



KOMPONEN *INTERFACING*

Yoyo somantri
Dosen Jurusan Pendidikan Teknik Elektro
FPTK Universitas Pendidikan Indonesia

Pendahuluan

Dalam bab ini akan dibahas tujuan perkuliahan, komponen – komponen input/output (IO) yang banyak digunakan pada mikroprosesor Z80, 8085, 8080, 8088/86, dan mikroprosesor 6800. Komponen-komponen itu antara lain: Z80 PIO, PPI 8255, CTC Z 80, SIO Z 80, ACIA, USART, dan PIA. Komponen input-output ini sebagai pendukung dalam sistem mikroprosesor atau mikrokomputer.

Tujuan Perkuliahan

Setelah mempelajari bab ini, diharapkan mahasiswa mampu untuk :

1. Memahami macam – macam komponen I/O yang terdapat dalam sistem mikroprosesor atau komputer.
2. Memahami karakteristik setiap komponen input/output
3. Memahami konfigurasi dan fungsi dari komponen I/O yang dipergunakan pada mikroprosesor atau komputer.
4. Mengaplikasikan komponen I/O pada sistem mikroprosesor atau komputer untuk aplikasi tertentu.

Suatu sistem mikroprosesor akan bekerja dengan baik apabila semua komponen pendukungnya telah terpasang dengan baik dan benar pada sistem mikroprosesor atau mikrokomputer. Salah satu komponen penting tersebut yaitu peripheral input dan output. Pada tabel di bawah ini ditunjukkan contoh *peripheral I/O*.

Tabel.1. Peripheral I/O

Zilog	Z80-DART; Z80-SIO	Z80-PIO
Intel	USART = (8251)	PPI = (8255)
Motorola	ACIA = (6850)	PIA = (6821)

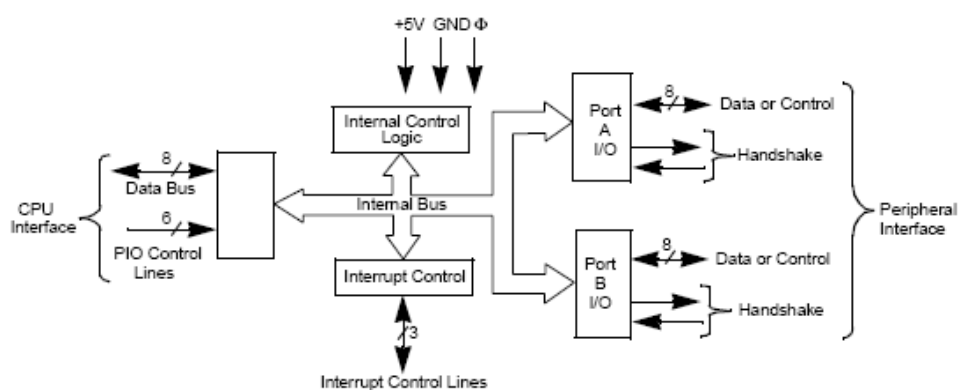
1. Input/Output Paralel

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

IC Z-80 PIO adalah IC I/O paralel terprogram yang prilakunya dapat diset menggunakan program. Z-80 PIO merupakan salah satu *chip* yang diproduksi untuk fasilitas antar muka dengan Z-80 CPU. Z-80 PIO memiliki kelengkapan :

- 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.

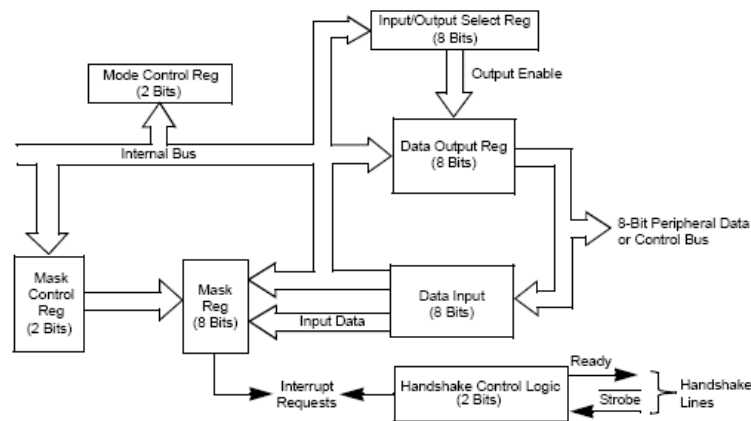
Diagram blok Z80 PIO diperlihatkan pada gambar di bawah ini. Struktur dalam Z80 PIO berisi bus antar muka, kontrol logic internal, Port A I/O, Port B I/O dan kontrol interupsi. Z80 PIO secara langsung dapat dihubungkan dengan CPU Z80 tanpa peripheral eksternal. Suatu *address decoder* akan dibutuhkan jika sistemnya cukup besar.



Gambar 1. Diagram blok PIO
(Sumber dari data book Zilog 1986)

Port I/O terdiri dari 6 *register* dengan jabat tangan, seperti diperlihatkan pada gambar 2. Keenam *register* tersebut yaitu *register* 8 bit input data, *register* 8 bit

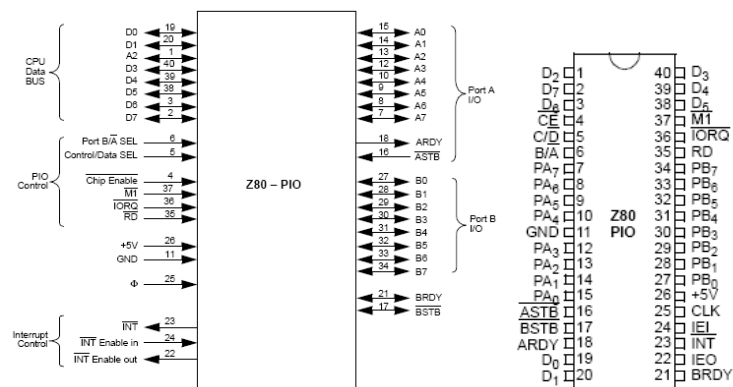
output data, *register 2 bit control mode*, *register 8 bit untuk pemilihan 6 bit input atau output dan register 2 bit mask control*.



Gambar 2. Diagram blok port PIO
(Sumber dari data book Zilog.1986)

1.1.1. Deskripsi pin Z80 PIO

Z80 PIO terdiri dari 40 pena, seperti diperlihatkan pada gambar 3



Gambar 3. Konfigurasi pin PIO
(Sumber dari data book Zilog.1986)

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 :

1. Kelompok Bus Data

- a. D0 – D7 adalah bus data 8 bit dua arah digunakan sebagai saluran data dan kata perintah.

- b. A0 – A7 adalah saluran dua arah untuk transfer data atau status dan sinyal kontrol antara peralatan I/O dan Port A.
- c. B0 – B7 merupakan saluran dua arah untuk transfer data atau status dan sinyal kontrol antara I/O dan Port B.

2. Kelompok Kontrol

- a. 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).
- b. 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.
- 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, 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*.
- 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.

3. Kelompok Interrupt

- a. INT* adalah sinyal interrupt aktif rendah yang digunakan oleh PIO untuk meminta layanan interupsi.
- b. IEI adalah sinyal Interrupt Enable Input aktif tinggi yang menunjukkan PIO siap menerima layanan interupsi.
- c. IEO adalah sinyal *Interrupt Enable Output* aktif tinggi yang menunjukkan PIO telah melayani interupsi.

4. Kelompok Status Kontrol *Port*

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

- Mode 0 : menunjukkan keadaan peralatan I/O telah menerima data yang dikirim oleh PIO.
 - Mode 1 : menunjukkan keadaan data telah dikirim ke *register Port A* oleh peralatan I/O.
 - Mode 2 : menunjukkan keadaan data dari *register Port A* telah diletakkan pada bus data dan kemudian data telah diterima oleh peralatan I/O.
 - Mode 3 : pulsa ini secara internal ditahan oleh PIO (tidak dimanfaatkan).
- b. A RDY adalah sinyal *ready* aktif tinggi untuk *Port A* bekerja tergantung mode operasi sebagai berikut :
- Mode 0 : menunjukkan register *Port A* berisi data byte dan telah disiapkan pada saluran bus data untuk ditransfer ke peralatan I/O.
 - Mode 1 : menunjukkan keadaan *register data Port A* kosong dan siap menerima data *word* berikutnya.
 - Mode 2 : menunjukkan keadaan *register data Port A* telah siap untuk diambil oleh peralatan I/O. Data akan dikeluarkan jika ada sinyal STB*.
 - Mode 3 : tidak dimanfaatkan
- c. B STB* adalah sinyal masukan *strobe* untuk Port B aktif rendah dimana operasinya sama dengan sinyal A STB
- d. B RDY adalah sinyal keluaran *ready* aktif tinggi untuk Port B dengan operasi kerja sama dengan A RDY.0

1.1.2. Pemrograman Z 80 PIO

RESET

Keadaan *reset* memiliki fungsi :

1. Kedua *port register mask reset* untuk menghalangi semua *port* bit data.
2. *Port bus data* di set pada keadaan impedansi tinggi dan sinyal jabat tangan tidak aktif maka secara otomatis mode 1 dipilih.
3. *Register* alamat vektor tidak direset.
4. Kedua *port interrupt enable flip – flop* di reset
5. Kedua *port register output* direset.

Pada *power-on reset* otomatis, PIO dapat direset dengan mengaplikasikan sinyal M1 tanpa menghadirkan sinyal RD atau IORQ. Jika dideteksi tidak ada RD atau IORQ selama M1, PIO akan segera memasuki keadaan reset setelah sinyal M1 menuju tidak aktif.

Pengisian Interupsi vektor

PIO dirancang untuk dapat dioperasikan dengan Z80 CPU menggunakan mode 2 tanggapan interupsi. Mode ini memerlukan sebuah interupsi vektor yang diberikan oleh perangkat interupsi. Vektor ini digunakan oleh CPU untuk membentuk alamat selama pelayanan *routine* pada *port* itu. Vektor ini ditempatkan pada data bus Z80 selama sebuah siklus interupsi *acknowledge* dengan prioritas tinggi. Interupsi vektor diisikan ke dalam PIO dengan menuliskan kode *word* dengan format data sebagai berikut :

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

Pemilihan mode operasi

Port A pada PIO dapat dioperasikan dalam 4 *mode* : *mode 0 (mode output)*, *mode 1 (mode input)*, *mode 2 (mode bidirectional)* dan *mode 3 (mode kontrol)*. *Port B* dapat dioperasikan juga untuk mode ini kecuali *mode 2*. format kontrol *word* yang harus diisikan pada PIO adalah :

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1
<i>Mode Word</i>		<i>X=Unused Bit</i>		<i>Signifies Modes Word To Be Set</i>			
D7		D6		Mode			
0		0		0 (<i>output</i>)			
0		1		1 (<i>input</i>)			
1		0		2 (<i>bidirectional</i>)			
1		1		3 (<i>control</i>)			

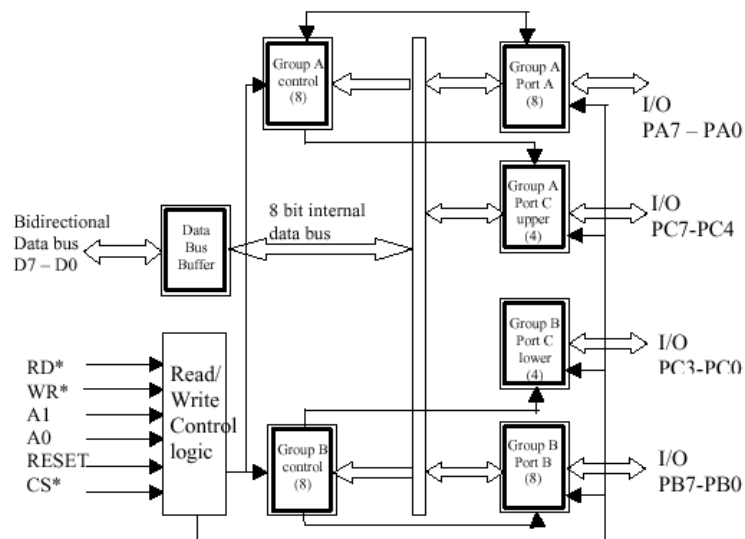
Bit D5 dan D4 diabaikan. Bit D3-D0 harus di set 1111 untuk menunjukkan *set mode*.

1. 2. PPI 8255

Peripheral Programmable Interface (PPI) 8255 adalah suatu komponen I/O yang dapat diprogram. Komponen ini dikemas dalam DIP 40 pin, 24 pin diantaranya adalah pin yang dapat diprogram.

1.2. 1. Diagram Blok PPI 8255

Pada gambar 4 di bawah ini dapat dilihat diagram blok dari PPI 8255. pada gambar tersebut dapat dilihat ada 4 blok yang dihubungkan ke saluran dari *peripheral input output* (PIO) yang dapat dihubungkan dengan peralatan luar. Saluran ini diberi nama PA0-PA7, PB0-PB7 dan PC0-PC7.



Gambar 4. Diagram Blok PPI 8255
(Sumber dari data book Intel 1985)

Kelompok-kelompok sinyal ini dibagi atas tiga *port* I/O yang berbeda, yaitu *port* A (PA), *port* B (PB) dan *port* C (PC). Ketiga *port* ini dapat diprogram sebagai *input* atau *output*. Tiap *port* mempunyai 2 blok I/O yang terpisah dan tiap blok dihubungkan ke bus data internal PPI 8255. melalui bus data internal informasi dapat dikirim dan diterima pada PPI 8255. PPI 8255 mempunyai 2 blok pengontrol, yaitu pengontrol group A dan pengontrol group B yang menentukan bagaimana ketiga *port* I/O dioperasikan (ada beberapa *mode* operasi yang berbeda untuk PPI 8255 dan *mode* ini harus ditentukan oleh program penulisan CPU atau “*control word*” dari *device*). Group C dari PPI 8255 terdiri dari 2 *port* 4 bit. Salah satu dari group ini digunakan bersama-sama dengan group A dan yang lainnya dengan group B.

Penyangga bus data dan pengontrol *read/write* memberikan *interface* antara mikroprosesor Z80 dan PPI 8255. penyangga bus data menyangga saluran input data dan saluran *output buffer* dari bus data CPU. Pengontrol *read/ write* mengirim data ke dan dari *register* internal yang sesuai.

1.2. 2. Konfigurasi Pin PPI 8255

Konfigurasi pin dari PPI 8255 dapat dilihat pada gambar 5

D0 – D7 adalah saluran data input dan output. Semua informasi yang akan ditulis ke atau dibaca dari PPI 8255 melalui saluran data ini.

\overline{cs} (*chip select input*). Bila saluran ini berlogika nol, mikroprosesor dapat membaca dan menulis ke PPI 8255.

\overline{RD} (*Read Input*). Bila saluran ini berlogika 0 dan input CS berlogika 0, data output dari PPI 8255 di enable ke bus data sistem.

WR (*Write input*). Bila saluran input ini berlogika 0 dan input cs berlogika 0, data dari *bus data system* ditulis ke PPI 8255.

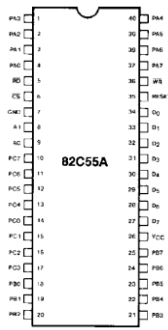
A0 – A1 (*address input*). Kombinasi logika dari 2 saluran input ini menentukan operasi pembacaan atau penulisan data dari *register* dalam PPI 8255.

RESET. Bila saluran data input ini berlogika 1, PPI 8255 berada dalam keadaan *reset*. Semua *port* peripheral di *set* ke *mode input*.

PA0 – PA7. saluran ini digunakan sebagai port I/O 8 bit yang dapat dihubungkan ke peralatan peripheral.

PB0 – PB7. saluran ini digunakan sebagai port I/O 8 bit yang dapat dihubungkan ke peralatan peripheral.

PC0 – PC7. saluran ini digunakan sebagai *port* I/O 8 bit yang dapat dihubungkan ke peralatan peripheral. Sinyal dari kelompok ini dibagi atas 2 kelompok 4 bit dan digunakan untuk mengontrol saluran PB0 – PB7 dan saluran PA0 – PA7 dalam beberapa mode operasi PPI 8255.



Gambar 5. Konfigurasi pin PPI 8255
(Sumber dari data book Intel 1985)

1.2.3. Pengalamatan PPI 8255

PPI 8255 terdapat 4 *register internal* yaitu *register read* dan *write*. *Register-register* tersebut dapat didefinisikan sebagai *register read* atau *write* dengan cara sebagai berikut :

Table 2. Pengalamatan PPI 8255

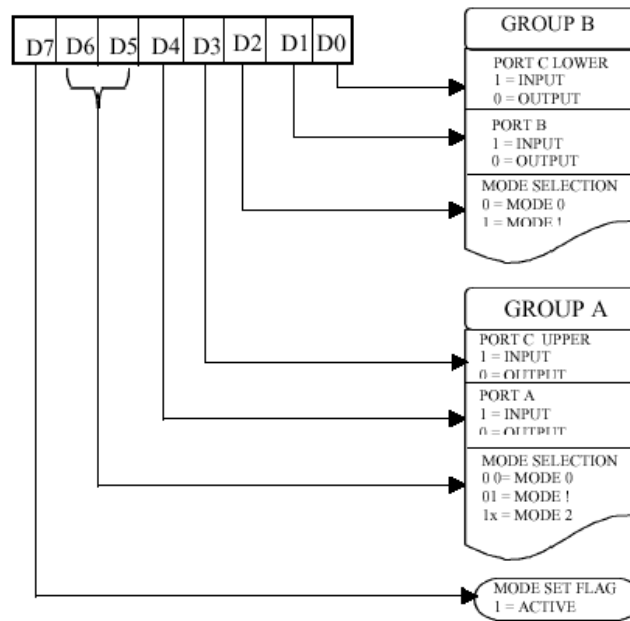
PIN				NAMA REGISTER
RD	WR	A1	A0	
1	0	0	0	<i>Port A</i> untuk penulisan data
0	1	0	0	<i>Port A</i> untuk pembacaan data
1	0	0	1	<i>Port B</i> untuk penulisan data
0	1	0	1	<i>Port B</i> untuk pembacaan data
1	0	1	0	<i>Port C</i> untuk penulisan data
0	1	1	0	<i>Port C</i> untuk pembacaan data
1	0	1	1	Penulisan <i>control word</i>
0	1	1	1	Pembacaan <i>register</i> tidak benar (illegal)

1.2.4. Inisialisasi PPI 8255

Inisialisasi adalah cara memprogram port PPI 8255 sehingga *port* tersebut berfungsi sebagai *input* atau *output*. Cara menginisialisasi komponen PPI 8255 dapat dilakukan dengan memasukan data dari CPU ke *control word* PPI 8255. *control word* mempunyai 8 bit yang formatnya sebagai berikut :

Bit D7 digunakan untuk mendefinisikan bit ini sebagai *control word*.

Bit D6 dan D5 digunakan untuk mendefinisikan mode operasi pada *port A*.



Gambar 6. Definisi bit-bit dari register control PPI 8255

Bit D4 digunakan untuk mendefinisikan *port A* sebagai *input* atau *output*.

Bit D3 digunakan untuk mendefinisikan 4 bit orde tinggi dari *port C* sebagai *input* atau *output*.

Bit D2 digunakan untuk mendefinisikan mode operasi pada *port B*.

Bit D1 digunakan untuk mendefinisikan *port B* sebagai *input* atau *output*.

Bit D0 digunakan untuk mendefinisikan 4 bit orde rendah dari *port C* sebagai *input* atau *output*.

Untuk inisialisasi digunakan mode 0.

Di dalam *control word* dapat ditentukan mode operasi yang diinginkan. Mode operasi ini ada 3, yaitu :

1. Mode 0 – *basic I/O*.

Pada mode ini tiap group dapat dikontrol sebagai input atau output tanpa kontrol *handshaking*.

2. Mode 1

Pada mode ini saluran I/O dibagi atas group A dan dapat diprogram sebagai input atau output dengan *port C (upper)* sebagai saluran kontrol *handshaking* untuk *port A* dan *port C (lower)* sebagai saluran kontrol *handshaking* untuk *port B*.

3. Mode 2

Pada mode ini *device* dapat menggunakan *port A* sebagai *bidirectional data port* yaitu 8 saluran port A dapat mengirim data ke dan dari alat luar.

Berikut ini contoh inisialisasi PPI 8255.

Contoh : buatlah program inisialisasi sehingga *port A* sebagai input, *port B* sebagai output dan 4 bit orde tinggi port C sebagai input dan 4 bit orde rendah port C sebagai output.

```
LD A, 98H
OUT (PPI 8255), A
```

Berdasarkan contoh di atas, control word akan berisi data sebagai berikut :

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

Pada instruksi OUT (PPI8255), A → (PPI 8255) adalah alamat kontrol dari PPI 8255.

2. Referensi :

1. Laventhal, (1986). *Z80 Assembly Language Programming*, Mc Graw Hill, Singapore.
2. Hall, (1985), *Microprocessor and Digital Systems*, Mc Graw Hill.
3. Rodney Zaks and Austin Lesea. (1979). *Microprocessor Interfacing Techniques*. Sybex Inc.
4. Hartono Partoharsodjo. (1990).*Bahasa Assembly*. Jakarta : PT. Elek Media Komputindo.
5. James W. Coffron. (1983). *Practical Hardware Details For 8080,Z80,and 6800*.
6. Inelco, (1986). *Guru Mikro Saya*.
7. Brey, Barry B. (2003). *The intel microprocessors : 8086/8088/80186/80286/80386/80486, Pentium, Pentium Pro processor, Pentium II, Pentium III, and Pentium 4: architecture, programming, and Interfacing-* 6 th ed. New Jersey : Pearson Education.
8. Greenfield, Joseph D.(1992). *The 68HC11 Microcontroller*.Orlando, FL:
9. Puadi. (1995). *Upaya Pengembangan Kegunaan TrainerBGC 8088 V3 Sebagai Alat Bantu Belajar Mengajar*. Bandung: IKIP.
10. Endra Pitawarno, (2005). *Mikroprosesor dan Interfacing*. Penerbit ANDI Yogyakarta.
11. Data book Zilog dan Intel 1986.