumus yang terdapat pada persamaan 1.

$$\text{Baud Rate} = \frac{\text{Frekuensi Osilator}}{12} \quad (1)$$

Sedangkan *baud rate* untuk mode 2 memiliki 2 variasi tergantung dari kondisi SMOD. Rumus *baud rate* untuk mode 2 terdapat pada persamaan 2.

$$\text{Baud Rate} = \frac{2^{\text{SMOD}}}{64} \times \text{Frekuensi Osilator} \quad (2)$$

Baud rate untuk mode 1 dan 3 dihasilkan oleh *Timer 1*. Pengaturan *baud rate* untuk mode 1 dan 3 dapat dilakukan dengan cara mengubah nilai SMOD, TMOD, dan TH1. Nilai *baud rate* dapat diperoleh dengan menggunakan persamaan 3.

$$\text{Baud Rate} = \frac{2^{\text{SMOD}}}{32} \times \text{Timer 1 Overflow Rate} \quad (3)$$

Umumnya *Timer 1* dioperasikan pada mode 2 (*8-bit Auto Reload*) sehingga didapat persamaan 4.

$$\text{Baud Rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Frekuensi Osilator}}{12 \times (256 - \text{TH1})} \quad (4)$$

Berdasarkan persamaan 4, *user* dapat menghitung berapa nilai TH1 yang dibutuhkan jika diketahui *baud rate* yang diinginkan dengan persamaan 5.

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}} \times \text{Frekuensi Osilator}}{384 \times \text{Baud Rate}} \quad (5)$$

Satu hal yang harus diperhatikan dalam pengaturan *baud rate* adalah nilai *baud rate* dan nilai TH1 diusahakan harus tepat dan bukan merupakan pembulatan. Untuk komunikasi serial kecepatan tinggi, pembulatan terhadap nilai-nilai tersebut dapat mengakibatkan kekacauan dalam proses pengiriman atau penerimaan. Jika terdapat nilai pecahan, *user* disarankan untuk mengganti osilator dengan frekuensi yang sesuai. Untuk komunikasi dengan kecepatan rendah, toleransi terhadap kesalahan cukup besar sehingga pembulatan masih boleh dilakukan.

Misalkan *baud rate* yang diinginkan adalah 19200 bps dengan frekuensi osilator 11,0592 MHz. Dengan memasukkan data ini ke dalam persamaan 5 maka akan didapat persamaan 6.

$$\text{TH1} = 256 - (2^{\text{SMOD}} \times 1,5) \quad (6)$$

Jika 2^{SMOD} bernilai '1', maka akan didapat TH1 sebesar 254,5. Untuk menghindari TH1 berupa pecahan, 2^{SMOD} harus bernilai '2' (SMOD bernilai '1') sehingga didapat TH1 sebesar 253 atau FDh.

Untuk mendapatkan *baud rate* yang lambat, *user* dapat mengoperasikan *Timer 1* pada mode 1 dengan rumus pada persamaan 7.

$$\text{Baud Rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Frekuensi Osilator}}{12 \times (65536 - \text{Timer 1})} \quad (7)$$

Beberapa konfigurasi *baud rate* yang umum digunakan terdapat dalam tabel 16.

Tabel 16. Nilai dan Konfigurasi *Baud Rate*

Serial		Timer 1				
Mode	Baud Rate	Frekuensi Osilator	SMOD	C/T	Mode	Reload
0	1,6667 Mbps (max.)	20 MHz	X	X	X	X
2	625 Kbps (max.)	20 MHz	1	X	X	X
1, 3	104,1667 Kbps (max.)	20 MHz	1	0	2	FFh
1, 3	19,2 Kbps	11,0592 MHz	1	0	2	FDh
1, 3	9,6 Kbps	11,0592 MHz	0	0	2	FDh
1, 3	4,8 Kbps	11,0592 MHz	0	0	2	FAh
1, 3	2,4 Kbps	11,0592 MHz	0	0	2	F4h
1, 3	1,2 Kbps	11,0592 MHz	0	0	2	E8h
1, 3	137,5 bps	11,9856 MHz	0	0	2	1Dh
1, 3	110 bps	6 MHz	0	0	2	72h
1, 3	110 bps	12 MHz	0	0	1	FEEBh

7.5. INISIALISASI SERIAL

Proses inisialisasi bertujuan untuk menentukan mode komunikasi serial dan *baud rate* yang digunakan.

Register yang harus diatur terlebih dahulu meliputi:

1. SCON

Langkah pertama adalah menentukan mode yang akan digunakan (mode 0, 1, 2, atau 3), kemampuan menerima data, dan nilai *bit* ke-9.

Misalnya mode yang digunakan adalah mode 1 dengan kemampuan menerima data namun tanpa komunikasi multiprosesor, maka instruksinya adalah sebagai berikut:

```
MOV    SCON, #01010000b
atau
MOV    SCON, #50h
atau
SETB   SM1
SETB   REN
```

2. TMOD, TH1 dan/atau TL1, PCON, dan TCON

Jika komunikasi serial digunakan dalam mode 1 atau 3, maka langkah berikutnya adalah menentukan *baud rate*.

Misalnya *Timer/Counter* 1 digunakan sebagai *timer* dalam mode 2 untuk membangkitkan *baud rate* 19200 bps, maka instruksinya adalah sebagai berikut:

```
MOV    TMOD, #00100000b
MOV    TH1, #0FDh
MOV    PCON, #10000000b
MOV    TCON, #01000000b
atau
MOV    TMOD, #20h
MOV    TH1, #0FDh
MOV    PCON, #80h
MOV    TCON, #40h
atau
MOV    TMOD, #20h
MOV    TH1, #0FDh
MOV    PCON, #80h
SETB  TR1
```

3. IE dan/atau IP

Jika komunikasi serial yang diprogram akan digunakan sebagai sumber *interrupt*, maka IE dan/atau IP juga harus diatur.

Misalnya komunikasi serial digunakan sebagai sumber *interrupt* dengan prioritas tinggi, maka instruksinya adalah sebagai berikut:

```
MOV    IP, #00010000b
MOV    IE, #10010000b
atau
MOV    IP, #10h
MOV    IE, #90h
atau
SETB  PS
SETB  ES
SETB  EA
```