

## BAB II

### LANDASAN TEORI

#### 2.1 Pendahuluan

Sistem digital adalah gabungan/kombinasi peralatan yang dirancang untuk memanipulasi satuan-satuan fisis atau informasi yang ditampilkan dalam bentuk digital yaitu informasi tersebut hanya menggunakan nilai-nilai diskrit [Tocci, 1995]. Peralatan ini sebagian besar dapat berupa untaian elektronik tapi dapat juga berupa rangkaian mekanik, magnetik atau pneumatis.

Rangkaian logika dikelompokkan dalam dua kelompok besar. Kelompok-kelompok gerbang tersebut adalah

1. Rangkaian logika kombinatorial
2. Rangkaian logika sekuensial.

Bentuk dasar dari logika kombinatorial adalah gerbang logika. Bentuk dasar dari rangkaian logika sekuensial adalah rangkaian *flip-flop*. Rangkaian logika sekuensial sangat bermanfaat karena karakteristik *memorinya*.

Beberapa jenis *flip-flop* akan dijelaskan secara terperinci pada bab ini. *Flip-flop* disebut juga "*multivibrator*", atau "*biner*". Dalam hal ini digunakan istilah "*flip-flop*". *Flip-flop* yang bermanfaat tersebut dapat dirangkai dari gerbang logika seperti gerbang NAND, atau dapat juga dibeli dalam bentuk IC. *Flip-flop* diinterkoneksi untuk membentuk rangkaian logika sekuensial untuk penyimpanan, pewaktu, perhitungan, dan pengurutan (*sequencing*).

## 2.1 Gerbang Kombinatorial Dasar

Gerbang logika (*logic gate*) merupakan dasar pembentuk sistem digital. Tegangan yang digunakan dalam gerbang logika adalah logika 1 (*high*) atau logika 0 (*low*).

Semua sistem digital disusun hanya menggunakan tiga gerbang logika dasar. Gerbang-gerbang dasar ini disebut gerbang AND, gerbang OR dan gerbang NOT.

### 2.2.1 Gerbang AND

Gerbang AND disebut gerbang “semua atau tidak satu pun”.

Simbol logika standar untuk gerbang AND



Gambar 2.1 Simbol Gerbang AND

Simbol ini menunjukkan A dan B sebagai masukan, sedangkan keluaran dinyatakan sebagai Y. Simbol tersebut merupakan simbol untuk gerbang AND 2 masukan. Tabel kebenaran untuk gerbang AND 2 masukan ditunjukkan pada tabel 2.1.

Masukan-masukan ditunjukkan sebagai digit biner (bit). Perlu diperhatikan bahwa keluaran akan menjadi 1 hanya apabila masukan A dan B adalah 1.

Tabel 2.1 Tabel Kebenaran Gerbang AND

Masukan		Keluaran
B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

0 = tegangan rendah

1 = tegangan tinggi

Dituliskan dengan persamaan Boolean :

$$A \cdot B = Y$$

(1)

### 2.2.2 Gerbang OR

Gerbang OR disebut gerbang “setiap atau semua” simbol logika untuk gerbang OR adalah:



Gambar 2.2 Simbol Gerbang OR

Tabel kebenaran gerbang OR, A dan B untuk masukan dan Y untuk keluaran. Persamaan Boolean adalah

$$A + B = Y$$

(2)

Tabel 2.2 Tabel Kebenaran Gerbang OR

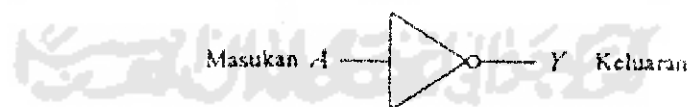
Masukan		Keluaran
B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

0 = tegangan rendah  
1 = tegangan tinggi

### 2.2.3 Gerbang NOT

Gerbang NOT disebut gerbang pembalik atau *inverter*. Gerbang NOT merupakan gerbang yang tidak biasa. Gerbang NOT hanya mempunyai satu masukan dan satu keluaran.

Proses pembalikan gerbang NOT sangat sederhana, pada saat A sebagai masukan mempunyai nilai 1, Y sebagai keluaran akan bernilai 0. Simbol gerbang NOT ditunjukkan pada gambar 2.3



Gambar 2.3 Simbol Gerbang NOT

Tabel 2.3 Tabel Kebenaran Gerbang NOT

Masukan	Keluaran
A	Y
0	1
1	0

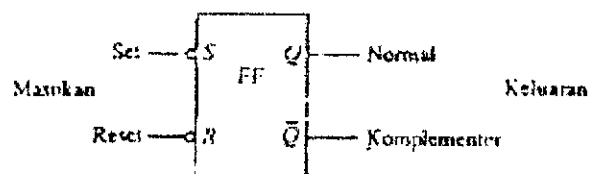
## 2.3 Gerbang Sekuensial Dasar

### 2.3.1 *Flip-Flop* RS

Kebanyakan *flip-flop* dasar disebut *flip-flop* RS. Simbol logika untuk *flip-flop* RS ditunjukkan gambar 2.4.

Simbol logika tersebut menunjukkan dua masukan, yang diberi label dengan set (S) dan reset (R) di sebelah kiri. *Flip-flop* RS pada simbol ini mempunyai masukan *low* aktif yang ditunjukkan dengan gelembung kecil pada masukan S dan R. Tidak seperti gerbang logika *flip-flop* mempunyai dua keluaran komplementer. Keluaran tersebut diberi label dengan Q dan  $\bar{Q}$ . Keluaran Q dianggap merupakan keluaran “normal” dan paling sering digunakan keluaran lain  $\bar{Q}$  merupakan komplemen dari keluaran Q dan disebut sebagai keluaran komplementer.

Pada kondisi normal, keluaran-keluaran ini selalu merupakan komplementer. Dengan demikian, bila  $Q = 1$  maka  $\bar{Q} = 0$ , atau bila  $Q = 0$  maka  $\bar{Q} = 1$ .



Gambar 2.4 Simbol Logika Untuk *Flip Flop* RS

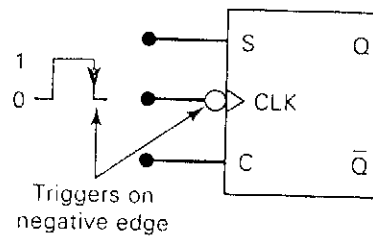
Tabel 2.4 Tabel Kebenaran *Flip Flop* RS

Inputs			Output
S	C	CLK	Q
0	0	↓	Q <sub>0</sub> (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous

### 2.3.2 Detak *Flip-flop* RS

*Latch* RS pada dasarnya merupakan suatu piranti asinkron. Peralatan seperti itu beroperasi serempak dengan detak atau piranti pewaktu. Bila masukan (seperti masukan set) diaktifkan, maka keluaran akan segera diaktifkan seperti pada rangkaian logika kombinatorial. Rangkaian-rangkaian penggerbangan dan kancing RS beroperasi secara asinkron.

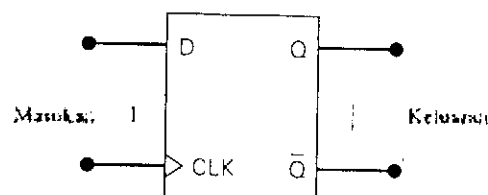
*Flip-flop* yang berdetak menambahkan suatu sifat sinkron yang berguna untuk kancing RS. *Flip-flop* RS yang berdetak akan beroperasi secara serempak dengan detak atau piranti pewaktu. Dengan kata lain, *flip-flop* tersebut beroperasi secara sinkron. Simbol logika untuk *flip-flop* RS yang berdetak diilustrasikan pada gambar, *flip-flop* ini mempunyai masukan set (S) dan masukan detak (CK) tambahan. *Flip-flop* RS yang berdetak mempunyai masukan normal (Q) dan keluaran komplementer ( $\bar{Q}$ ) seperti biasa.



Gambar 2.5 Simbol Logika Untuk *Flip Flop* Dengan *Clock*

### 2.3.3 *Flip-Flop D*

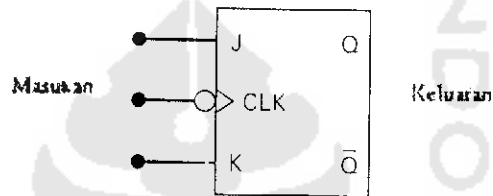
Simbol logika untuk jenis *flip-flop D*, diilustrasikan pada gambar 2.6. *Flip-flop D* tersebut hanya mempunyai masukan data tunggal ( $D$ ) dan masukan ( $CK$ ). Keluaran  $Q$  dan  $\bar{Q}$  ditunjukkan pada sebelah kanan simbol tersebut. *Flip-flop D* sering kali disebut sebagai *flip-flop tunda*. Nama ini menggambarkan dengan operasi unit ini. Apapun bentuk masukan pada masukan data ( $D$ ), masukan tersebut akan tertunda selama satu pulsa detak untuk mencapai keluaran normal ( $Q$ ). Data dipindahkan ke keluaran pada transisi pulsa *clock* rendah-ke-tinggi. *Flip-flop RS* yang berdetak dapat dirubah menjadi *flip-flop D* dengan menambahkan satu pembalik.



Gambar 2.6 Simbol Logika Untuk *Flip Flop D*

### 2.3.4 Flip-Flop JK

Simbol logika untuk *flip-flop* JK ditunjukkan pada gambar 2.7. Piranti ini dapat dianggap sebagai *flip-flop* universal. *Flip-flop* jenis lain dapat dibuat dari *flip-flop* JK. Simbol logika pada gambar mengilustrasikan tiga masukan sinkron (J,K,CLK). Masukan pada *flip-flop* JK adalah J dan K serta 1 detik. Simbol logika pada gambar menunjukkan keluaran normal (Q) dan keluaran komplementer ( $\bar{Q}$ ).



Gambar 2.7 Simbol Logika Untuk *Flip Flop* JK

Tabel 2.5 Tabel Kebenaran Untuk *Flip Flop* JK

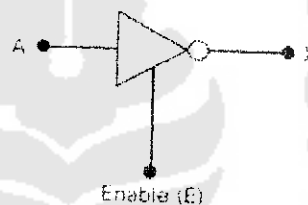
J	K	CLK	Q
0	0	↓	$Q_0$ (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\bar{Q}_0$ (toggles)

### 2.4. *Tristate Buffer* sebagai *Switch Digital*

Konfigurasi *tristate* merupakan 3 tipe konfigurasi keluaran TTL. *Tristate* menggunakan operasi *high speed* dari susunan *totem pole* dengan memperbolehkan keluaran di-AND-kan dan disebut TTL karena memperbolehkan 3 keadaan output yang mungkin, yaitu *Low*, *High* dan *High Impedance*.

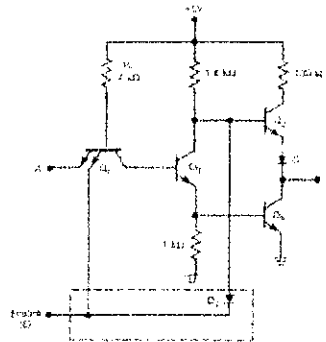


Keadaan *High Impedance* adalah kondisi dimana kedua transistor dalam susunan *totem pole* dimatikan (*off*) sehingga terminal keluaran dalam kondisi *High Impedance*, baik terhadap *ground* maupun *Vcc*. Dengan kata lain, *output* dalam keadaan terminal terbuka/mengambang, yaitu tidak dalam keadaan *Low* ataupun *High*. Dalam prakteknya, terminal *output* tidak terbuka sama sekali tetapi memiliki hambatan (*resistance*) relatif terhadap *Vcc* dan *ground*. Simbol *tristate* ditunjukkan pada gambar 2.8



Gambar 2.8 Simbol *Tristate Buffer*

Operasi *tristate* diperoleh dengan memodifikasi sirkuit/untai *totem pole* dasar. Gambar 2.9 menunjukkan *inverter tristate* dimana bagian didalam garis terputus-putus telah ditambahkan pada untai dasar sehingga untai memiliki 2 input. A adalah masukan logika normal. E adalah masukan *enable* yang bisa menghasilkan keadaan *High Impedance*. Tabel kondisi dari gambar seperti pada tabel 2.6.



Gambar 2.9 Rangkaian Internal Pembentuk *Inverter Tristate*

Tabel 2.6 Kondisi Untai Gambar 2.9

NAMA PIN		TRANSISTOR				KELUARAN
E	A	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	
0	0	On	Off	Off	Off	Hi-Z
0	1	Off	On	Off	Off	Hi-Z
1	0	On	Off	On	Off	1
1	1	Off	On	Off	On	0

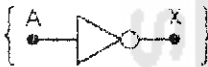

Operasi 2 keadaan ditentukan sebagai berikut :

1. Keadaan *Enable*

Dengan  $E = 1$ , untai beroperasi sebagai *inverter* normal karena tegangan *High* pada E tidak berpengaruh terhadap Q<sub>1</sub> atau Q<sub>2</sub>. Dalam kondisi *enable* ini, keluaran merupakan *invers* dari masukan logika A.

## 2. Keadaan *Disable*

Bila  $E = 0$ , untai menuju keadaan *Hi-Z* tanpa memperhatikan dari keadaan masukan logika A. Keadaan *Low* pada E memberi bias maju sambungan basis emitter transistor  $Q_1$  dan arus tidak melalui  $Q_2$ . Sehingga  $Q_2$  akan mati (*off*) dan  $Q_4$  juga mati (*off*). Keadaan *Low* pada E juga membias maju dioda  $D_2$  sehingga arus tidak melalui basis  $Q_3$  akibatnya  $Q_3$  juga akan mati (*off*). Dengan kedua transistor dalam konfigurasi *totem pole* maka dalam keadaan tidak menghantar, keluaran terminal adalah untai terbuka.

Operating conditions	
ENABLE (E)	OUTPUT
HIGH	Enabled: operates as an INVERTER 
LOW	Disabled: output is Hi-Z state. Input A has no effect. 

Gambar 2.10 Operasi *Tristate buffer*

## 2.5 Perangkat Keras (*Hardware*) Komputer

Perangkat keras yang digunakan dalam perancangan alat pengujian IC ini adalah *port* paralel. *Port* paralel banyak digunakan dalam berbagai macam aplikasi antarmuka. *Port* ini membolehkan kita memiliki masukan hingga 8 bit atau keluaran hingga 12 bit pada saat yang bersamaan, dengan hanya membutuhkan rangkaian eksternal sederhana untuk melakukan

suatu tugas tertentu. *Port* Paralel terdiri dari 4 jalur kontrol, 5 jalur status dan 8 jalur data. Biasanya dapat dijumpai sebagai *port printer*, dalam bentuk konektor DB-25 *female*.

*Port* paralel umumnya memiliki tiga alamat dasar yang bisa digunakan. Alamat dasar 3BCh pertama kali diperkenalkan sebagai alamat *port* paralel pada *card* video lama. Alamat ini kemudian sempat menghilang, saat *port* paralel dicabut dari *card* video. Sekarang muncul kembali sebagai pilihan untuk *port* paralel yang terpadu dengan *motherboard*, yang konfigurasinya dapat diubah melalui BIOS.

LPT1 biasanya memiliki alamat dasar 378h, sedangkan LPT2 adalah 278h. Ini adalah alamat umum yang bisa dijumpai, namun alamat-alamat dasar ini bisa berlainan antara satu komputer dengan komputer lainnya. Berikut gambar *parallel port* :



Gambar 2.11 Susunan kaki DB 25 untuk *Parallel Port*

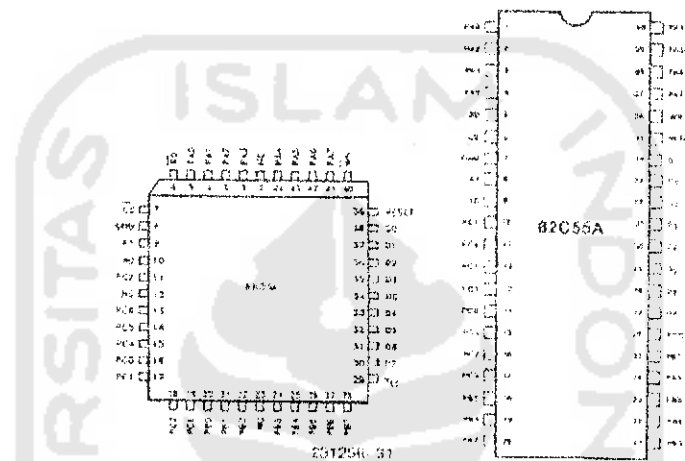
Tabel 2.7 Tabel Pin *Port* Paralel

Pin	Name	Dir	Description
1	/STROBE	→	Strobe
2	D0	↔	Data Bit 0
3	D1	↔	Data Bit 1
4	D2	↔	Data Bit 2
5	D3	↔	Data Bit 3
6	D4	↔	Data Bit 4
7	D5	↔	Data Bit 5
8	D6	↔	Data Bit 6
9	D7	↔	Data Bit 7
10	/ACK	←	Acknowledge
11	BUSY	↔	Busy
12	POUT	↔	Paper Out
13	SEL	↔	Select (Shared with RS232 RING-indicator)
14	GND	—	Signal Ground
15	GND	—	Signal Ground
16	GND	—	Signal Ground
17	GND	—	Signal Ground
18	GND	—	Signal Ground
19	GND	—	Signal Ground
20	GND	—	Signal Ground
21	GND	—	Signal Ground
22	GND	—	Signal Ground
23	+5V	→	+5 Volts DC (10 mA max)
24	N/c	-	Not connected
25	/RESET	→	Reset

## 2.6. PPI (*Programmable Peripheral Interface*) 8255

PPI (*Programmable Peripheral Interface*) 8255 merupakan piranti masukan/keluaran serbaguna yang dirancang untuk pemakaian bersama dengan pengolah mikro. Piranti ini dipakai sebagai antarmuka alat penguji IC dengan PC (*Personal Computer*) menggunakan *parallel port*.

Piranti memiliki 24 kaki masukan/keluaran yang dapat diprogram menjadi 2 kelompok. Tiap kelompok terdiri atas 12 bit I/O (*Input Output* = Masukan/Keluaran) dan dapat digunakan dalam 3 mode operasi



Gambar 2.12 Susunan kaki IC PPI 8255

### 2.6.1. Mode 0 (*Simple Protocol*) Basic Input-Output

Terdapat dua saluran 8 bit dan dua saluran 4 bit, masing-masing saluran dapat diprogram sebagai masukan atau keluaran, 16 macam konfigurasi I/O dapat dibentuk pada mode pertama atau mode 0 (nol) ini.

Transfer data tidak memerlukan sinyal khusus yang menandakan apakah telah terjadi transfer data atau belum. Contoh transfer data adalah jika akan dikeluarkan data ke peraga Led (1 = led hidup, 0 = mati) atau untuk memasukan data dari *thermostat* sederhana, yang senantiasa ada dan siap digunakan.

### 2.6.2 Mode 1 ( *Strobed Input/Output* )

Terdapat dua kelompok, yaitu kelompok A dan B, masing-masing mempunyai saluran data 8 bit yang dapat diprogram sebagai masukan atau keluaran dan status dari saluran data 8 bit.

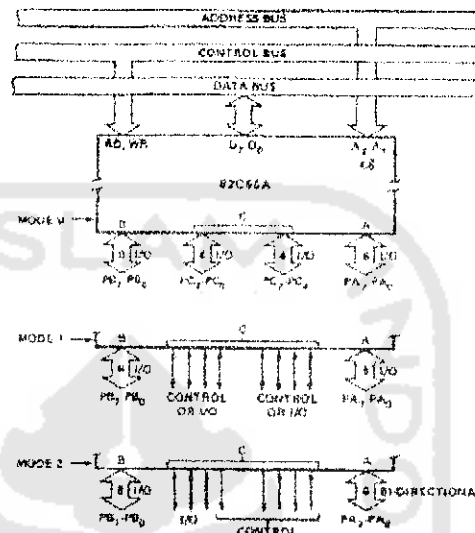
Transfer data semacam ini bisa dilihat pada *printer*. Cara kerjanya sebagai berikut: mikroprosesor mengeluarkan *strobe signal* ke *printer* yang artinya pemberitahuan bahwa ada data untuk *printer*. Jika *printer* telah menerima data maka *printer* memberikan *acknowledge signal* ke mikroprosesor yang artinya *printer* sudah menerima data dan sudah siap menerima kiriman berikutnya. Percakapan antara mikroprosesor dan *printer* membentuk suatu protokol jabat-tangan (*handshake*).

### 2.6.3. Mode 2 (*double handshaking protokol*) / *Bi-directional Bus*

Hanya menggunakan satu kelompok saja yaitu A, sebuah saluran 8 bit dan jalur data dua arah (*bidirectional*).

Transfer data semacam ini menggunakan aturan sebagai berikut: mikroprosesor sebagai pengirim mengeluarkan *strobe low signal* yang artinya pertanyaan ke *printer* apakah sudah siap dikirim data. Jika *printer* telah siap maka *printer* sebagai penerima data mengeluarkan *acknowledge high signal* sebagai pemberitahuan bahwa *printer* sudah siap. Setelah ada persetujuan, mikroprosesor mulai mengirim data yang ditandai dengan *strobe high signal* sebagai tanda mulai pengiriman data. Jika data telah diterima, *printer* mengeluarkan *acknowledge low signal* sebagai

pemberitahuan bahwa data sudah diterima dan siap menerima data berikutnya.



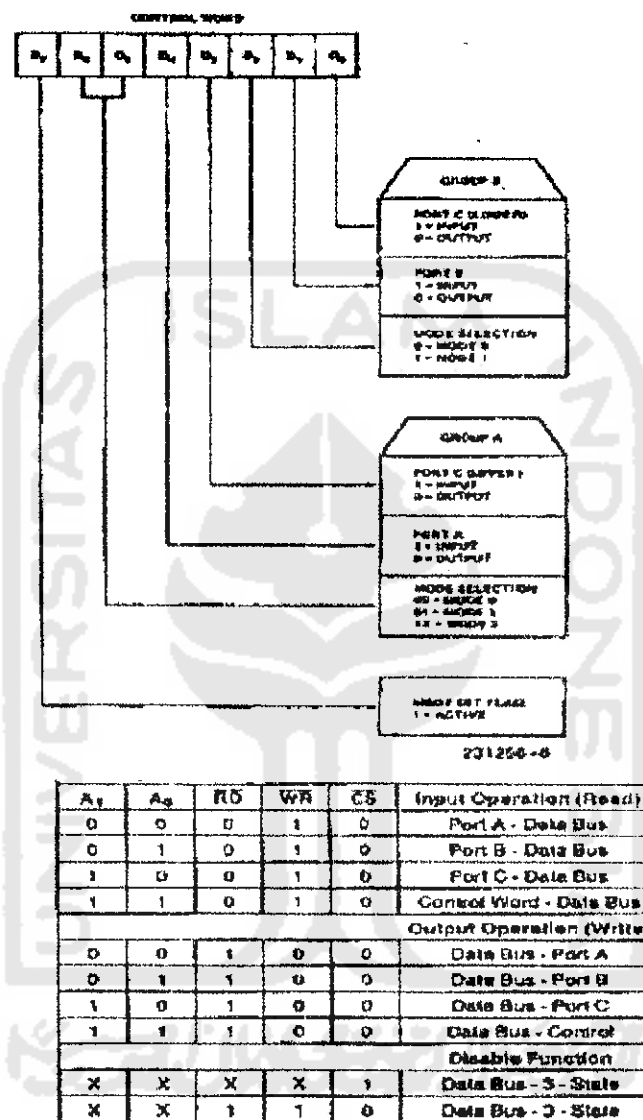
Gambar 2.13 Tiga macam mode operasi PPI 8255

Pada gambar 2.13 ditunjukkan konfigurasi untuk setiap mode yang telah dijelaskan serta konfigurasi kaki-kaki (pin) 8255.

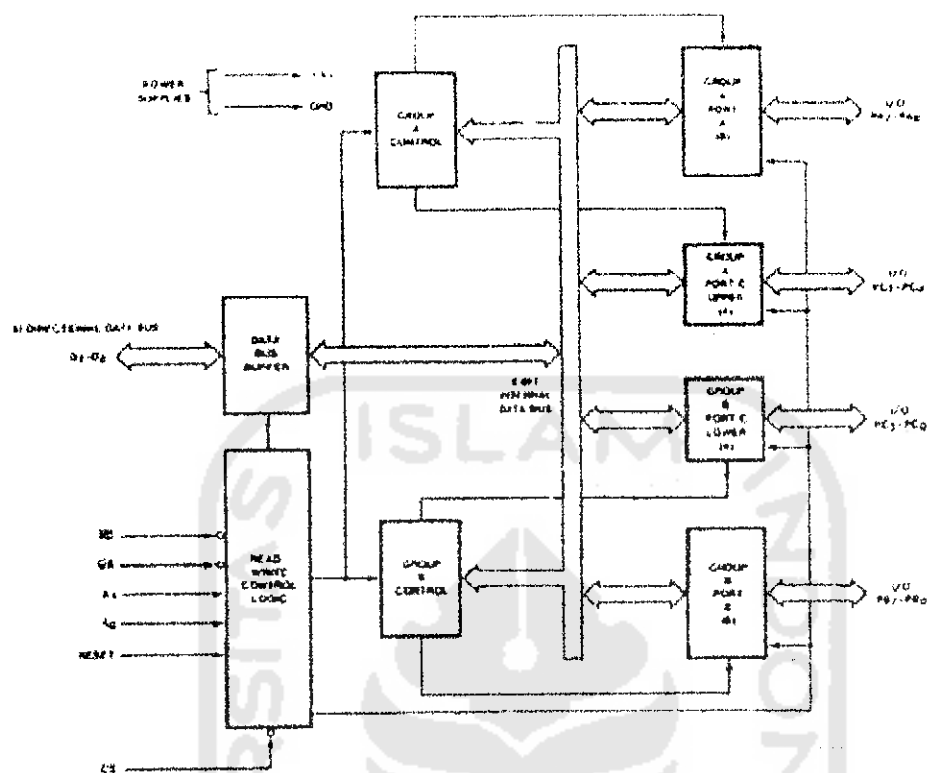
Pada gambar 2.14 ditunjukkan konfigurasi untuk *control word* dan tabel operasional PPI 8255 berdasarkan pin A<sub>1</sub>, A<sub>0</sub>, RD (*read*), WR (*write*) dan CS (*chip select*).

Agar PPI 8255 dapat berhubungan dengan komputer serta dapat diprogram melalui perangkat lunak, maka perlu ditambahkan rangkaian penerjemah alamat (*address decoder*) yang bisa terdiri dari piranti IC 7404 (*Hex-inverter*) dan 74133 (13-input NAND). Hubungan ke komputer dilakukan melalui slot tambahan atau *expansion slot* ISA bus.





Gambar 2.14 Konfigurasi *Control Word* dan Tabel Operasional PPI 8255



Gambar 2.15 Diagram blok PPI 8255

## 2.7 Perangkat Lunak (Software)

Perangkat lunak yang digunakan adalah Bahasa Pemrograman Delphi.

Delphi merupakan suatu bahasa pemrograman yang berorientasi objek atau OOP (*Object Oriented Programming*) yang dikembangkan dari bahasa pascal. Delphi dijalankan pada sistem operasi *microsoft windows*. Delphi dirilis oleh perusahaan perangkat lunak bernama Inprise (sebelumnya bernama Borland).

Sampai saat ini Inprise telah merilis delapan versi delphi, yaitu delphi 1 sampai delphi 8. Namun perbedaan nyata diantara keenam versi

tersebut terletak pada versi 1 dan versi 2 dan setelahnya. Hasil kompilasi program yang dibuat pada delphi 1 masih berupa aplikasi 16-bit dan berjalan pada sistem operasi 16 bit, yaitu windows 3.1. Sedangkan hasil kompilasi delphi 2 dan sesudahnya berupa aplikasi 32 bit dan hanya bisa dijalankan pada sistem operasi 32-bit (Windows 9X/ME/NT/XP/2000).

Pada sistem 16-bit, beberapa aplikasi tidak dapat dijalankan secara bersamaan (*multi tasking*). Jika suatu program melakukan pengulangan (*looping*) panjang, maka aplikasi lain akan terhenti dan seolah-olah menjadi *hang*. Hal ini tidak terjadi pada sistem 32 bit. Suatu proses akan dipecah menjadi paket-paket kecil kemudian dimasukkan dalam antrian proses. Prosesor akan mengeksekusi paket-paket tersebut sesuai dengan urutan posisi antriannya. Waktu yang digunakan prosesor untuk menjalankan proses dibagi menjadi satuan-satuan waktu yang dinamakan *time slice*. Satu paket dijalankan selama waktu *time slice*. Jika panjang *time slice* tidak mencukupi untuk proses suatu paket, maka proses akan ditunda dahulu untuk memproses paket yang lain. Pada putaran berikutnya, proses yang tertunda dilanjutkan kembali. Dengan demikian, suatu aplikasi yang melakukan proses panjang tidak akan mengganggu proses milik aplikasi lain.