

BAB 4

UNIT INPUT OUTPUT SISTEM MIKROPROSESOR

Unit Input Output atau yang lebih dikenal dengan I/O merupakan bagian dari sistem mikroprosesor yang bekerja menjembatani mikroprosesor dengan peralatan I/O seperti keyboard, printer, monitor, mouse, dan sebagainya. I/O umumnya bersifat dapat diprogram (*programmable*).

Kata kunci: I/O, *progammable*

1. Unit Input Output

Selain CPU dan Unit Memori, Unit Input Output (I/O) merupakan komponen pokok dalam sebuah sistem mikroprosesor. Sistem mikroprosesor memerlukan unit I/O untuk menyajikan proses dan hasil pengolahan CPU. Unit I/O bekerja sebagai penghubung antara CPU dengan alat-alat input seperti keypad, keyboard, mouse dan juga sebagai penghubung dengan alat output seperti monitor, printer dan sebagainya. Unit I/O juga bekerja sebagai saluran komunikasi antara CPU dengan sistem diluar. Biasanya untuk berkomunikasi keluar unit I/O yang digunakan dalam sistem mikroprosesor adalah jenis I/O serial.

Monitor sebagai salah satu alat output digunakan untuk menyajikan proses dan hasil pengolahan sebuah komputer. Monitor dalam sistem mikroprosesor semula berfungsi hanya sebagai umpan balik bagi user. Tetapi sekarang telah berkembang dimana lewat monitor sebuah komputer

bekerja menerima masukan karakter. Sebenarnya jika keseluruhan proses dan hasil pengolahan CPU telah diyakini kebenarannya bisa saja sebuah komputer dioperasikan tanpa monitor. Akan tetapi karena proses kerja komputer saat ini telah berkembang dalam ukuran data yang besar maka sangat sulit bekerja dengan komputer tanpa monitor. Keberadaan monitor dalam sistem mikroprosesor menjadi mutlak diperlukan. Monitor sebuah komputer dapat berupa sebuah indikator lampu LED, seven segment, LCD, atau monitor tabung CRT.

Untuk mencetak hasil pengolahan data dalam sebuah komputer diperlukan printer. Printer bekerja menghasilkan cetakan berupa *hardcopy*. Peralatan output sejenis lainnya yang serupa dengan printer adalah plotter. Plotter digunakan untuk mencetak gambar dengan ukuran yang lebih besar dan dengan berbagai jenis warna. Saat ini peralatan cetak berbasis komputer sudah semakin canggih disediakan di pusat-pusat photo copy.

Informasi di luar CPU harus diakses sebagai data masukan. Seperti entri input dari keyboard, mouse, atau beberapa jenis alat input seperti kamera dan sebagainya. Setiap tombol yang ditekan pada keyboard harus ada mekanisme membaca dan memorikan kode-kode tombol. Demikian juga dengan berbagai jenis input yang lain seperti mouse, scanner, atau dari berbagai jenis sensor yang digunakan dalam sistem kendali. Begitu diproses informasi harus disajikan ke monitor sehingga pengguna komputer mendapat kejelasan baik secara visual atau audio terhadap interaksi kerja yang sedang dilaksanakan. Disinilah peran monitor dan sound diperlukan.

Perkembangan mikroelektronika telah mendukung perkembangan I/O dari *unprogrammable* ke *programmable* sistem. I/O semula bekerja tetap dengan perangkat *hardware* semata tanpa program (*unprogrammable*). I/O semacam ini fungsinya menjadi terbatas dan tidak fleksibel. *Programmable* I/O dikembangkan untuk memberi jawaban kelemahan-kelemahan *unprogrammable* I/O. Dengan *programmable* I/O sistem mikroprosesor menjadi sangat fleksibel. Sebelum ada program sebuah *programmable* I/O belum membentuk format fungsi port. Apakah sebuah port sebagai input atau output. Fungsi port terbentuk setelah ada inialisasi. Inialisasi adalah proses dimana sebuah program kecil bekerja membentuk

fungsi port apakah sebagai masukan, keluaran, atau dua arah. Inialisasi dilakukan diawal pemakaian melalui program inialisasi. Setelah program inialisasi bekerja baru I/O memiliki format fungsi khusus. Port I/O bekerja menjembatani CPU dengan alat input dan alat output.

Beberapa komponen I/O terprogram yang sangat populer dalam dunia sistem mikroprosesor adalah Z-80 PIO dan PPI 8255. Berdasarkan pola aliran data pada I/O dapat digolongkan menjadi dua yaitu :

- I/O Paralel
- I/O Serial

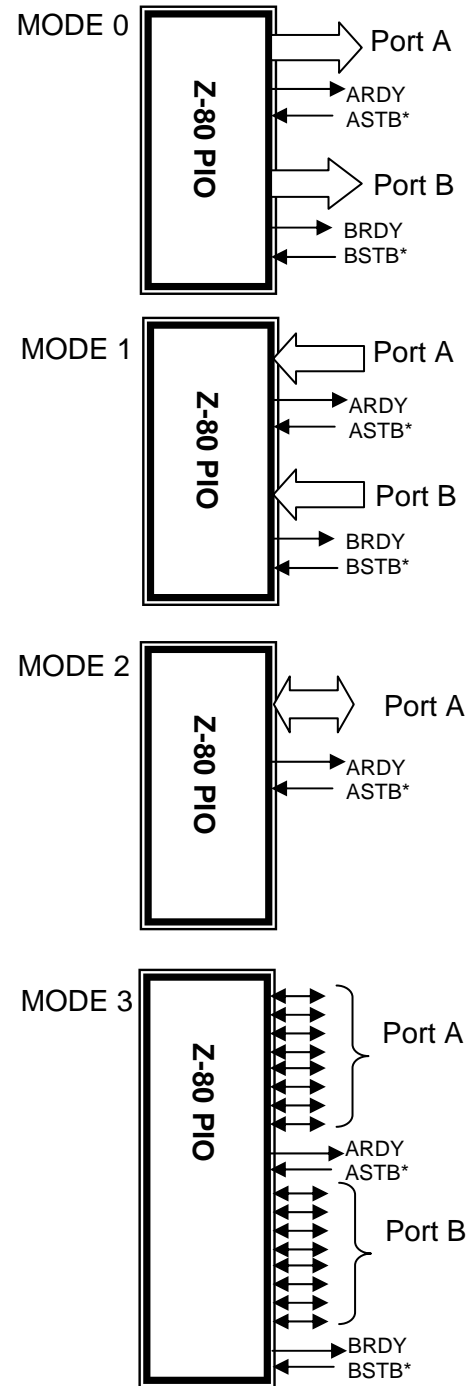
1.1. I/O Paralel

I/O paralel adalah jenis I/O yang mengalihkan data pada setiap port saluran secara paralel. Alih data secara paralel bekerja mengalihkan data secara serempak dari D0 sampai dengan D7. Ada sejumlah saluran pengalir data yang disebut dengan Port. Biasanya setiap port terdiri dari 8 bit saluran. Setiap port paralel dapat diprogram fungsi dan arah aliran data yang dibutuhkan. Port paralel menyediakan keuntungan pada kecepatan akses karena data ditransmisikan secara simultan. Berikut dibahas dua jenis I/O paralel yang sangat populer digunakan di lapangan yaitu Z-80 PIO dan PPI 8255.

1.1.1. Z-80 PIO (*Programmable Input Output*)

IC Z-80 PIO adalah IC I/O paralel terprogram buatan Zilog yang perilakunya dapat disetel menggunakan program. Z-80 PIO adalah salah satu chip yang diproduksi untuk fasilitas antar muka Z-80 CPU dengan peralatan input output. Z-80 PIO memiliki kelengkapan sebagai berikut:

- Dua peripheral port antar muka paralel 8 bit *independent* dengan kendali jabat tangan.
- Penggerak I/O terinterupsi.
- Empat mode operasi
 - Mode 0 : Byte Output dengan jabat tangan
 - Mode 1 : Byte Input dengan jabat tangan
 - Mode 2 : Byte Bidirectional dengan jabat tangan (hanya untuk Port A)
 - Mode 3 : untuk Bit kontrol
- Logika interupsi dengan prioritas *daisy chain*.
- Semua input dan output kompatibel dengan TTL.
- Susunan pin IC Z-80 PIO dilukiskan seperti Gambar 4.2.
- Gambar 4.1 melukiskan diagram pengaturan mode kerja Z-80 PIO.



Gambar 4.1. Diagram Mode Kerja Z-80 PIO

Z-80 PIO terdiri dari dua port yaitu **Port A** dan **Port B**. Masing-masing port dilengkapi dengan pena-pena jabat tangan. Dengan 40 pin dalam dua lajur fungsi masing-masing pin dapat dikelompokkan dalam empat kelompok yaitu:

➤ **Kelompok Bus Data**

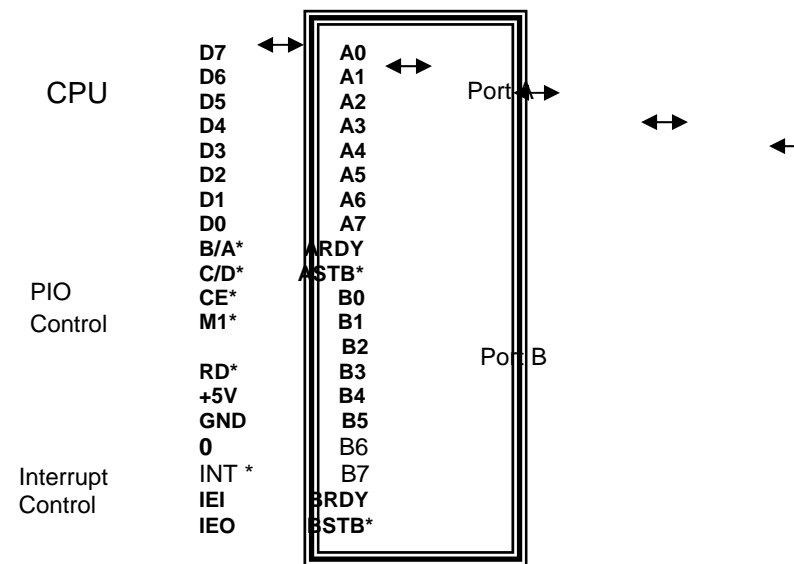
- D0 – D7 adalah bus data 8 bit dua arah digunakan sebagai saluran data dan kata perintah.
- A0 - A7 adalah saluran dua arah untuk transfer data antara peralatan I/O dan Port A.
- ARDY dan ASTB adalah saluran sinyal status dan sinyal kontrol untuk operasi jabat tangan pada Port A
- B0 – B7 merupakan saluran dua arah untuk transfer data antara I/O dan Port B.
- BRDY dan BSTB adalah saluran sinyal status dan sinyal kontrol untuk operasi jabat tangan pada Port B

➤ **Kelompok Kontrol**

- B/A* sel adalah pin saluran sinyal pemilih port. Pada kondisi rendah (0) yang aktif adalah Port A, dan Port B aktif jika pin ini berkondisi tinggi (1).
- C/D* sel adalah pin saluran sinyal pemilih register kontrol atau register data. Jika C/D*= 0 register yang aktif adalah register data dan C/D*= 1 register yang aktif adalah register perintah.



Gambar 4.2a. Bentuk fisik IC Z-80 PIO

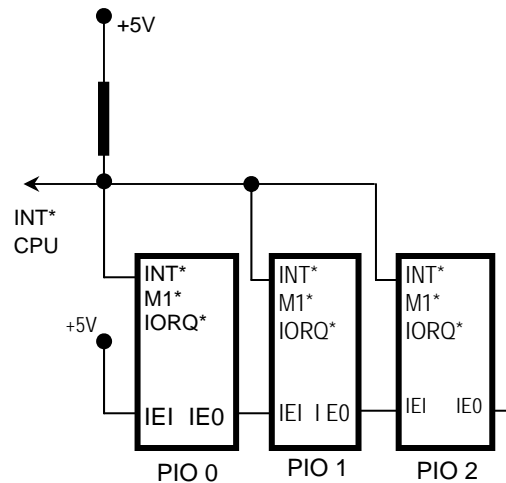


Gambar 4.2b. Susunan Pin IC Z-80 PIO

- c. CE* adalah sinyal aktif rendah yang berfungsi sebagai pin pengaktif chip Z-80 PIO.
- d. M1* adalah sinyal aktif rendah bekerja mensinkronkan *kerja interrupt logic*. Pada saat M1* dan RD* aktif, maka Z-80 CPU melakukan fetching sebuah instruksi ke memori. Sebaliknya pada saat M1* dan IORQ* aktif, CPU melakukan pengenalan interupsi. Dan jika M1* aktif tanpa IORQ* atau RD*, Z-80 PIO ada dalam keadaan reset. Dalam keadaan Reset PIO akan berperilaku:
- Jalur bus data berkeadaan impedansi tinggi (*tri state*). Pena RDY aktif tinggi. Kedua Port dibuat berkeadaan modus 1 (sebagai Input).
 - Register alamat vector interupsi tidak di Reset
 - Interupsi oleh kedua Port dibuat non aktif.
 - Register keluaran kedua Port dibuat Reset.
- e. IORQ* adalah sinyal *Input Output Request* aktif rendah bekerja pada saat CPU mentransfer perintah atau data ke Z-80 CPU.
- f. READ* adalah sinyal aktif rendah yang menunjukkan CPU membaca data dari I/O.

➤ Kelompok Interrupt

- INT* adalah sinyal interrupt aktif rendah yang digunakan oleh PIO untuk memintakan layanan interupsi.
- IEI adalah sinyal *Interrupt Enable Input* aktif tinggi yang menunjukkan PIO siap menerima layanan interupsi.
- IEO adalah sinyal *Interrupt Enable Output* aktif tinggi yang menunjukkan PIO telah melayani interupsi.
- Kedua pena IEI dan IEO digunakan untuk menentukan prioritas interupsi. Prioritas interupsi pada PIO menggunakan sistem Tusuk Sate (*Daisy Chain*) seperti Gambar 4.3 di bawah ini.



Gambar 4.3. Susunan Interupsi bertingkat

Setiap IC PIO dapat menghasilkan interupsi jika pin IEI berkeadaan tinggi. PIO0 memiliki prioritas interupsi tertinggi baru disusul oleh PIO1, dan PIO2. Jika PIO0 tidak membangkitkan interupsi maka IEO-PIO0 aktif berlogika 1 dan memberi

kesempatan interupsi pada PIO1. Jika PIO1 membangkitkan interupsi maka IO-PIO1 berlogika 0 dan mengunci interupsi PIO2.

➤ **Kelompok Status Kontrol Port**

a. ASTB^* adalah sinyal strobe Port A, aktif rendah yang operasinya tergantung pada mode operasi yang dipilih.

- i) Mode 0: menunjukkan keadaan peralatan I/O telah menerima data yang dikirim oleh PIO.
- ii) Mode 1: menunjukkan keadaan data telah dikirim ke register Port A oleh peralatan I/O.
- iii) Mode 2: menunjukkan keadaan data dari register Port A telah diletakkan pada bus data dan kemudian data telah diterima oleh peralatan I/O.
- iv) Mode 3: pulsa ini secara internal ditahan oleh PIO (tidak dimanfaatkan).

b. ARDY adalah sinyal ready aktif tinggi untuk Port A bekerja tergantung mode operasi sebagai berikut :

- i) Mode 0: menunjukkan register Port A berisi data byte dan telah disiapkan pada saluran bus data untuk ditransfer ke peralatan I/O.
- ii) Mode 1: menunjukkan keadaan register data Port A kosong dan siap menerima data word berikutnya.

iii) Mode 2: menunjukkan keadaan register data Port A telah siap untuk diambil oleh peralatan I/O. Data akan dikeluarkan jika ada sinyal STB^* .

iv) Mode 3: tidak dimanfaatkan

c. BSTB^* adalah sinyal masukan strobe untuk Port B aktif rendah dimana operasinya sama dengan sinyal ASTB^* .

d. BRDY adalah sinyal keluaran ready aktif tinggi untuk Port B dengan operasi kerja sama dengan ARDY . Masing-masing Port dilengkapi dengan dua register, yaitu register data dan register perintah. Selengkapnya Z-80 PIO memiliki empat buah register yaitu:

- Register Data A
- Register Data B
- Register Kontrol Port A
- Register Kontrol Port B

Register data digunakan untuk memegang data dan register perintah digunakan untuk mengatur mode kerja dan perilaku masing-masing port. Pemilihan register-register pada Z-80 CPU dikerjakan melalui pena port B/A^* dan pena Control/Data seperti Gambar 4.4.

Pada mikrokomputer MPF-1 atau pada mikrokomputer GMS-1, jalur address A0 dan

A1 dihubungkan dengan pin B/A* dan pin C/D*, sedangkan saluran address A2 s/d A5 tidak dikodekan, address A6 dan A7 dikodekan menggunakan Dekoder 74LS139. IC 74LS139 adalah IC dekoder dua ke empat artinya ada dua masukan A dan B dan empat keluaran yaitu Y0*, Y1*, Y2*, Y3*. Y0*=0 jika A=0 dan B=0, Y1*=0 jika A=1 dan B=0, Y2*=0 jika A=0 dan B=1, Y3*=0 jika A=1 dan B=1. Output dekoder diambil dari output Y2*. Output Y2* berlogika 0 jika input A=0 dan input B=1. Bentuk rangkaian dan pengalamatan Z-80 PIO seperti Gambar 4.5. Jika A3 s/d A5 diberi nilai 0 maka alamat port Z-80 PIO adalah seperti Gambar 4.6.

Oleh karena A5, A4, A3, A2 tidak dikodekan dalam pengalamatan, maka ada 2⁴ yaitu 16 jenis pengalamatan sebagai alamat duplikat untuk setiap Port. Ke enam belas alamat duplikat untuk setiap port tersebut adalah :

Data PORT A: 80H 84H 88H 8CH
 90H 94H 98H 9CH
 A0H A4H A8H ACH
 B0H B4H B8H BCH

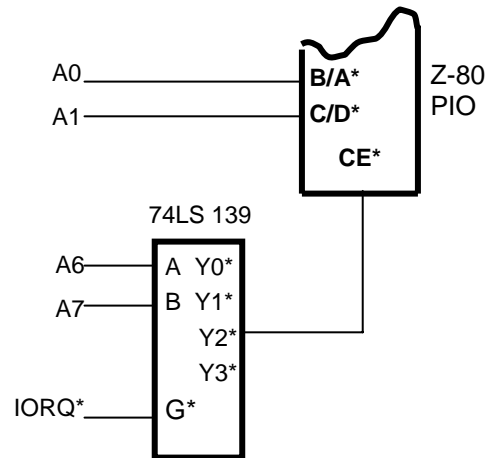
Data PORT B: 81H 85H 89H 8DH
 91H 95H 99H 9DH
 A1H A5H A9H ADH
 B1H B5H B9H BDH

Kontrol PORT A: 82H 87H 8AH 8EH
 92H 97H 9AH 9EH
 A2H A7H AAH AEH
 B2H B7H BAH BEH

Kontrol PORT B: 83H 88H 8BH 8FH
 93H 98H 9BH 9FH
 A3H A8H ABH AFH
 B3H B8H BBH BFH

C/D*	B/A*	Register
0	0	Data Port A
0	1	Data Port B
1	0	Kontrol Port A
1	1	Kontrol Port B

Gambar 4.4. Model pemilihan register pada Z-80 PIO



Gambar 4.5. Pengalamatan PIO

B	A	X	X	X	X	C/D*	B/A*	alamat	Register
A7	A6	A5	A4	A3	A2	A1	A0		
1	0	0	0	0	0	0	0	80H	Data Port A
1	0	0	0	0	0	0	1	81H	Data Port B
1	0	0	0	0	0	1	0	82H	Kontrol Port A
1	0	0	0	0	0	1	1	83H	Kontrol Port B

Gambar 4.6. alamat Z-80 PIO

Alamat port masing-masing port ada 16 alamat, dapat digunakan dan dipilih salah satu untuk setiap port.

o Pemrograman Z-80 PIO

Perilaku masing-masing Port pada PIO dapat diatur melalui register kontrol masing-masing port. Pengaturan perilaku Port menggunakan sebuah data 8 bit yang disebut dengan **Control Word**.

Format Control Word untuk beberapa fungsi pemrograman adalah sebagai berikut:

o SET MODE OPERASI

Format set mode operasi untuk PORT A dan PORT B adalah seperti Gambar 4.7.

D7	D6	D5	D4	D3	D2	D1	D0
M	M	X	X	1	1	1	1

0 0 = Mode 0
 0 1 = Mode 1
 1 0 = Mode 2
 1 1 = Mode 3

Gambar 4.7. Format Control Word

D5 dan D4 dapat bernilai 0 atau 1, maka untuk setiap Mode ada 4 kemungkinan data Control Word yaitu :

Mode 0: 0FH, 1FH, 2FH, 3FH
Mode 1: 4FH, 5FH, 6FH, 7FH
Mode 2: 8FH, 9FH, AFH, BFH
Mode 3: CFH, DFH, EFH, FFH

Untuk mode 3 harus diteruskan dengan pengaturan arah bit Port baik untuk Port A maupun Port B seperti Gambar 4.8.

o SET Bit I/O untuk Mode 3

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	X	X

Gambar 4.8. Format Bit set-reset

Jika X bernilai 0 berarti set bit sebagai output. Jika X berniali 1 berarti set bit sebagai input.

Contoh :

Jika PORT A bekerja pada mode 3 dimana PA0-PA3 sebagai Output dan PA4-PA7 sebagai Input. Maka format programnya inisialisasinya adalah:

LD A, CFH : Control Word Mode 3
OUT (Kontrol A), A : Kirim Ke register Kendali A

LD A, F0H : Bit PA7-PA4=1: Input
 Bit PA3-PA0=0: Output

OUT (Kontrol A), A : Kirim Ke register Kendali A

Data control word mode 3 adalah CFH. Dapat juga menggunakan data DFH,EFH, dan FFH. Data CFH diisikan ke register A lalu dikirimkan ke register kontrol A. Sampai langkah ini port A telah terpilih bekerja dengan mode 3 (lihat Gambar 4.1). Untuk menetapkan fungsi masing-masing bit port A dimana PA0 sampai dengan PA3 bekerja sebagai output dan PA4 sampai dengan PA7 sebagai input diperlukan data F0H

untuk dikirim kembali ke register kontrol A. Untuk kasus port B caranya sama seperti cara pemrograman port A.

Untuk kasus yang lain jika seluruh saluran berfungsi sama misalnya seluruh saluran Port A sebagai output. Maka mode yang dipilih adalah mode 0. Program inisialisasi untuk port A sebagai output adalah sebagai berikut:

LD A, 0FH : Control Word Mode 0
OUT (Kontrol A), A : Kirim Ke register Kendali A

Cukup dua langkah karena tidak perlu melakukan set-reset bit. Jika dikehendaki bekerja sebagai input byte melalui port A maka mode yang dipilih adalah mode 1. Program inisialisasi untuk port A sebagai input adalah sebagai berikut:

LD A, 4FH : Control Word Mode 1
OUT (Kontrol A), A : Kirim Ke register Kendali A

Pemilihan mode kerja baik untuk port A maupun untuk port B berkaitan dengan kebutuhan antarmuka terhadap alat input dan alat output. Untuk mengendalikan alat output membutuhkan mode 0 atau mode 3. Sedangkan untuk mengendalikan alat input dibutuhkan mode 1 atau mode 3. Kiranya menjadi jelas fungsi port akan terbentuk setelah ada program inisialisasi. Sebelum diinisialisasi port belum menunjukkan fungsi.

o SET ENABLE-DISABLE INTERUPSI

Untuk Mode 0, Mode 1, dan Mode 2 interupsi dapat diaktifkan atau di non aktifkan dengan melakukan setting Control Word seperti Gambar 4.9 berikut:

D7	D6	D5	D4	D3	D2	D1	D0
E/D	X	X	X	0	1	1	1

D7 = 0 Interupsi Disable

D7 = 1 Interupsi Enable

Gambar 4.9. Format control word interupsi

Dari Gambar 4.9. terlihat D0=1, D1=1, D2=1 sedangkan D6, D5, dan D4 tidak terkodekan. Maka ada 2^3 atau delapan kemungkinan data untuk mengeset Interupsi Disable dan Interupsi Enable. Kedelapan data itu masing-masing adalah:

Interupsi Disable
Data Control Word 07H, 17H, 27H, 37H
 47H, 57H, 67H, 77H

Interupsi Enable
Data Control Word 87H, 87H, A7H, B7H
 C7H, D7H, E7H, F7H

o SET KENDALI INTERUPSI

Format dari kendali Interupsi adalah seperti Gambar 4.10 berikut :

D7	D6	D5	D4	D3	D2	D1	D0
E/D	K	A	DW	0	1	1	1

Gambar 4.10. Format kendali interupsi

D3, D2, D1, D0	Sebagai penentu kendali interupsi
DW (Data Word)	1 : data word berikutnya Mask Bit 0 : tidak ada data word
A (Aktif)	1 : aktif High 0 : Aktif Low
K (Konfigurasi)	1 : Konfigurasi AND 0 : Konfigurasi OR
E/D (enable/Dis)	1 : Interrupt enable 0 : Interrupt Disable

o SET VEKTOR INTERUPSI

Untuk melakukan setting vector interupsi data 8 bit diperlukan oleh kontrol register PIO. Format set interupsi pada PIO adalah seperti Gambar 4.11.

D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	X	X	X	0

Gambar 4.11. Format set interupsi PIO

Bit D0 = 0 dinyatakan oleh CPU sebagai vektor interupsi. Isi register ini akan dikirim ke bus data sewaktu CPU mengakui interupsi yaitu dengan serentak mengaktifkan M1* dan IORQ*. Pada modus interupsi IM1, isi register ini digabungkan dengan isi register I dari CPU, untuk menunjukkan tempat alamat penyimpanan subrutin pelayanan interupsi.

Andaikata register I = 19H, dan isi register vector interupsi V = 00100000b = 20H

maka pada lokasi 1920H tersimpan bait bawah subrutin pelayanan interupsi, dan alamat 1921H tersimpan byte alamat subrutin tersebut. Perlu diperhatikan alamat subrutin interupsi harus pada daerah lokasi memori aktif dan cenderung pada lokasi memori RWM.

Cara memuat vector interupsi pada Z-80 CPU :

```
LD A, 20H ; Byte bawah
OUT (82H), A; Kirim ke kontrol Port A
alamat 82H
```

Alamat 82H dibentuk oleh rangkaian pengkode alamat pada Gambar 4.5. Angka 82H merujuk pada alamat bagian kontrol dari Port A. Jika dalam sistem mikroprosesor menggunakan pengalamatan lain maka pengalamatan kontrol Port A harus disesuaikan bukan 82H. Ini penting untuk menjadi perhatian. Kesalahan nilai alamat akan membuat program interupsi tidak berjalan.

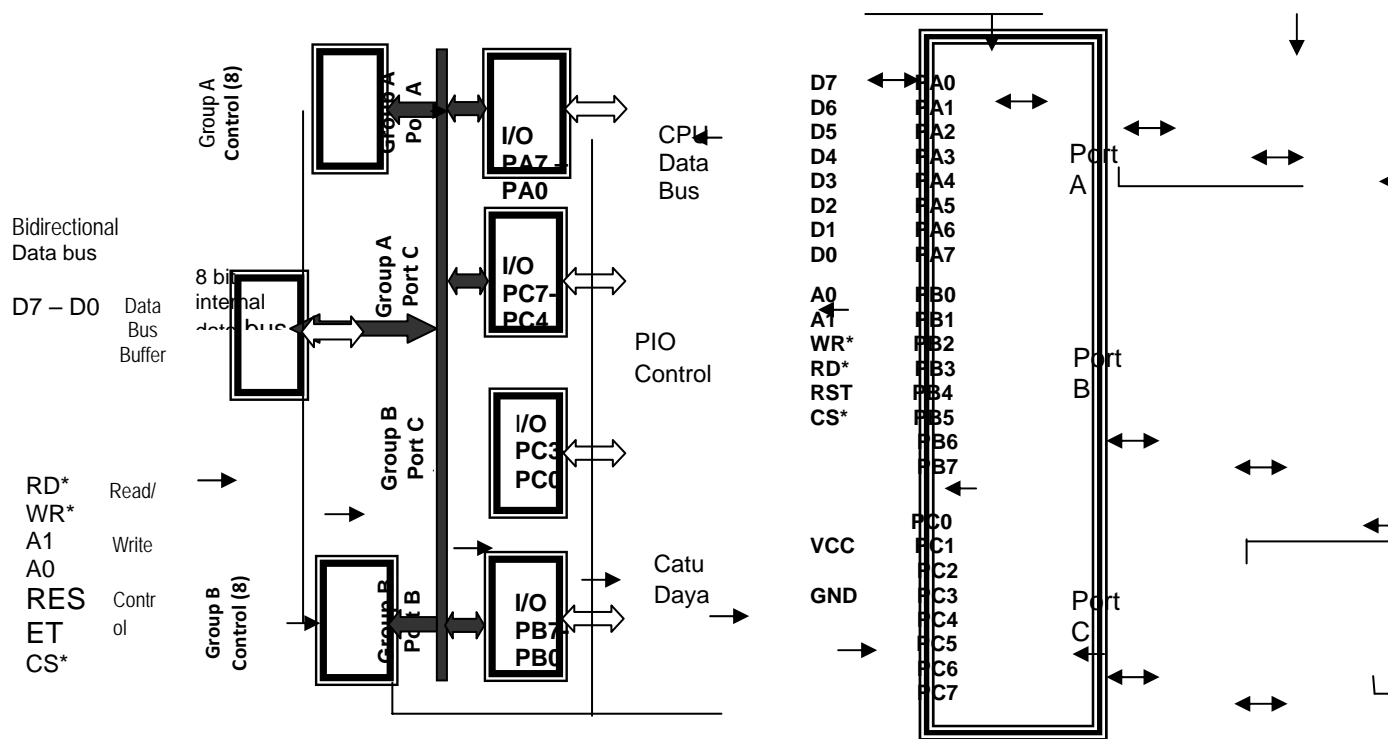
1.1.2. PPI 8255

PPI 8255 adalah chip *Programmable Peripheral Interface*, berfungsi untuk antar muka paralel dengan perilaku dapat diatur dengan program. PPI 8255 terdiri dari tiga port I/O 8 bit yaitu : **Port A**, **Port B**, dan **Port C**. Masing-masing port dapat dibuat menjadi port masukan maupun port keluaran. Gambar 4.12. menunjukkan bentuk fisik, konfigurasi susunan pin, dan diagram blok bagian dalam dari PPI 8255.

PPI 8255 sangat populer dan banyak digunakan dalam sistem-sistem antar muka paralel.



Gambar 4.12.a. Bentuk fisik PPI 8255
Sumber: www.google.co.id/images



Gambar 4.12.b Blok Diagram PPI 8255

Gambar 4.12.c. Susunan Pin PPI 8255

PPI 8255 memiliki buffer bus data dua arah, yang berarti dapat berfungsi baik sebagai port input maupun port output. Arah aliran data dapat dijelaskan menggunakan pengaturan logika Read/Write. Secara mudah dapat diuraikan dengan format Gambar 4.13. berikut.

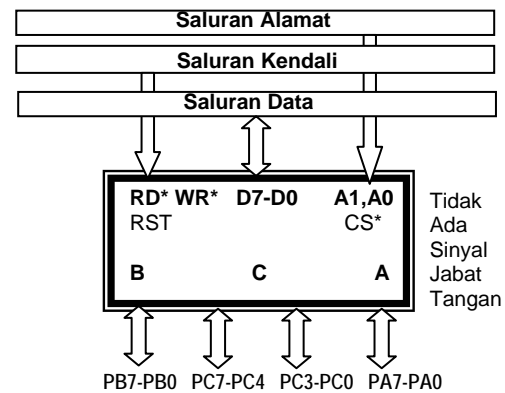
PPI 8255 bekerja dalam tiga mode, yaitu :

1. **Mode 0** : Port A, Port B, dan Port C bekerja sebagai port I/O sederhana tanpa jabat tangan. Pada mode ini CPU sama sekali tidak memperhatikan status 8255. CPU mentransfer data tanpa mempersoalkan apa yang terjadi pada 8255. Port A dan Port B bekerja sebagai port 8 bit sedangkan Port C dapat dibuat bekerja dalam 8 bit atau berdiri sendiri dalam 4 bit lower dan 4 bit upper secara terpisah. Pemakaian mode 0 pada PPI 8255 secara diagram dapat digambarkan pada Gambar 4.14.

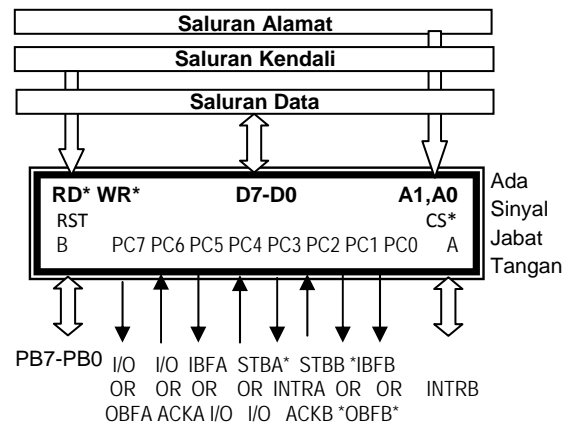
2. **Mode 1** : Port A, Port B bekerja sebagai port I/O dengan jabat tangan menggunakan sebagian dari pena Port C. Saluran PC0, PC1, dan PC2 berfungsi sebagai saluran jabat tangan untuk Port B sedangkan Port A menggunakan saluran PC3, PC4, dan PC5 sebagai sinyal jabat tangan. PC6 dan PC7 dapat digunakan untuk saluran I/O. Diagram operasi 8255 pada mode 1 digambarkan pada Gambar 4.15.

Logika Pin Kendali						Operasi
CS*	Reset	RD*	WR*	A1	A0	
0	0	1	0	0	0	Penulisan ke Port A
0	0	1	0	0	1	Penulisan ke Port B
0	0	1	0	1	0	Penulisan ke Port C
0	0	1	0	1	1	Penulisan ke Reg. Kontrol
0	0	0	1	0	0	Pembacaan dari Port A
0	0	0	1	0	1	Pembacaan dari Port B
0	0	0	1	1	0	Pembacaan dari Port C
1	0	x	x	x	x	8255 tidak aktif

Gambar 4.13. Format Pembacaan dan Penulisan PPI 8255

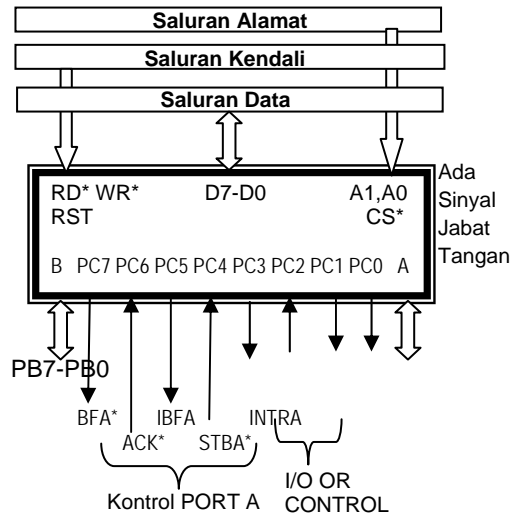


Gambar 4.14. Diagram Operasi PPI 8255 Mode 0



Gambar 4.15. Diagram Operasi PPI 8255 Mode 1

3. Modus 2: Hanya Port A dapat dibuat sebagai port I/O dua arah dengan jabat tangan. Port A dapat digunakan sebagai port untuk transfer data dua arah dengan jabat tangan. Ini artinya data dapat masuk atau keluar dari saluran yang sama. Mode ini mengembangkan sistem saluran (bus) ke mikroprosesor atau mentransfer byte data ke dan dari floppy disk controller. Pada mode 2 saluran PC3 sampai PC7 digunakan sebagai saluran jabat tangan untuk Port A. Bentuk operasi 8255 sebagai mode 2 digambarkan pada diagram Gambar 4.16.

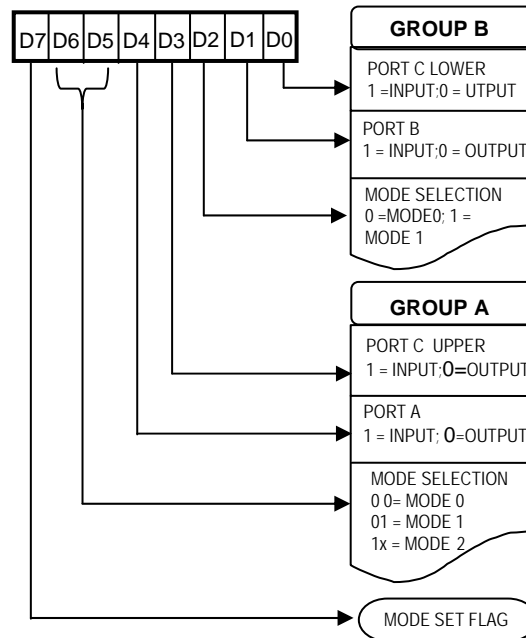


Gambar 4.16. Diagram Operasi PPI 8255 Mode 2

• **Penyusunan dan Pengiriman Control Word**

Format *control word* PPI 8255 ditunjukkan pada Gambar 4.17. disamping.

Gambar 4.17. digunakan untuk menformat nilai *control word* berdasar pada mode kerja. Sedangkan Gambar 4.18. digunakan untuk menformat nilai *control word* untuk Port C pada operasi set/reset bit. Dengan bantuan format Gambar 4.17 dan Gambar 4.18 *control word* mudah disusun.



Gambar 4.17 Format Control Word Mode Set

Pada komputer mikro MPF-1 PPI 8255 digunakan sebagai I/O komunikasi untuk mengatur *display seven segment* dan keyboard, audio tone ke Loud Speaker, dan input alat rekam. Konfigurasi pengalamatan PPI 8255 di MPF-1 seperti Gambar 4.20.

Jika A3 s/d A5 diberi nilai 0 maka alamat port PPI 8255 dapat digambarkan seperti Gambar 4.19.

B	A	X	X	X	X	A1	A0	alamat	Register
A7	A6	A5	A4	A3	A2	A1	A0		
0	0	0	0	0	0	0	0	00H	Port A
0	0	0	0	0	0	0	1	01H	Port B
0	0	0	0	0	0	1	0	02H	Port C
0	0	0	0	0	0	1	1	03H	Register Kontrol

Gambar 4.19. Pemetaan alamat PPI 8255

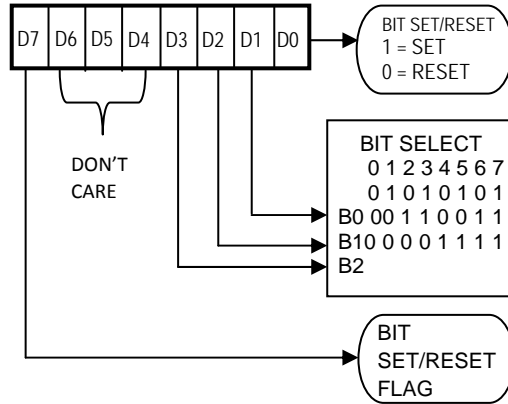
Oleh karena A5, A4, A3, A2 tidak dikodekan dalam pengalamatan, maka ada 2⁴ yaitu 16 jenis pengalamatan sebagai alamat duplikat untuk setiap Port. Ke enam belas alamat duplikat untuk setiap port tersebut adalah:

PORT A: 00H 04H 08H 0CH
 10H 14H 18H 1CH
 20H 24H 28H 2CH
 30H 34H 38H 3CH

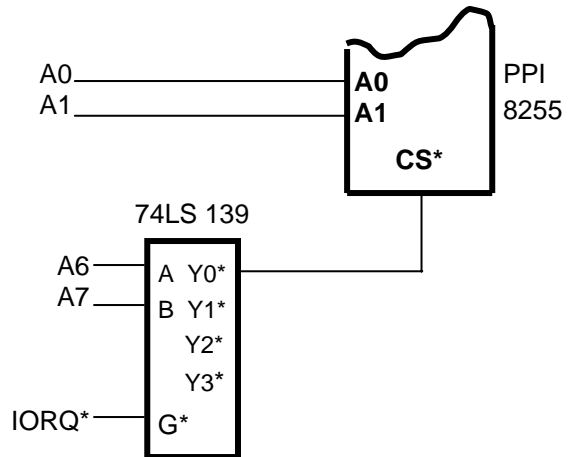
PORT B: 01H 05H 09H 0DH
 11H 15H 19H 1DH
 21H 25H 29H 2DH
 31H 35H 39H 3DH

PORT C: 02H 06H 0AH 0EH
 12H 16H 1AH 1EH
 22H 26H 2AH 2EH
 32H 36H 3AH 3EH

Kontrol Reg: 03H 07H 0BH 0FH
 13H 17H 1BH 1FH
 23H 27H 2BH 2FH
 33H 37H 3BH 3FH



Gambar 4.18. Format Control Word Port C Bit Set/Reset



Gambar 4.20. Pengalamatan PPI 8255

Ada duplikasi alamat port dimana satu port memiliki 16 alamat. Masing-masing alamat dapat digunakan dan dipilih salah satu untuk setiap port. Untuk membangun

fungsi PPI 8255 maka harus dilakukan inisialisasi.

Contoh :

Untuk membuat PORT A, PORT B, PORT C, sebagai Output mode 0, pertama harus ditetapkan nilai data control word. Dengan menggunakan Gambar 4.17 data control word dapat dijabarkan seperti Gambar 4.21 berikut.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

Gambar 4.21. Data control word PORT A, PORT B, PORT C, sebagai Output mode 0

Data control word hasil penjabaran dari Gambar 4.21 adalah 1000 0000b atau 80H. Nilai ini digunakan sebagai data control word yang dimasukkan ke register control. Dengan demikian format programnya inisialisasinya adalah:

LD A, 80H : Register A diisi data 80H sebagai Control Word Mode 0 semua port sebagai Output

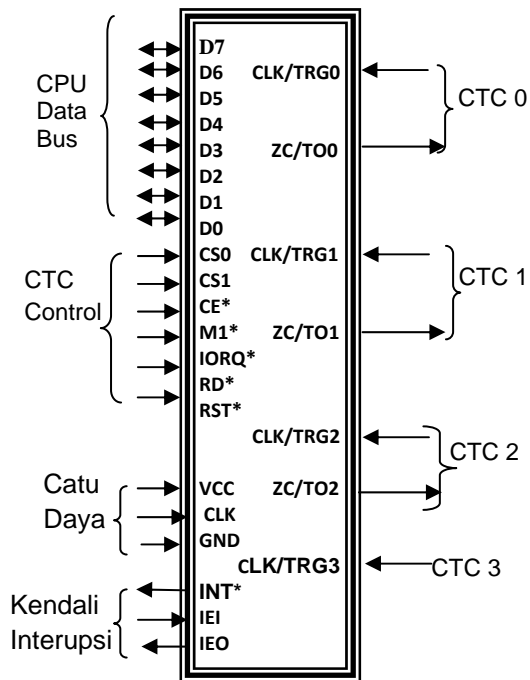
OUT (Kontrol), A : Kirim data 80H di register A Ke register Kendali

2. PENCACAH DAN PEWAKTU TERPROGRAM (CTC)

Pencacah dan pewaktu sangat diperlukan dalam sistem mikroprosesor. Pencacah dan pewaktu terprogram banyak jenis dan ragamnya. Intel mengeluarkan seri Programmable Interval Timer (PIT 8253), Zilog mengeluarkan seri Z-80 Counter Timer Clock (CTC).

CTC adalah IC pencacah dan pewaktu terprogram yang dapat diinterfacekan ke mikroprosesor Z-80 CPU. IC ini dapat mencacah pulsa atau peristiwa ada-tidak, sesuai dengan keadaan dari peristiwa, misalnya mencacah jumlah pengunjung dalam suatu Super Market, Jumlah bebek yang keluar dari sebuah kandang, jumlah kendaraan yang masuk dalam suatu tempat parkir dan sebagainya. IC ini juga dapat digunakan sebagai pengukur tegangan dengan menambahkan IC LM 331 konverter dari tegangan ke frekuensi.

Disamping untuk pencacah CTC dapat digunakan sebagai pewaktu/timer yang dapat membuat komputer diinterupsi tiap selang waktu tertentu. Dalam fungsi ini CTC dikatakan berfungsi sebagai **Real Time Clock**. Susunan Pin CTC dapat digambarkan seperti Gambar 4.22.



Gambar 4.22. Susunan Pin Z-80 CTC

Fungsi masing-masing pin pada CTC dapat dijelaskan sebagai berikut:

- **CE* = Chip Enable**
1 : tidak aktif
0 : aktif
- **CS0, CS1 = Channel Select**
Dua jalur penentu alamat untuk memilih salah satu Counter/Timer dari 4 Counter/Timer yang ada seperti Gambar 4.23.

Kondisi Logika		Counter/Timer
CS1	CS0	
0	0	CTC 0
0	1	CTC 1
1	0	CTC 2
1	1	CTC 3

Gambar 4.23. Data Alamat CTC

➤ D0 - D7 = Data Bus

Jalur data yang dihubungkan dengan bus data dari CPU

➤ CLK = Clock

Clock satu phasa yang digunakan untuk mengatur sinkronisasi kerja CTC

➤ CLK/TRG = Clock / Trigger

Clock atau Triger luar untuk memicu pencacah atau timer

➤ ZC/TO = Zerro Count/Time Out

Berupa isyarat pulsa positif jika cacahan hitungan telah mencapai nol.

➤ IEI = Interrupt Enable Input

Digunakan untuk membentuk interupsi bertingkat dalam menentukan prioritas lebih dari satu pheriperal.

1: CTC dapat meminta layanan interupsi ke CPU

0: CTC tidak dapat meminta layanan interupsi ke CPU

➤ IEO = Interrupt Enable Output

Digunakan untuk membentuk interupsi bertingkat dalam menentukan prioritas lebih dari satu pheriperah.

1 : CPU tidak melayani interupsi CTC

0: CPU melayani interupsi CTC

➤ INT* = Interrupt Request dari CTC ke CPU

1 : CTC tidak melakukan interupsi

0: CTC melakukan interupsi

CTC adalah komponen yang bekerja sebagai pencacah turun (*Down Counter*). Sebagai down counter CTC diisi dengan sebuah bilangan sebagai tetapan awal. Tiap transisi pulsa pada masukan CLK/TRGn

akan menyebabkan bilangan tersebut berkurang satu. Pada waktu bilangan tersebut menjadi nol maka CTC akan menghasilkan transisi pada keluaran ZC/TOn (**Zero Count/Time Out**). Disamping itu pada keadaan ini CTC juga menghasilkan interupsi ke CPU.

Ada empat buah counter/timer pada CTC. Tiga buah counter/timer memiliki pasangan :

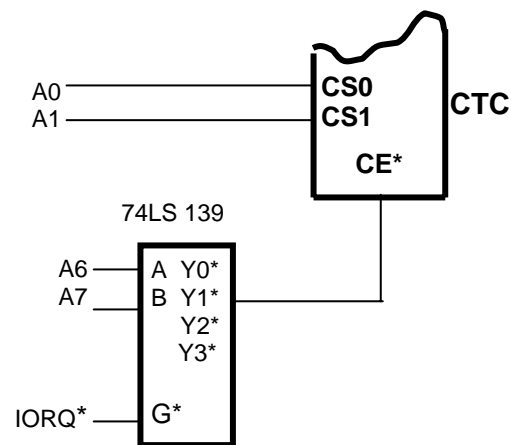
- CLK/TRG0 & C/TO0 : Untuk CTC 0
 - CLK/TRG1 & C/TO1 : Untuk CTC 1
 - CLK/TRG2 & C/TO2 : Untuk CTC 2
 - CLK/TRG3 : Untuk CTC 3
- Sebagai pencacah masukan

CLK/TRG 0 dihubungkan dengan sumber pulsa tegangan dari luar untuk mencacah pulsa. Tiap pulsa akan mengurangi bilangan 8 bit yang ada pada register dengan 1 (decrement). Setelah bilangan itu menjadi nol maka pena ZC/TO0 akan aktif high sekejap.

Jika keluaran ZC/TO0 dihubungkan dengan CLK/TRG 1 maka CTC0 dan CTC1 membentuk pencacah 16 bit. Jika CTC0 berdiri sendiri maka hanya dapat bekerja sebagai pencacah 8 bit.

Pencacah 8 bit dapat mencacah maksimum pulsa $2^8 = 256$ pulsa sedangkan pencacah 16 bit dapat mencacah $2^{16} = 65536$ pulsa. Selanjutnya jika keluaran ZC/TO1 dihubungkan dengan CLK/TRG 2 maka CTC0, CTC1, dan CTC2 membentuk

pencacah 2^{24} bit. Jika CTC0, CTC1, CTC2, dan CTC3 kita gunakan akan membentuk pencacah 32 bit. Untuk bekerja sebagai pewaktu maka masukan CLK/TRG dihubungkan dengan suatu pulsa trigger yang akan memulai pewaktuan. Setiap saluran CTC mempunyai register 8 bit yang dapat diisi data. Pada Komputer mikro MPF-1 CTC dipasang dengan konfigurasi seperti Gambar 4.24 dibawah.



Gambar 4.24 Pengalamatan CTC

Jika A3 s/d A5 diberi nilai 0 maka alamat port Z-80 PIO adalah seperti Gambar 4.25. berikut.

B	A	X	X	X	X	CS1	CS0	alamat	Register
A7	A6	A5	A4	A3	A2	A1	A0		
0	1	0	0	0	0	0	0	40H	CTC 0
0	1	0	0	0	0	0	1	41H	CTC 1
0	1	0	0	0	0	1	0	42H	CTC 2
0	1	0	0	0	0	1	1	43H	CTC 3

Gambar 4.25. Pengkodean alamat CTC

Oleh karena A5, A4, A3, A2 tidak dikodekan dalam pengalamatan, maka ada 2^4 yaitu 16 jenis pengalamatan sebagai alamat duplikat untuk setiap Port. Ke enam belas alamat duplikat untuk setiap port tersebut adalah :

CTC 0: 40H 44H 48H 4CH
50H 54H 58H 5CH
60H 64H 68H 6CH
70H 74H 78H 7CH

CTC 1: 41H 45H 49H 4DH
51H 55H 59H 5DH
61H 65H 69H 6DH
71H 75H 79H 7DH

CTC 2: 42H 46H 4AH 4EH
52H 56H 5AH 5EH
62H 66H 6AH 6EH
72H 76H 7AH 7EH

CTC 3: 43H 47H 4BH 4FH
53H 57H 5BH 5FH
63H 67H 6BH 6FH
73H 77H 7BH 7FH

➤ Pemrograman CTC

Masing-masing CTC mempunyai sebuah register 8 bit yang digunakan untuk :

- Memasang vektor interupsi
- Memprogram fungsi CTC
- Memasukkan bilangan awal cacahan

Tiap byte data yang dimasukkan ke dalam register bergantung pada isi bit D0. Bila D0 = 0 control word diartikan sebagai vektor interupsi. Dalam hal ini register diartikan seperti Gambar 4.26.

Format Bit Vektor Interupsi

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	X	X	0

Gambar 4.26. Format bit interupsi

Pemilihan saluran ditentukan oleh nilai D2 dan D1 dengan data pemilihan seperti Gambar 2.27. berikut.

D7	D6	D5	D4	D3	D2	D1	D0	data	Saluran
V7	V6	V5	V4	V3	X	X	0		
0	0	0	0	0	0	0	0	00H	CTC 0
0	0	0	0	0	0	1	0	02H	CTC 1
0	0	0	0	0	1	0	0	04H	CTC 2
0	0	0	0	0	1	1	0	06H	CTC 3

Gambar 4.27. Format data interupsi

Misalnya digunakan interupsi modus 2 (IM2), sehingga vektor diatas harus digabungkan dengan isi register I dalam CPU. Jika register I diisi 19H, interupsi CTC0 akan mencabang ke alamat 1900 (byte bawah dan 1901 byte atas. Interupsi CTC1 akan mencabang ke alamat 1902H dan 1903H dan CTC2 akan mencabang ke alamat 1904H dan 1905H, CTC3 akan mencabang ke alamat 1906H dan 1907H. Jadi vektor interupsi bersama register bersama register I menunjuk tempat alamat subrutin layanan interupsi.

Jika D0 = 1 maka control word diartikan lain oleh CTC. Dalam hal ini register

digunakan untuk memprogram operasi CTC seperti Gambar 4.28 di bawah ini.

D7	D6	D5	D4	D3	D2	D1	D0
Interrupt Enable	Mode Timer/Counter	Range 256/16	Slope +/-	Trigger ON/OFF	Load Time Constant	Reset	1

Gambar 4.28. Format bit control word

Bit D7 : *Interrupt Enable* digunakan untuk mengaktifkan atau menonaktifkan interupsi masing-masing saluran. Jika D7 = 1 maka interupsi aktif yaitu setelah cacah berisi nol maka CTC akan menghasilkan pulsa interupsi. Jika D7 = 0 interupsi tidak aktif.

Bit D6 : *Mode* digunakan untuk memilih pengaturan CTC menjadi Pencacah (D6=1) atau pewaktu (D6 = 0). Sebagai pewaktu keluaran CTC akan berupa isyarat pulsa pada pena ZC/TO dengan perioda : $tc \times P \times Tc$

Dimana : tc = perioda clock CPU, P = 256 atau 16, dan

Tc = isi register tetapan waktu.

Bit D5 dan Bit D3 hanya mempunyai arti jika CTC diprogram sebagai pewaktu.

Bit D5 : Batas 256 atau 16; Jika D5 = 1 maka clock CPU dibagi 256 dan jika D5 = 0 maka clock CPU dibagi 16.

Bit D4 : Slope +/- ; Jika D4 = 1 maka pewaktu atau pencacah akan dipicu oleh tepi + yaitu perubahan

dari nol ke satu. Jika D4 = 0 maka CTC sebagai pewaktu akan dipicu oleh tepi – yaitu perubahan dari satu ke nol. Pada saat sebagai pencacah CTC peka terhadap transisi + atau -.

Bit D3 : Trigger ON/OFF untuk modus Timer saja. Jika D3 = 1 maka timer diaktifkan oleh isyarat picu dari luar. Sedangkan jika D3 = 0 maka timer diaktifkan oleh clock dari dalam CPU.

Bit D2 : Load Timer Constant ,

jika D2 = 1 maka data berikutnya yang dilempar ke register saluran yang bersangkutan akan diartikan sebagai data tetapan waktu (Timer Constant). Jika data yang diisikan pada waktu CTC sedang pencacahan turun, maka data baru akan dimuat setelah isi data cacahan menjadi nol. Setelah tercacah turun menjadi nol maka isi tetapan waktu akan dimuat kembali untuk cacahan turun.

Jika D2 = 0 berarti isi register cacahan tetap tak berubah. Keadaan ini digunakan untuk mengubah bit-bit tanpa mempengaruhi isi data cacahan.

Bit D1 : RESET Jika bit D1 = 1 maka saluran akan direset, saluran akan berhenti mencacah, tanpa menambah isi register saluran. Jika D2 = 1 dan D1 = 1 maka saluran akan kembali mencacah setelah data cacahan dimuat kembali.

Contoh KASUS Penerapan CTC untuk Pengukuran

Penggunaan CTC untuk pengukuran besaran fisis misalnya suhu air yang dipanaskan dapat dibaca secara terprogram setiap dua menit. Pembacaan setiap sampling data berlangsung selama 0,1 detik dengan jumlah pengukuran 10 kali pengukuran.

Untuk mengukur suhu digunakan rangkaian yang dapat mengubah nilai suhu menjadi frekuensi atau menjadi perioda isyarat pulsa. Rangkaian ini sering disebut V/F (*Voltage to Frekuensi Converter*). Untuk tujuan itu CTC dirangkai seperti Gambar 28.

Pengukuran dimulai dengan menekan tombol Switch. CD4040 (pencacah 12 bit) digunakan untuk memperlama keluaran ZC/TO0 dengan menggandakan perioda dengan 1024. CD4040 mencacah turun saluran CTC1. Dengan demikian CTC1 akan mencapai cacahan nol dan melakukan interupsi selang waktu.

$$\Delta T1 = tc \times P0 \times TC0 \times 1024 \times TC2$$

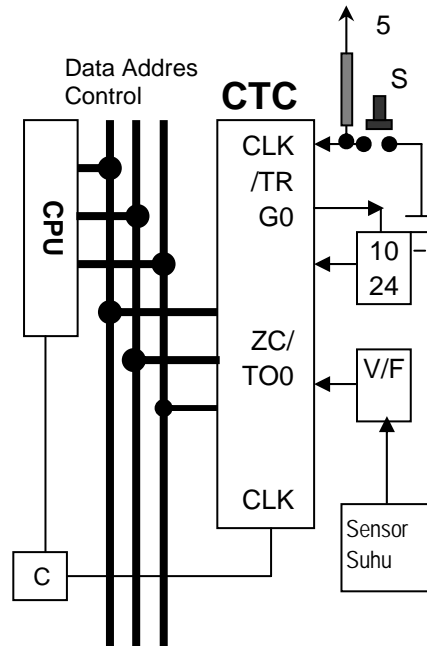
Dimana $tc = 0,5\mu S$ untuk frekuensi clock CPU 2 MHz.

$P0$ adalah nilai faktor skala CTC yaitu 16 atau 256, $TC0$ adalah nilai tetapan waktu CTC0, $P1$ adalah nilai skala awal untuk CTC1 dan $TC1$ adalah nilai tetapan waktu

CTC1. Untuk membuat $\Delta T1$ sekitar 2 menit = 120 S kita gunakan

$$P0 = 16, TC0 = 256, TC2 = 464$$

Maka kita peroleh $T1 = 119,5295 = 2$ menit.



Gambar 4.29. Rangkaian Pengukur Suhu dengan CTC

Pada akhir selang waktu ini CTC2 diaktifkan (bit RESET = 0), kemudian CPU disusun membuat loop pewaktuan 100 ms = 0,1 ms. Pada akhir ini CTC1 dibuat reset sehingga cacahn terhenti.

Register cacahan dibaca untuk menentukan banyaknya cacahan selama selang waktu 100 ms. Data ini disimpan di dalam memori, kemudian CPU diinterupsi

lagi oleh CTC1 untuk pengambilan data yang ke dua. Demikian seterusnya setelah 10 buah data suhu terbaca, CTC dibuat reset sampai menerima perintah lagi dari Switch.

♦ INISIALISASI

♦ Untuk CTC 0

CTC0 dibuat sebagai timer, peka terhadap slope, trigger aktif, TC = 01H, dan Interupsi aktif.

Vektor interupsi CTC 0 disiapkan pada 00H. Register I = 19H agar interupsi oleh CTC 0 memvektor ke subrutin alamat tersimpan pada 1900H dan 1901H. Misalnya alamat 1900 berisi A0H dan 1901 berisi 18H. Jadi Subrutin CTC 0 adalah di 18A0H. Data untuk memprogram CTC adalah seperti Gambar 4.30 berikut ini.

1. Vektor Interupsi CTC 0

D7	D6	D5	D4	D3	D2	D1	D0	Data
0	0	0	0	0	0	0	0	00H

Gambar 4.30a. Data Vektor interupsi

2. Kontrol:

D7	D6	D5	D4	D3	D2	D1	D0	Data
1	0	0	0	1	1	0	1	8CH

Gambar 4.30b. Data Kontrol

3. TC 0

D7	D6	D5	D4	D3	D2	D1	D0	Data
0	0	0	0	0	0	0	1	01H

Gambar 4.30c. Data TC0

Dengan cara ini jika Switch ditekan satu kali maka terjadi cacahan nol dan terjadi interupsi yang akan membuat CPU melaksanakan subrutin pada lokasi 18A0H.

Pada subrutin pelayanan interupsi ini, CTC diprogram lagi agar menjadi pewaktu yang dipicu dari dalam (Trigger Off), sehingga CTC 0 akan berlaku sebagai multivibrator astabil. Oleh karena itu CTC 0 harus diprogram kembali sebagai berikut:

- ♦ Non-aktifkan interupsi
- ♦ Modus Timer
- ♦ Range 16
- ♦ Slope +
- ♦ Load time constant TC = $256_{10} = FFH$
- ♦ Reset 0

Dengan demikian keluaran CTC 1 yaitu pena ZC/TO1 akan menghasilkan pulsa-pulsa dengan perioda

$$T1 = tc \times P1 \times TC1 = 0,5 \mu S \times 16 \times 256 = 2,048 \text{ mS}$$

Setelah frekuensi isyarat ini dibagi 1024 oleh pencacah 12 bit CD 4040, maka keluaran pencacah akan mempunyai perioda:

$$T2 = 1024 \times T1 = 2,097 \mu S$$

Pulsa-pulsa ini akan dicacah turun oleh CTC 2. Jika CTC 2 diisi dengan 60_{10} kita akan membuat interupsi pada CPU setiap $T3 = 60 \times T2 = 2$ menit. Interupsi ini digunakan untuk mengambil suhu secara periodik.

Selanjutnya rancangan inisialisasi CTC untuk pengambilan data suhu dilakukan melalui CTC 2.

- ✦ Buat CTC 2 sebagai pencacah
- ✦ Non Aktifkan interupsi
- ✦ Slope -
- ✦ Load time constant
TC= $256_{10} = FFH$
- ✦ Non aktifkan Reset

Selanjutnya buat timing loop 0,1 S dan pada akhir timing loop reset CTC2 dengan mengirim kata kontrol seperti di atas tetapi reset diaktifkan.

Jika isi register CTC dibaca oleh CPU maka akan diperoleh data FFH – N. Dimana N adalah data cacahan pulsa masuk tiap 0,1 sekond.

3. I/O SERIAL

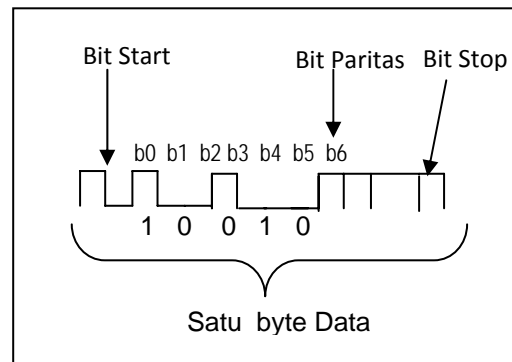
I/O serial adalah unit masukan keluaran yang bekerja atas dasar prinsip urut/seri. Dalam hal ini diperlukan proses konversi dari data paralel ke bentuk serial dan sebaliknya .

☞ Bentuk Data Seri

Ada dua cara mengalihkan data seri, yaitu dengan alih data :

- *Alih Data Asinkron dan*
- *Alih Data Sinkron.*

Alih data secara asinkron lebih sederhana dari pada alih data sinkron tetapi tidak dapat dilakukan secepat alih data sinkron. Pada alih data asinkron satu byte data, yang terdiri dari 8 bit data dikirim bit demi bit seperti Gambar 4.31 berikut:



Gambar 4.31. Bentuk alih Data Asinkron

Pada penerimaan data awal dari satu data ditandai dengan waktu transisi dari logika satu ke logika nol yang diikuti keadaan logika nol selama 1 bit. Bit ini disebut **bit start**. Setelah itu baru diterima

bit-bit data sebanyak 7 bit, yaitu bit b0, b1, b2, b3, b4, b5, dan b6. Bit kedelapan setelah bit start disebut **bit paritas**.

Bit paritas berhubungan dengan genap atau ganjilnya bit berkeadaan 1 dalam satu byte data. Bit paritas digunakan agar penerima dapat menentukan dan mendeteksi adanya kesalahan pada data yang diterima.

Pengirim dan penerima harus lebih dahulu sepakat apakah data yang dikirim menggunakan paritas genap atau paritas ganjil atau tanpa paritas.

Apabila alih data menggunakan paritas genap, maka bit paritas akan dibuat nol jika jumlah bit yang berlogika 1 genap. Dan bit paritas akan berlogika 1 jika jumlah bit berlogika 1 ganjil.

Apabila penerima mendapatkan jumlah bit 1 ganjil maka tentu ada kesalahan pada alih data. Kesalahan seperti ini disebut **Kesalahan Paritas**. Selanjutnya penerima dapat meminta kembali pengirim untuk mengirim ulang data tersebut.

Setelah bit paritas satu atau dua bit disediakan untuk bit stop. Setelah bit stop, setiap terjadi transisi ke nol, akan dianggap sebagai bit start untuk data berikutnya. Jumlah bit stop yang digunakan harus disepakati lebih dahulu antara pengirim dan penerima data.

Misalnya disepakati ada satu bit stop, dan ternyata setelah 7 bit data dan bit paritas, penerima tidak mendeteksi bit stop, maka penerima akan mengalami kesukaran menentukan bit start data berikutnya. Kesalahan semacam ini disebut **kesalahan bingkai (Frame error)**.

Pada alih data asinkron dengan adanya bit start dan bit stop pengiriman dan penerimaan data dapat dilaksanakan dengan baik. Namun kurang efesien sebab untuk mengirim 7 bit data butuh 10 bit data, termasuk bit start, bit paritas dan bit stop.

Pada alih data seri sinkron tak digunakan bit start dan bit stop. Jadi pengiriman data dapat berjalan lebih efisien. Untuk memastikan tidak terjadi kesalahan perlu dibuat sinkronisasi antara clock pengirim dan clock penerima data, dan dikirim data tertentu sebagai protokol alih data.

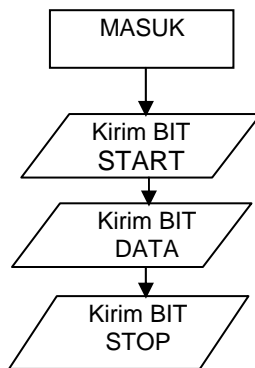
Pada alih data seri laju alih data dinyatakan dalam bit/detik yaitu banyaknya bit per detik yang disebut dengan Baudrate.

Ada dua teknik konversi yang ditawarkan yaitu :

- Teknik perangkat lunak
- Teknik perangkat keras.

3.1. I/O Serial Perangkat Lunak

Serialisasi dan deserialisasi suatu data diselenggarakan oleh suatu perangkat program. Pada masukan program menunggu sampai menerima suatu bit start, kemudian membaca bit data. Pada keluaran program mengirim suatu urutan bit demi bit. Diagram alir program I/O serial ditunjukkan pada Gambar 4.32.



Gambar 4.32. Diagram Alir Program I/O Serial

Prinsip utama serialisasi adalah merakit data 8 bit (atau lebih) di akumulator dan menggeser keluar bit demi bit pada frekuensi tertentu. Cara yang sederhana adalah mengeluarkan isi akumulator ke salah satu saluran dari port I/O (Port 0). Akumulator kemudian digeser ke kanan satu bit, diimplementasikan suatu tunda dan bit selanjutnya dikeluarkan sampai semua bit data paralel dikeluarkan.

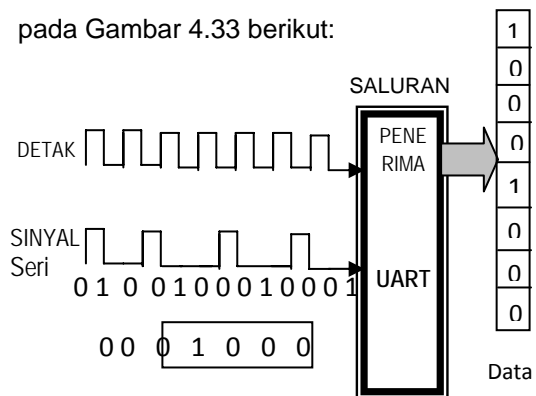
Sebaliknya deserialisasi dilakukan dengan membaca bit 0 dan merekamnya ke

akumulator. Akumulator di geser kekiri satu posisi dengan tunda tertentu. Kemudian bit 0 dibaca lagi dan dilakukan proses pencatatan dan penggeseran akumulator sampai data byte terselesaikan. Keuntungan I/O terprogram terletak pada ketersederhanaannya dan tidak perlu harus menyiapkan perangkat keras. Kelemahannya terletak pada masalah waktu yaitu lambatnya proses.

3.2. I/O Serial Perangkat Keras

Salah satu komponen LSI standar adalah Universal Asynchronous Receiver-Transmitter (UART). UART bekerja mengubah data serial ke paralel dan data paralel ke serial. UART paling sering digunakan untuk operasi kecepatan rendah ke sedang. Sedangkan untuk transmisi kecepatan tinggi digunakan jenis Universal Synchronous Receiver Transmitter (USRT).

Fungsi UART adalah sebagai pengubah serial-paralel. Prinsip pokok pengubahan serial ke paralel dilukiskan pada Gambar 4.33 berikut:



Gambar 4.33. Pengubahan Serial ke Paralel

Dua fungsi pokok UART adalah :

- Mengambil data paralel dan mengubah menjadi arus bit serial dengan diawali bit start, bit data, bit paritas, dan karakter penghenti.
- Mengambil arus bit serial dan mengubahnya menjadi bit paralel.

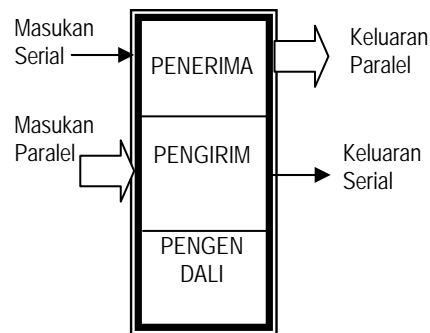
Sebuah UART standar mempunyai tiga seksi yaitu: sebuah penerima, sebuah pengirim, dan sebuah seksi pengendali. Gambar 4.34 menunjukkan diagram UART. UART memerlukan baik port masukan maupun port keluaran untuk perantara dengan sistem mikroprosesor. Dua diantara piranti UART adalah :

- MC 8650 adalah Asynchronous Communication Interface Adaptor (ACIA) dari Motorola.
- 8251 Universal Synchronous and Asynchronous Receiver Transmitter (USART) dari Intel.

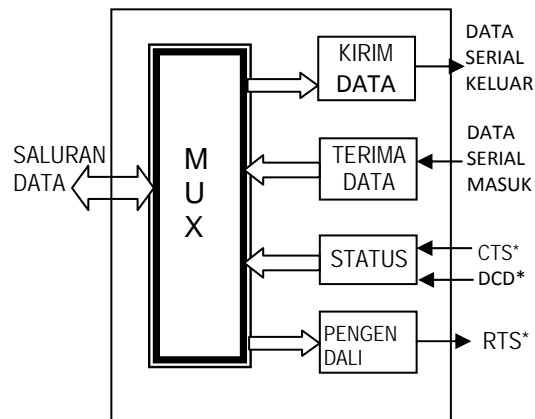
3.3. Motorola 6850 ACIA

6850 tersusun dari sejumlah register serial paralel masukan keluaran dan rangkaian pengendali standar EIA RS 232. Diagram blok ACIA digambarkan seperti Gambar 4.35. **Universal Asynchronous Receiver/Transmitter** (USART). 8251 menyediakan pasilitas pengiriman dan

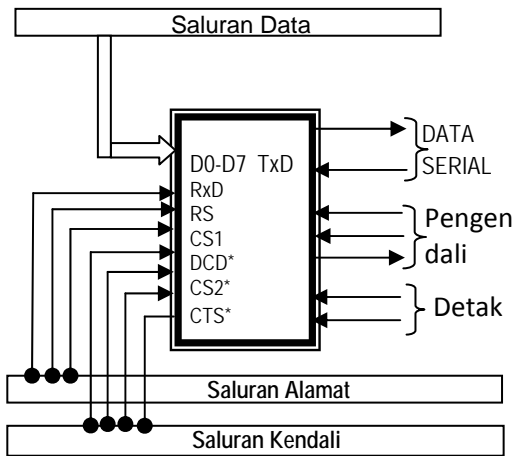
penerimaan data sinkron dan tak sinkron. Organisasi logika 8251 ditunjukkan pada Gambar 4.36. Penghantaraan 6850 pada saluran/bus dari sistem dapat digambarkan seperti Gambar 4.37. Data serial yang masuk dan keluar adalah sinyal kompatibel TTL dan harus di bufer untuk memberikan tingkatan yang diperlukan untuk menggerakkan alat-alat serial.



Gambar 4.34. Diagram UART



Gambar 4.35. Blok Diagram 6850 ACIA



Gambar 4.36. Bus 6850 ACIA

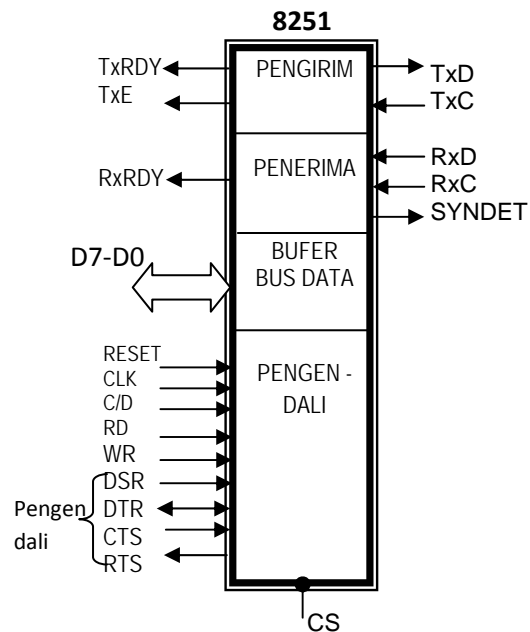
3.4. Intel 8251 USART

8251 dirancang oleh Intel yang memiliki fasilitas sebagai UART dan juga USRT. Dengan kata lain 8251 dapat dipakai baik sebagai alat tak serempak maupun alat serempak. Sehingga 8251 diberi nama USART.

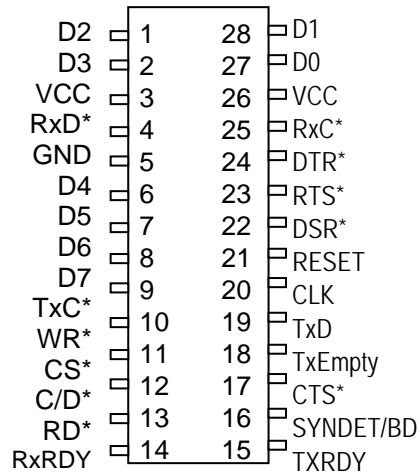
Dalam USART ada sebuah transmitter yang dapat mengubah dua data paralel dari dua sumber berbeda yaitu data paralel yang masuk melalui bus data D7–D0 menjadi data seri pada Tx/D, dan sebuah lagi berasal dari masukan receiver Rx/D yang diubah menjadi data paparel lalu diubah kembali menjadi data seri pada Tx/D.

Bagian lain adalah sekumpulan saluran kendali 8251 oleh CPU, dan bagian kendali Modem yaitu DSR*, DTR*, CTS*, dan RTS*.

Dalam *Transmitter Buffer* ada suatu register geser yang bekerja menggeser data paralel dari bus data menjadi data seri pada Tx/D. Pergeseran data terjadi setiap clock transmitter Tx/C* menjadi rendah. Jadi laju alih data (bit per second (bps)) dan laju pengisyarat Baud rate ditentukan oleh frekuensi clock transmitter pada Tx/C*. Pada transmisi asinkron baudrate dapat dipilih agar sama dengan satu kali, 1/16 kali atau 1/64 kali frekuensi clock pada Tx/C*. Pemilihan faktor perkalian ini dilakukan dengan mengisi bit-bit tertentu pada salah satu register di dalam 8251. Susunan logika dan susunan pin 8251 ditunjukkan pada Gambar 4.37.



Gambar 4.37a. Diagram Logika USART 8251



Gambar 4.37b. Susunan Pin USART 8251

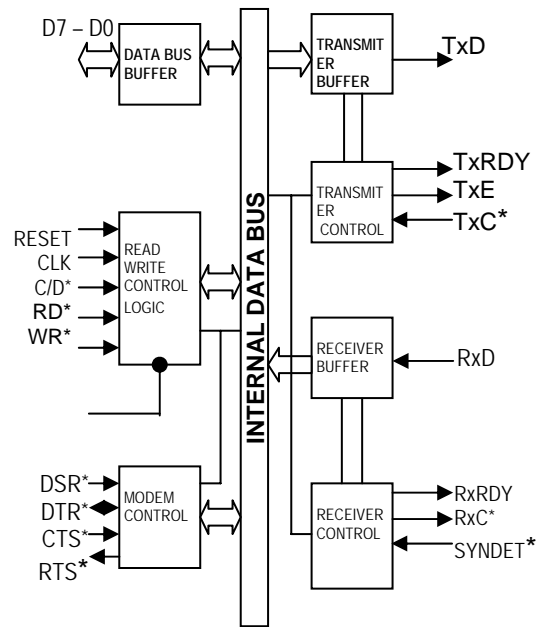
CPU mengirim data melalui bus data paralel, kemudian ditampung di dalam buffer data atau *buffer control*. Dari buffer ini data dimasukkan ke register geser. Tiap transisi rendah pada TxD* akan menggeser satu byte data keluar TxD.

Jika buffer data telah kosong maka 8251 akan memberi tahu kepada CPU dengan mengaktifkan TxRDY, yaitu membuat menjadi H (high), asalkan transmitter telah diaktifkan oleh keadaan rendah pada CTS* dan bit TxEn dalam register Instruksi Command dibuat aktif (bit1).

Isyarat keluaran TxRDY ini dapat digunakan untuk interupsi pada CPU jika data telah dimuat ke dalam register geser. Setelah CPU memuat data ke dalam buffer data TxRDY akan dibuat rendah.

Setelah dat selesai digeser keluar maka 8251 akan memberitahu hal ini dengan membuat TxE (*Transmitter Empty*) menjadi high.

Kerja bagian penerima adalah kebalikan dari bagian transmitter. Data seri yang masuk melalui RxD digeser masuk kedalam register geser dan setelah diubah menjadi data paralel dimuatkan kedalam buffer data untuk diambil oleh CPU melalui bus data. 8251 memberi tahu CPU dengan mengaktifkan RxRDY asalkan receiver telah diaktifkan oleh bit RxEn pada register kontrol instruksi. Arsitektur USART 8251 ditunjukkan pada Gambar 4.37c.



Gambar 4.37c. Arsitektur Internal USART 8251

Data masukan digeser oleh clock pada masukan RxC*. Pena SYDNET hanya digunakan pada modem sinkron. Pena-pena untuk kendali 8251 oleh CPU adalah sebagai berikut. Pena RESET digunakan untuk mengembalikan 8251 ke keadaan awal. Jika pena RESET dibuat high maka 8251 akan Idle atau tidak bekerja dan dalam keadaan reset yaitu kembali ke keadaan awal. 8251 akan bekerja kembali setelah mendapat program inisialisasi.

Pena Clock =CLK digunakan untuk pulsa pewaktuan di dalam 8251. Untuk operasi asinkron frekuensi CLK adalah 4,5 kali 3 frekuensi clock transmitter (TxC*) atau clock receiver (RxC*).

Selain itu perioda CLK harus mempunyai nilai antara 0,42 μ S dan 1,35 μ S. Pena CS* digunakan untuk mengaktifkan 8251 dengan memberi logika Low.

Pena C/D* (control/data) berfungsi yaitu pada operasi baca, jika pena ini dibuat tinggi, register status/control akan dialihkan ke bus data, sedang jika pena ini dibuat rendah maka register data akan dialihkan ke bus data. Pada operasi tulis jika pena ini tinggi maka isi bus data akan dialihkan ke register perintah/command atau register modus dalam 8251. Jika pena ini rendah maka isi bus data akan dialihkan ke register data untuk dikirim keluar menjadi data seri.

Pena WR* diaktifkan oleh CPU jika CPU mau menulis data pada 8251 dan pena RD* diaktifkan oleh CPU jika CPU hendak membaca data atau status pada 8251.

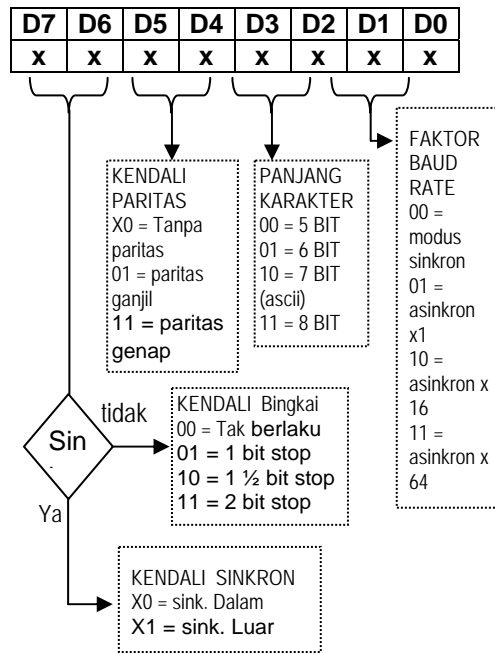
3.5. Pemrograman 8251

USART 8251 dapat bekerja dalam beberapa mode yang dapat dipilih dengan mengisi register-register di dalam 8251. Register-register tersebut yaitu:

1. Register Data
2. Register Instruksi Modus
3. Register Instruksi Perintah (Command)
4. Register Status

Register data dapat dibaca atau ditulisi oleh CPU. Register ini berisi data dari CPU yang akan diubah menjadi data seri pada keluaran TxD, atau data yang berasal dari masukan data seri RxD untuk dibaca oleh CPU. Register data dialamati dengan pena C/D* berlogika rendah (L).

Setelah terjadi RESET oleh pena RESET pada 8251 menjadi tinggi atau karena dilakukan reset internal melalui bit D6 pada register Instruksi, maka jika pena C/D* berkeadaan tinggi, maka bus data akan terhubung dengan register instruksi modus. Adapun modus mempunyai arti seperti Gambar 4.38.



Gambar 4.38 Modus Register Instruksi USART 8251

Sebagai contoh jika kita ingin 8251 bekerja dalam modus asinkron dengan baudrate 1/16 x frekuensi clock Tx_C, panjang karakter 7 bit, paritas ganjil, satu bit stop maka Register Instruksi akan berisi data seperti Gambar 4.39.

D7	D6	D5	D4	D3	D2	D1	D0	Data
0	1	0	1	1	0	1	0	5A

Gambar 4.39. Data modus asinkron

Maka : inisialisasi modus dapat dijalankan dengan perintah :

```
LD A, 5Ah
OUT Instruksi Modus, A
```

Bagaimana kita menentukan alamat I/O untuk instruksi Modus? Untuk menentukan alamat I/O dapat digunakan pengalaman seperti rangkaian Gambar 4.40. Dari pengkodean alamat ini maka terbangun alamat:

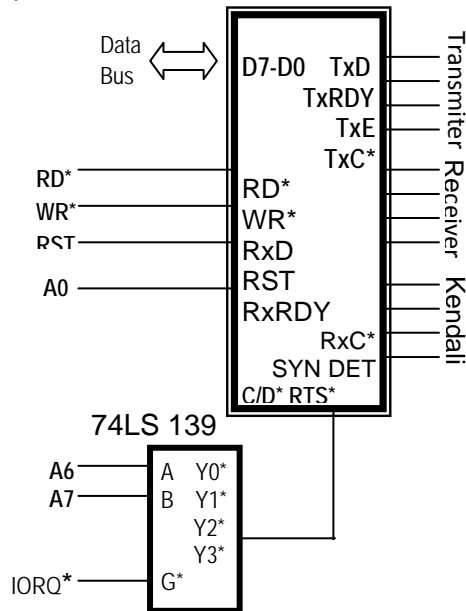
Alamat C0H untuk Data

Alamat C1H untuk Control

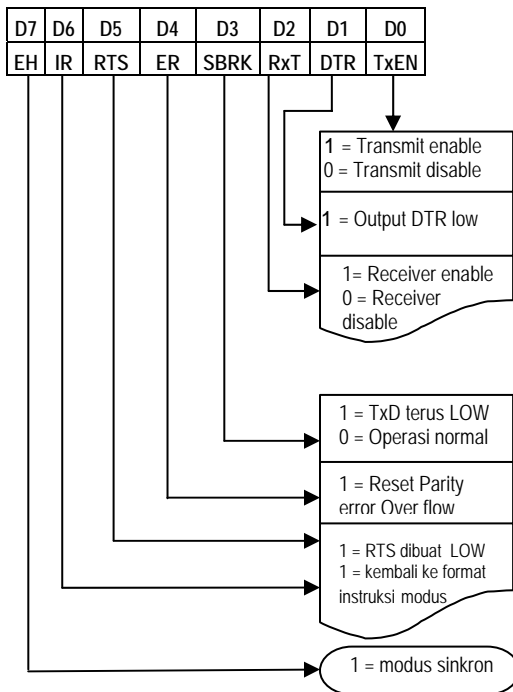
Jadi pengiriman modus dapat dijalankan dengan perintah :

```
LD A, 5Ah
OUT C1, A
```

Setelah register instruksi modus diisi data, maka pada port dengan alamat yang sama akan menghubungkan bus data dengan register instruksi (*Command Instruction Register*). Arti masing-masing bit seperti Gambar 4.41.



Gambar 4.40. Pengalamatan USART 8251



Gambar 4.41. Makna Bit Register Instruksi

Keterangan :

- TxEN : Transmit Enable
- DTR : Data Terminal Ready
- RxE : Receiver Enable
- SBRK : Send Break Character
- ER : Error Reset
- RTS : Request to Send
- IR : internal Reset
- EH : Enter Hunt Mode

Sebagai contoh andaikan akan membuat 8251 bekerja sebagai transmitter yaitu ingin mengirim data dari MPF-1 ke IBM PC melalui 8251. Untuk itu kita harus mengaktifkan TxEN dengan mengisi bit 1 pada D0, kita ingin membuat pena kendali modem RTS aktif (low) dengan membuat bit D5 = 1 dan kita ingin mereset flag error pada

register status dengan mengisi D4= 1. Untuk melaksanakan ini kita isikan data seperti Gambar 4.42.

0	0	1	1	0	0	0	1
EH	IR	RTS	ES	SBRK	RxE	DTR	TxEN

Gambar 4.42. Data register status

Caranya adalah sebagai berikut :

LD A, 31H

OUT C1, A

Pengiriman data selanjutnya ke Port alamat C1H akan merubah isi Register Instruksi. Juga bit D6 (IR) pada register instruksi. Register Instruksi diisi dengan dengan bit "1" maka lemparan data selanjutnya mengisi register instruksi modus kembali.

Status atau keadaan operasi 8251 dapat dibaca dengan membaca isi register status. Alamat register status sama dengan alamat register perintah, namun register status adalah register baca (Port masukan), sedangkan register Perintah adalah dua register tulis.

Isi register status dapat disalin ke Akumulator dengan instruksi:

IN A, C1H (alamat port 8251)

Bit-bit pada register adalah sebagai berikut:

D7	D6	D5	D4	D3	D2	D1	D0
DSR	SYNDET BRKDET	FE	OE	PE	TxE	RxRdy	TxRDY

Gambar 4.43. Data status 8251

- D0 TxRDY menyatakan bahwa isi buffer data telah dimuat ke dalam register geser untuk digeser keluar oleh pulsa-pulsa clock pada pena TxD. Bit ini dapat dipantau oleh CPU, sehingga buffer data tidak diisi oleh CPU sebelum proses serialisasi selesai. Dimuatnya isi buffer data ke dalam register geser juga ditandai dengan mengaktifkan pena TxRDY, namun yang terakhir ini hanya terjadi jika pena CTS* berlogika 0 dan bit TxEN pada register instruksi berisi bit 1. Keadaan pena TxRDY dapat digunakan membangun interupsi ke CPU.
- D1 RxRDY menyatakan bahwa data seri yang masuk melalui pena RxD telah selesai digeser dan dimuat ke buffer data, siap diambil oleh CPU. Untuk decoder alamat pada MPF-1 digunakan perintah
- IN A, C1H
- CP bit
- D2 TxEMPTY menyatakan bahwa register geser di dalam transmitter telah selesai menggeser data keluar ke pena TxD. Peristiwa ini ditandai dengan membuat bit D2 ini berisi 1, dan dapat dipantau
- D3 OE, Jika berisi 1 berarti telah terjadi salah paritas, yaitu 8251 tidak mendeteksi adanya bit paritas. Ini berarti terjadi kesalahan pada karakter yang diterima. Dengan memantau bit ini CPU dapat melompat ke subrutin tertentu jika terjadi salah paritas.
- D4 OE; menyatakan **Overrun Error**, yaitu buffer data tak keburu dibaca sudah keluar data baru dari masukan seri RxD.
- D5 FE; (**Framming Error**) menyatakan 8251 tidak mendeteksi bit stop, sehingga tidak dapat menentukan mulainya byte baru. Flag-flag error D3,D4, dan D5 dapat direset dengan mengisi bit ER pada register instruksi dengan bit 1.
- D6 SYN-DET digunakan pada modus transisi sinkron
- D7 DSR ; mencerminkan keadaan pena-pena masukan DSR (*Data Set Ready*)
- oleh CPU, atau menyebabkan pena TxE aktif dapat digunakan untuk pemberian interupsi pada CPU.

4. Metoda Pengendalian I/O

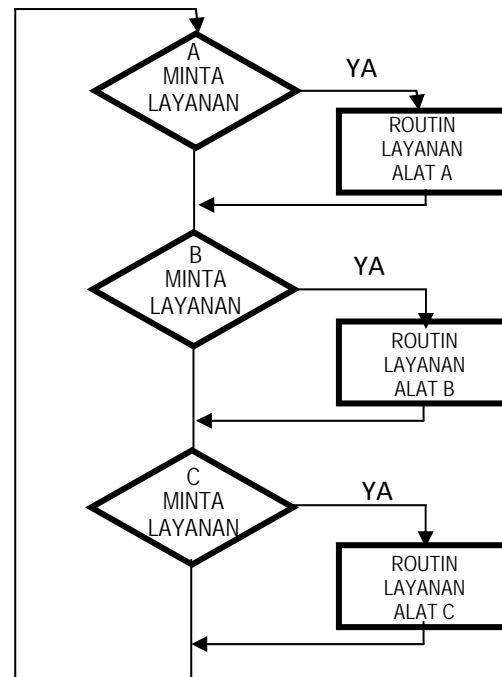
Pengaturan alih data dari alat luar dengan sistem komputer/sistem mikroprosesor menerapkan suatu strategi penjadwalan. Pada pengendalian alat I/O dikenal adanya tiga metoda yaitu :

- **Metoda Polling**
- **Metoda Interupsi**
- **Metoda Akses Memori Langsung (AML) /Direct Memory Acces (DMA)**

4.1. Metoda Polling

Metoda polling merupakan metoda pengendalian I/O melalui program. Semua pengalihan data dari dan ke alat I/O diselenggarakan oleh program. Prosesor mengirim dan meminta data sepenuhnya dibawah kendali program. Pengalihan data dapat dilaksanakan baik melalui mekanisme jabat tangan maupun tanpa jabat tangan. Dalam mekanisme jabat tangan isyarat diperiksa secara terus menerus. Program terus menerus berputar lewat sejumlah pengetesan untuk menentukan apakah masukan atau keluaran dapat diselenggarakan pelayanannya atau tidak. Bila ditemukan alat yang memerlukan pelayanan, rutin pelayanan diaktifkan dan pemilihan saluran diproses. Gambar 4.44 menunjukkan diagram alir pengendalian I/O dengan metoda polling. Metoda polling

adalah metoda pengendalian I/O yang paling sederhana dan paling umum digunakan. Metoda ini tidak memerlukan perangkat keras khusus dan semua pengalihan I/O dikendalikan oleh program. Pengalihan semacam ini disebut pengalihan serempak dengan program.



Gambar 4.44. Diagram Alir Pengendalian I/O Sistem Polling

4.2. Metoda Interupsi

Pengendalian I/O dengan metoda polling mempunyai dua kelemahan :

- Pemborosan waktu prosesor karena status semua periferal diperiksa terus menerus secara berurutan.

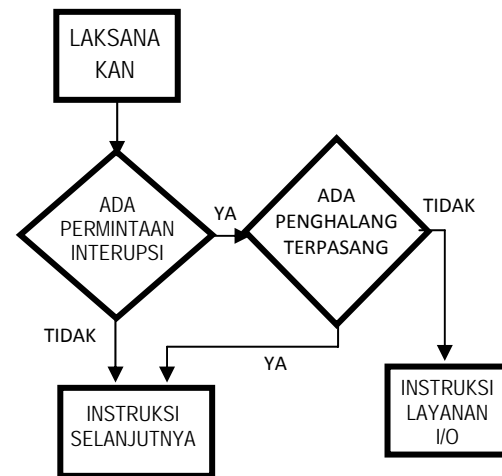
- Karena harus memeriksa status semua alat I/O maka waktu kerjanya menjadi lambat. Ini merupakan kelemahan dalam sistem waktu nyata (Real Time), dimana satu periferal mengharap layanan dalam satu waktu tertentu.

Kelemahan ini diatasi dengan menggunakan layanan waktu tak sinkron menggunakan interupsi. Tiap alat I/O atau pengendalinya dihubungkan ke sebuah saluran interupsi. Saluran interupsi menggerbangkan sebuah permintaan interupsi ke mikroprosesor. Bilamana sebuah alat I/O memerlukan layanan, alat akan membangkitkan pulsa interupsi atau status suatu tingkatan saluran untuk menarik perhatian mikroprosesor. Mikroprosesor akan memberikan layanan pada alat I/O jika ada interupsi dan jika tidak ada interupsi mikroprosesor melakukan instruksi selanjutnya. Logika pengendalian I/O dengan metoda interupsi ditunjukkan pada diagram alir Gambar 4.45.

Begitu permintaan interupsi diterima dan disetujui oleh mikroprosesor, alat I/O harus dilayani. Untuk melayani alat I/O, maka mikroprosesor melaksanakan suatu rutin pelayanan khusus. Ada dua masalah yang muncul pada saat melakukan layanan interupsi :

- Bagaimana status program yang dilaksanakan pada mikroprosesor pada saat interupsi harus diperilakaha dalam *stack*.

- Bagaimana mikroprosesor dapat mengenali secara tepat alat I/O mana yang membangkitkan interupsi. Identifikasi ini dapat dilakukan dengan perangkat keras, perangkat lunak, atau kombinasi perangkat keras dan perangkat lunak. Pencabangan ke alamat alat I/O disebut



Gambar 4.45. Diagram Alir Logika Pengendalian I/O Metoda Interupsi

• Pemvektoran Interupsi

Rutin perangkat lunak menetapkan identitas alat yang meminta layanan interupsi. Rutin identifikasi interupsi akan memilih saluran setiap alat yang dihubungkan dengan sistem. Setelah dikenal alat mana yang mencetuskan interupsi maka ia kemudian bercabang ke alamat rutin penanganan interupsi yang sesuai.

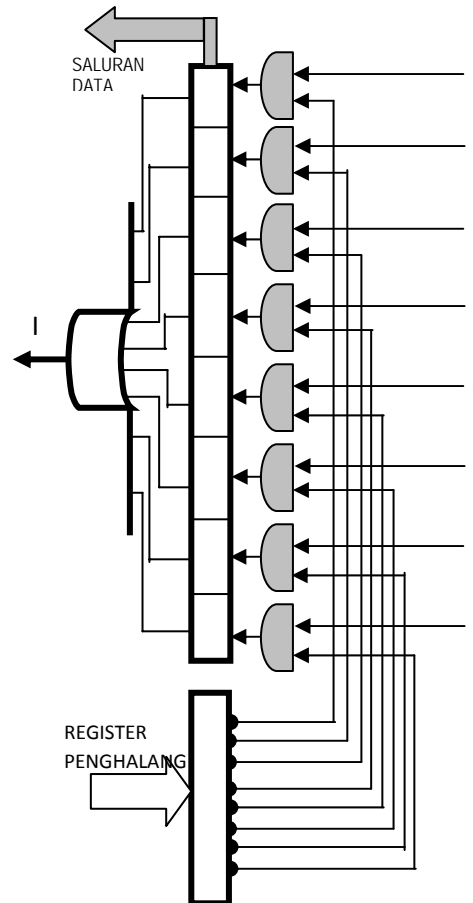
Metoda ke dua yang digerakkan oleh perangkat lunak, tetapi dengan pertolongan beberapa perangkat keras tambahan. Metoda ini menggunakan rantai beranting (*daisy chain*) untuk mengenal alat yang mencetuskan interupsi.

Metoda tercepat adalah interupsi yang divektorkan. Adalah menjadi tanggung jawab pengendali alat I/O untuk memberikan baik interupsi maupun pengenalan alat yang menyebabkan interupsi atau lebih baik lagi alamat pencabangan bagi rutin penanganan interupsi. Bila pengendali hanya memberikan pengenalan alat, adalah tugas perangkat lunak mencari tabel alamat pencabangan bagi tiap alat. Ini sederhana bagi perangkat keras tapi tak mencapai performansi tertinggi.

- **Prioritas**

Beberapa interupsi dapat dibangkitkan serentak. Mikroprosesor diberi tugas untuk memutuskan bagaimana urutan pelayanannya. Setiap alat diberikan suatu prioritas. Mikroprosesor melayani setiap alat sesuai prioritasnya. Dalam dunia komputer prioritas 0, menurut konvensi memiliki prioritas tertinggi, prioritas 1 yang kedua demikian seterusnya. Prioritas dapat diset baik pada perangkat keras maupun perangkat lunak. Pengaturan prioritas

dengan perangkat keras dikerjakan oleh suatu piranti yang disebut Programmable Interrupt Controller (PIC). Struktur dasar logika PIC dapat digambarkan seperti Gambar 4.46.



Gambar 4.46. Struktur Dasar Logika PIC

4.3. Interupsi pada Z-80 CPU

Ada dua jenis interupsi pada mikroprosesor Z-80 CPU yaitu :

- ❑ Non Maskable Interrupt (NMI) : yaitu interupsi yang sama sekali tidak dapat dihalangi. Tak peduli apakah CPU dalam keadaan dapat di interupsi atau tidak CPU akan melayani dan mencabang se subruitin yang berada pada lokasi alamat **0066H**. NMI digunakan pada keadaan darurat dan imergensi.
- ❑ Maskable Interrupt (INT) adalah interupsi yang dibuat aktif rendah oleh alat luar yang minta layanan interupsi. Dalam hal ini CPU akan melayani jika ia diset dalam keadaan Enable Interupt, sedangkan CPU tidak akan melayani jika dalam keadaan disable interrupt.

Jika pena INT pada CPU berlogika rendah, ini pertanda ada alat luar yang meminta layanan interupsi. Jika CPU ada dalam keadaan dapat diinterupsi, maka CPU akan menyelesaikan proses yang sedang diolah dan selanjutnya memasuki proses layanan interupsi.

CPU dapat dibuat peka terhadap interupsi dengan memasalng perintah EI (Enable Interupt) di dalam program atau dibuat tidak peka terhadap interupsi dengan memberi perintah DI (Disable Interupt) pada program.

Mikroprosesor Z-80 CPU memiliki tiga modus Interupsi yaitu:

- ❑ Modus Intrupsi Nol (IM0)
- ❑ Modus Interupsi Satu (IM1)
- ❑ Modus interupsi Dua (IM2).

Modus interupsi nol (IM0) bekerja dimana jika pena INT* aktif rendah, CPU akan membuat pena M1 rendah sebagai pertanda CPU mengambil op-code, dan disertai dengan aktifnya pena IORQ* yang berarti Op-code tidak diambil dari memori tapi diambil dari I/O melalui bus data. Kedua isyarat ini dapat di OR kan ($M1^* = 0$ dan $IORQ^*=0$ maka output $OR= 0$), untuk membentuk isyarat INTA*= Interupt Acknowledge yang aktif rendah. Isyarat INTA* digunakan untuk memasukkan data ke CPU melalui bus data. Data yang masuk akan diteruskan ke Register Instruksi untuk dibuka sandinya.

Jika yang masuk adalah sandi Heksadesimal untuk instruksi RST n, maka CPU akan melaksanakan CALL subrutin yang bermula dari lokasi $(n \times 8)_{10}$.

Sandi biner untuk instruksi RST n adalah seperti Gambar 4.47.

RST n (biner)	D7	D6	D5	D4	D3	D2	D1	D0	Op- code (H)	RST (DES)	RST (HEX)
RST 0	1	1	0	0	0	1	1	1	C7	0	0
RST 1	1	1	0	0	1	1	1	1	CF	8	8
RST 2	1	1	0	1	0	1	1	1	D7	16	10
RST 3	1	1	0	1	1	1	1	1	DF	24	18
RST 4	1	1	1	0	0	1	1	1	E7	32	20
RST 5	1	1	1	0	1	1	1	1	EF	40	28
RST 6	1	1	1	1	0	1	1	1	F7	48	30
RST 7	1	1	1	1	1	1	1	1	FF	56	38

Gambar 4.47. Sandi biner instruksi RST n

Pada waktu INTA* aktif dan data yang masuk ke CPU adalah DFH akan diterjemahkan sebagai instruksi RST 18H atau RST 24D, yang akan membuat CPU menyambung ke subrutin lokasi 0018H. Jika data yang masuk misalnya FFH akan membuat CPU menyambung ke subrutin lokasi 0038H.

Daerah memori antara 0000 s/d 00FFh disebut daerah halaman nol atau page zero, sehingga RST n dikatakan memvektor ke Page Zero.

Pada MPF-1, Modus Interupsi nol IM0 tidak dapat digunakan oleh pemakai sebab daerah vector RST n didukui oleh ROM monitor. Dalam hal ini jika menggunakan interupsi harus memilih modus yang lain yaitu IM1.

Modus IM1 menggunakan register I yang diisi melalui register A. Misalnya kita ingin

pelayanan interupsi memvektor ke lokasi **1820H**. Untuk itu lokasi alamat 1820 ini harus disimpan di dua lokasi berurutan yaitu misalnya lokasi alamat **18A0H dan 18A1H**. Byte bawah vector (20H) disimpan pada lokasi 18A0 dan byte atas 18H disimpan pada lokasi alamat 18A1 selanjutnya register I diisi data 18 H sebagai berikut :

```
LD A, 18H
LD I, A
```

Selanjutnya byte bawah lokasi tempat simpan alamat vector akan diambil melalui bus data dengan menggunakan pulsa INTA*. Programnya adalah:

```
Mulai IM1 ; Pasang modus IM1
LD A, 18H ; A = 18
LD I, A ; Byte atas masuk ke I
LD SP, STACK
EI
SIN1 JP SIN1
ORG 18A0
DEFB A0H
DEFB 18H
```

4.4. Akses Memori Langsung

Interupsi menjamin tanggapan yang paling cepat dari proses pengendalian data pada I/O. Akan tetapi pelayanan pada alat masih diselenggarakan oleh perangkat lunak. Kecepatan transfer paralel sebuah mikroprosesor dibatasi oleh overhead perangkat lunak yang terlibat dalam pengiriman kata-kata berurutan. Ini mungkin masih tidak cukup cepat bagi pengolahan yang melibatkan alih memori cepat. Kembali

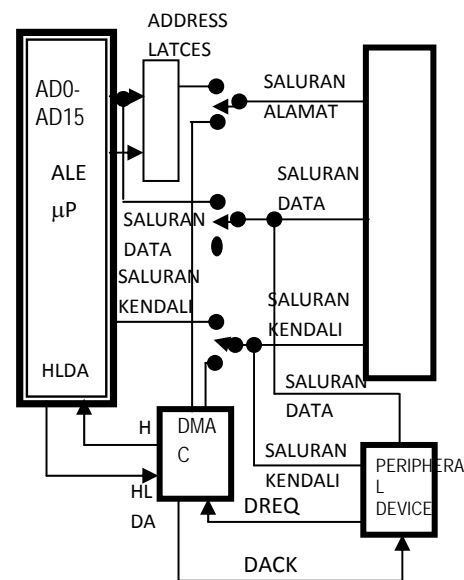
disini menggantikan perangkat lunak dengan perangkat keras. Rutin perangkat lunak yang menyelenggarakan alih data antara memori dengan alat I/O digantikan oleh prosesor perangkat keras khusus yang disebut dengan Direct Memory Access Controller (DMAC). Sebuah DMAC adalah prosesor khusus yang dirancang untuk menyelenggarakan alih data berkecepatan tinggi antara memori dengan alat luar. Dalam akses memori langsung digunakan dua teknik untuk berhubungan dengan memori :

- Prosesor dihentikan atau ditangguhkan oleh DMAC. DMAC memegang pengendalian bus dan membiarkan alat I/O berhubungan langsung dengan memori.
- DMAC mencuri satu siklus memori dari mikroprosesor, memberinya kepada pengiriman data antara memori dan alat I/O.

DMAC adalah prosesor khusus yang memutuskan hubungan atau mengisolasi MPU dari bus-bus dan mengatur pengiriman yang diperlukan antara memori dan alat I/O. Gambar 4.48 menunjukkan diagram blok kerja DMAC.

Pada saat sistem bekerja, saklar pada posisi atas sehingga saluran terhubung dari mikroprosesor ke sistem memori dan peripheral. Untuk membaca file ke disk

diperlukan sejumlah perintah ke disk controller, memerintahkan untuk mencari dan membaca blok data yang dari disk. Jika disk controller telah menemukan byte pertama dari blok data, disk controller mengirim sinyal DMA request (DREQ) ke DMAC. Jika DMAC tidak dalam terhalang maka DMAC mengirim sinyal hold request (HRQ) ke mikroprosesor melalui pin HOLD. Mikroprosesor menanggapi masukan ini dengan mengambangkan saluran/bus dan mengirim sinyal hold acknowledge (HLDA) ke DMAC. Jika DMAC menerima sinyal HLDA, akan mengirim sinyal untuk menghubungkan bus/saluran ke posisi DMAC.



Gambar 4.48. Blok Diagram Kerja DMAC

Pada saat DMAC mengontrol saluran, ia mengirim alamat memori dimana byte pertama dari disk controller di tulis. Selanjutnya DMAC mengirim sinyal DMA acknowledge (DACK) ke disk controller untuk memberitahukan kesiapan mengeluarkan byte. Akhirnya DMAC mengaktifkan saluran MEMW* dan IOR* pada saluran kontrol.