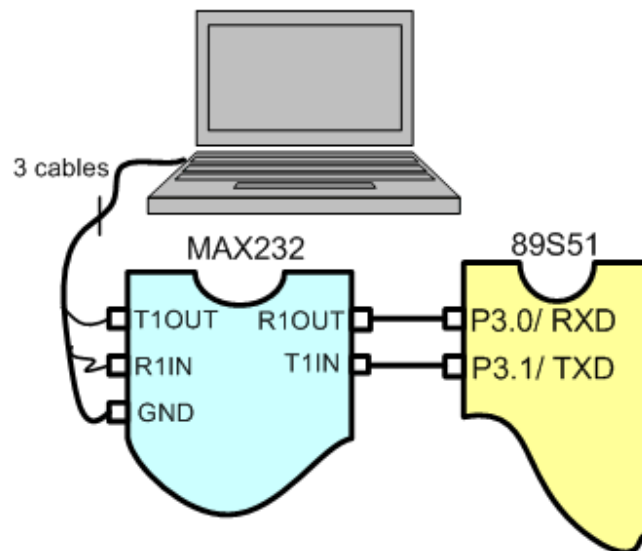


1.6. Port Serial

Umumnya orang selalu menganggap port seri pada MCS51 adalah UART yang bekerja secara asinkron, jarang yang menyadari port seri tersebut bisa pula bekerja secara sinkron, pada hal sebagai port seri yang bekerja secara sinkron merupakan sarana yang baik sekali untuk menambah input/output bagi mikrokontroler.



Gambar 1.17. Komunikasi serial dengan komputer

Dikenal 2 macam cara transmisi data secara seri. Kedua cara tersebut dibedakan oleh sinyal denyut (clock) yang dipakai untuk men-‘dorong’ data seri, kalau clock dikirim bersama dengan data seri, cara tersebut dikatakan sebagai transmisi data seri secara sinkron. Sedangkan dalam transmisi data seri secara asinkron, clock tidak dikirim bersama data seri, rangkaian penerima data harus membangkitkan sendiri clock pendorong data seri.

Port seri MCS51 bisa dipakai dalam 4 mode kerja yang berbeda. Dari 4 mode tersebut, 1 mode diantaranya bekerja secara sinkron dan 3 lainnya bekerja secara asinkron. Secara ringkas ke-empat mode kerja tersebut bisa dibedakan sebagai berikut:

Mode 0

Mode ini bekerja secara sinkron, data seri dikirim dan diterima melalui kaki P3.0 (RxD), dan kaki P3.1 (TxD) dipakai untuk menyalurkan clock pendorong data seri yang dibangkitkan MCS51.

Data dikirim/diterima 8 bit sekali gus, dimulai dari bit yang bobotnya paling kecil (bit 0) dan diakhiri dengan bit yang bobotnya paling besar (bit 7). Kecepatan pengiriman data (baud rate) adalah 1/12 frekuensi osilator kristal.

Mode 1

Mode ini dan mode-mode berikutnya bekerja secara asinkron, data dikirim melalui kaki P3.1 (TxD) dan diterima melalui kaki P3.0 (RxD).

Pada Mode 1 data dikirim/diterima 10 bit sekali gus, diawali dengan 1 bit start, disusul dengan 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), diakhiri dengan 1 bit stop. Pada MCS51 yang berfungsi sebagai penerima bit stop ditampung pada RB8 dalam register SCON. Kecepatan pengiriman data (baud rate) bisa diatur sesuai dengan keperluan.

Mode inilah yang umum dikenal sebagai UART (Universal Asynchronous Receiver/Transmitter).

Mode 2

Data dikirim/diterima 11 bit sekali gus, diawali dengan 1 bit start, disusul 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), kemudian bit ke 9 yang bisa diatur lebih lanjut, diakhiri dengan 1 bit stop.

Pada MCS51 yang berfungsi sebagai pengirim, bit 9 tersebut berasal dari bit TB8 dalam register SCON. Pada MCS52 yang berfungsi sebagai penerima, bit 9 ditampung pada bit RB8 dalam register SCON, sedangkan bit stop diabaikan tidak ditampung. Kecepatan pengiriman data (baud rate) bisa dipilih antara 1/32 atau 1/64 frekuensi osilator kristal.

Mode 3 Mode ini sama dengan Mode 2, hanya saja kecepatan pengiriman data (baud rate) bisa diatur sesuai dengan keperluan, seperti halnya Mode 1.

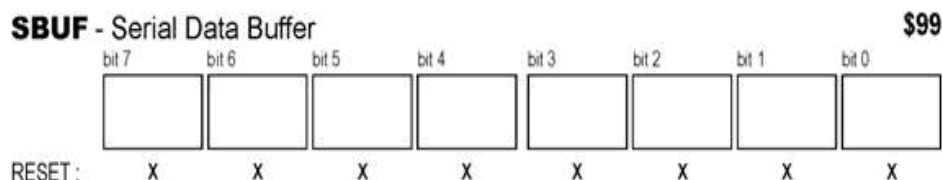
Pada mode asinkron (Mode 1, Mode 2 dan Mode 3), port seri MCS51 bekerja secara full duplex, artinya pada saat yang sama port seri ini bisa mengirim data sekali gus menerima data.

Register SBUF merupakan register penghubung port seri. Dalam ke-empat mode di atas, semua instruksi yang mengakibatkan perubahan isi SBUF akan mengakibatkan port seri mengirimkan data keluar dari MCS51. Agar port seri bisa menerima data, bit REN dalam register SCON harus bernilai '1'. Pada mode 0, proses penerimaan data dimulai dengan instruksi CLR RI, sedangkan dalam mode lainnya proses penerimaan data diawali oleh bit start yang bernilai '0'. Data yang diterima port seri dari luar MCS51, diambil dengan instruksi MOV A,SBUF.

Mengambil data dari SBUF dan menyimpan data ke SBUF sesungguhnya bekerja pada dua register yang berlainan, meskipun nama registernya sama-sama SBUF.

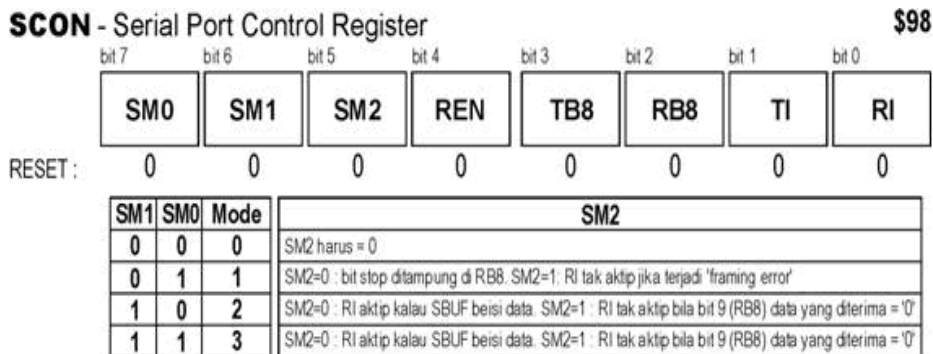
1.6.1 Register-register Port Seri MCS51

MCS51 dilengkapi dengan 2 register dan beberapa bit tambahan untuk keperluan pemakai port seri.



SBUF merupakan SFR (Special Function Register) yang terletak pada memori-data internal dengan nomor \$99. SBUF mempunyai kegunaan ganda, data yang disimpan pada

SBUF akan dikirim keluar MCS51 lewat port seri, sedangkan data dari luar MCS51 yang diterima port seri diambil dari SBUF pula. Jadi meskipun hanya menempati satu nomor memori-data internal (nomor \$99), sesungguhnya SBUF terdiri dari 2 register yang berbeda.



SCON merupakan SFR (Special Function Register) yang terletak pada memori-data internal dengan nomor \$98, merupakan register utama untuk mengatur kerja port seri MCS51. Setelah reset semua bit dalam SCON bernilai '0'.

1. Bit SM0 dan bit SM1 (bit 7 dan bit 6 pada register SMOD) dipakai untuk menentukan mode kerja port seri. Setelah reset kedua bit ini bernilai '0'
2. Bit REN (bit 4) dipakai untuk mengaktifkan kemampuan port seri menerima data. Pada mode 0 kaki RxD (kaki P3.0) dipakai untuk mengirim data seri (REN='0') dan juga untuk menerima data seri (REN='1'). Sifat ini terbawa pula pada saat port seri bekerja pada mode 1, 2 dan 3, meskipun pada mode-mode tersebut kaki RxD hanya dipakai untuk mengirim data, agar kaki RxD bisa dipakai untuk menerima data terlebih dulu harus dibuat REN='1'. Setelah reset bit REN bernilai '0'.

3. Pada mode kerja 2 dan mode kerja 3, port seri bekerja dengan 9 bit data, SBUF yang kapasitasnya 8 bit tidak cukup untuk keperluan ini. Bit ke-sembilan yang akan dikirim terlebih dulu diletakkan di TB8 (bit 3), sedangkan bit RB8 (bit 2) merupakan bit yang dipakai untuk menampung bit ke-sembilan yang diterima port seri.

4. Pada mode kerja 1, RB8 dipakai untuk menampung bit stop yang diterima, dengan demikian apa bila RB8 bernilai '1' maka data diterima dengan benar, sebaliknya apa bila RB8='0' berarti terjadi kesalahan kerangka (framing error).

Kalau bit SM2 (bit 5) bernilai '1', jika terjadi kesalahan kerangka, RI tidak akan menjadi '1' meskipun SBUF sudah berisi data dari port seri.

Bit ke 9 ini bisa dipakai sebagai bit pariti, hanya saja bit pariti yang dikirim harus ditentukan sendiri dengan program dan diletakkan pada TB8, dan bit pariti yang diterima pada RB8 dipakai untuk menentukan integritas data secara program pula. Tidak seperti dalam UART standard, semuanya itu dikerjakan oleh perangkat keras dalam IC UART.

5. Bit TI (bit 1) merupakan petanda yang setara dengan petanda TDRE (Transmitter Data Register Empty) yang umum dijumpai pada UART standard. Setelah port seri selesai mengirim data yang disimpan ke-dalam SBUF, bit TI akan bernilai '1' dengan sendirinya, bit ini harus di-nol-kan dengan program agar bisa dipakai untuk memantau keadaan SBUF dalam pengiriman data berikutnya.

Sub-rutin SerialOut berikut dipakai untuk mengirim data seri, bisa dipakai untuk semua mode port seri. Baris 02 menunggu TI menjadi '1', dimaksud untuk memastikan

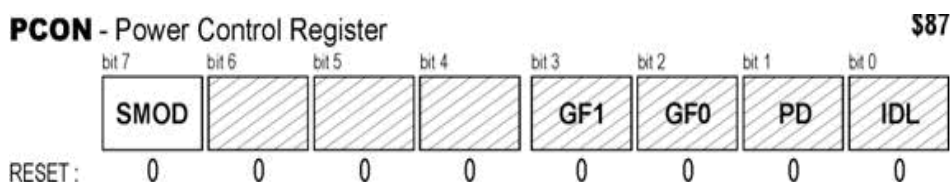
pengiriman data sebelumnya sudah selesai. Data yang akan dikirim sebelumnya sudah disimpan di A, pada baris 03 data tersebut dikirim melalui port seri dengan cara meletakkannya di SBUF. Agar TI bisa dipakai untuk memantau keadaan SBUF pada pengiriman data berikutnya, pada baris 04 TI di-nol-kan.

```
01: SerialOut:
02: JNB TI,$ ; tunggu data sebelumnya selesai dikirim
03: MOV SBUF,A ; kirim data baru
04: CLR TI ; petanda ada pengiriman baru
05: RET
```

6. Bit RI (bit 0) merupakan petanda yang setara dengan petanda RDRF (Receiver Data Register Full) yang umum dijumpai pada UART standard. Setelah SBUF menerima data dari port seri, bit RI akan bernilai '1' dengan sendirinya, bit ini harus di-nol-kan dengan program agar bisa dipakai untuk memantau keadaan SBUF dalam penerimaan data berikutnya.

Sub-rutin SerialIn berikut dipakai untuk menerima data seri, bisa dipakai untuk semua mode port seri. Baris 02 menunggu RI menjadi '1', dimaksud untuk memastikan sudah ada data baru yang diterima pada SBUF. Pada baris 03 data pada SBUF diambil ke A. Agar RI bisa dipakai untuk memantau keadaan SBUF pada pengiriman data berikutnya, pada baris 04 RI di-nol-kan.

```
01: SerialIn:
02: JNB RI,$ ; tunggu SBUF berisi data baru
03: MOV A,SBUF ; ambil data
04: CLR RI ; pentanda data sudah diambil
05: RET
```



Mode 0 port serial

UART merupakan standard yang dipakai untuk komunikasi data seri dengan komputer, komunikasi data seri dengan modem dan lain sebagainya.

Komunikasi data seri secara sinkron seperti mode 0, merupakan komunikasi data seri yang banyak dipakai untuk menghubungkan IC-IC digital dalam sebuah sistem, misalnya pada IC Serial EEPROM, cara ini belakangan menjadi makin populer karena rangkaiannya sederhana dan tidak makan tempat.

Dalam dunia digital, dikenal 3 macam teknik transmisi data seri secara sinkron untuk keperluan di atas, yang paling terkenal adalah teknik ciptaan Philips yang dinamakan sebagai I2C (Inter IC Communication), Motorola mengenalkan teknik yang dinamakan sebagai SPI (Serial Peripheral Interface) dan National Semiconductor menciptakan MicroWire.

Transmisi data seri yang dipakai pada mode 0, tidak sepadan dengan 3 teknik yang disebut di atas, tapi dengan perancangan yang cermat mode 0 ini bisa dihubungkan ke SPI, sehingga bisa dipakai untuk menghubungkan MCS51 dengan mikrokontroler Motorola MC68HC11.

Sinyal data seri sinkron yang ada pada kaki P3.0 dan P3.1, sesungguhnya murni merupakan sinyal yang biasa dipakai untuk mengendalikan shift-register, dengan demikian dengan menghubungkan shift register ke port seri, bisa menambah port input maupun port output dengan mudah.

Baud Rate

Baud rate pada mode 0 adalah tertentu: pada mode 0, Baud Rate = $1/12 \times$ Frekuensi Osilator. Baud rate pada mode 2 bergantung pada nilai bit SMOD pada SFR PCON. Jika SMOD = 0, baud rate adalah $1/64$ frekuensi osilator. Jika SMOD=1, baud rate adalah $1/32$ frekuensi osilator. Penentuan baud rate mode 2 adalah sebagai berikut:

$$BaudRate = \frac{2^{SMOD}}{64} \times (Oscillator \ Frequency)$$

Sedangkan baud rate pada mode 1 dan 3 ditentukan oleh nilai laju overflow dari Timer 1.

Menggunakan Timer 1 untuk membangkitkan Baud Rate

Ketika timer 1 digunakan untuk membangkitkan clock baud rate, baud rate pada mode 1 dan 3 adalah ditentukan oleh laju overflow timer 1 dan nilai dari SMOD. Penentuan baud rate untuk mode 1 dan 3 adalah sebagai berikut:

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{Oscillator \ Frequency}{12 \times [256 - (TH \ 1)]}$$

Interupsi timer 1 harus disable pada aplikasi ini. Pada kebanyakan aplikasi, timer ini dioperasikan sebagai timer, dengan mode auto reload mode 2. Pada kasus ini baud rate diberikan dengan rumus sebagai berikut:

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{Oscillator \ Frequency}{12 \times [256 - (TH \ 1)]}$$