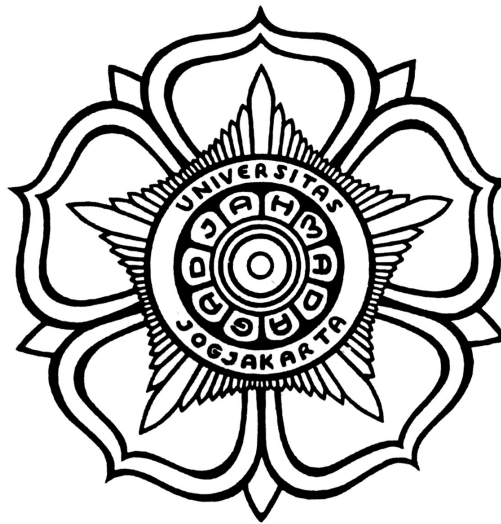


MODUL 05

# PRAKTIKUM SISTEM DIGITAL

TNF 2178 - 1 SKS



Disusun oleh:  
**Prof. Ir. Sunarno, M.Eng., Ph.D.**  
dan  
Tim Asisten Praktikum Sistem Digital

**LABORATORIUM SENSOR DAN SISTEM TELEKONTROL  
DEPARTEMEN TEKNIK NUKLIR DAN TEKNIK FISIKA  
FAKULTAS TEKNIK  
UNIVERSITAS GADJAH MADA  
2018**



# Chapter 1

## ENCODER-DECODER & MULTIPLEXER- DEMULTIPLEXER

### 1.1 Tujuan

- a. Mahasiswa mengenal prinsip kerja rangkaian *encoder-decoder* dan *multiplexer-demultiplexer*.
- b. Mahasiswa mengetahui aplikasi dari *encoder-decoder* dan *multiplexer-demultiplexer*.

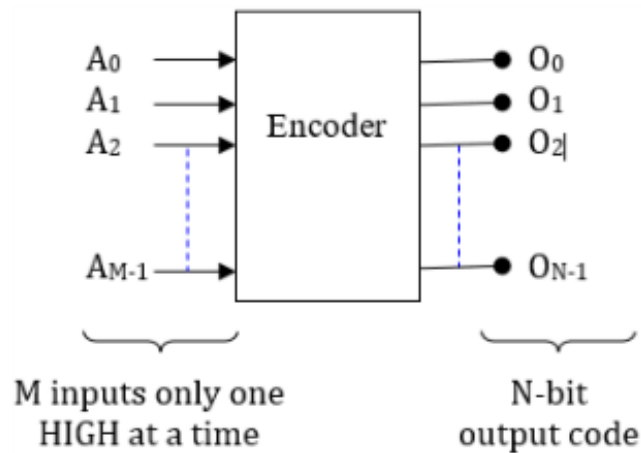
### 1.2 Materi

- a. *Encoder*
- b. *Decoder*
- c. *Multiplexer*
- d. *Demultiplexer*

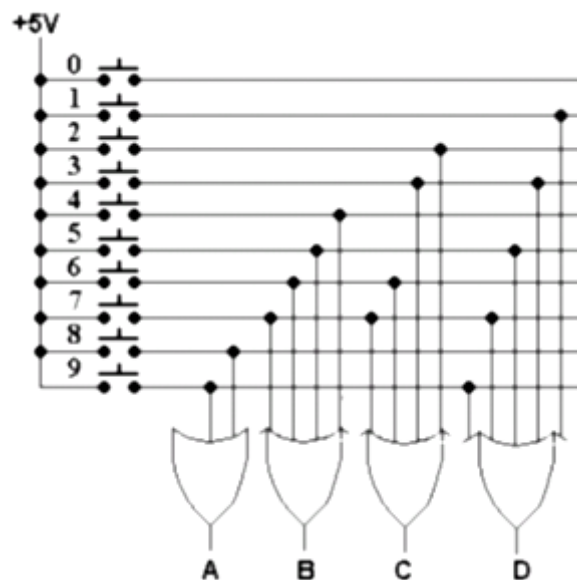
### 1.3 Teori

#### 1.3.1 Encoder

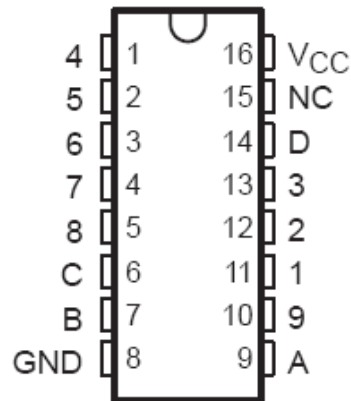
*Encoder* merupakan rangkaian logika yang berfungsi mengubah data yang ada pada *input*-nya menjadi kode-kode biner pada *output*-nya. Contoh *encoder* oktal ke biner atau disebut juga *encoder* 8 ke 3, berfungsi mengubah data bilangan oktal pada *input*-nya menjadi kode biner 3-bit pada *output*-nya.

Gambar 1.1: Diagram *encoder**Decimal To BCD Encoder*

Gambar 1.2 menunjukkan suatu tipe *encoder* yang sudah umum yaitu *decimal to BCD encoder*. *Switch* dengan penekan tombol mirip dengan tombol kalkulator dihubungkan dengan tegangan Vcc. Jika tombol 3 ditekan, maka gerbang-gerbang OR pada jalur C dan D akan mempunyai *input* bernilai 1. Oleh karena itu maka *output*-nya menjadi : ABCD = 0011, dan seterusnya.

Gambar 1.2: *Decimal to BCD encoder**IC 74LS147 (Encoder 9 ke 4)*

IC ini memiliki 2 bagian; *Input* dan *Output*. Pin A ... D merupakan *output* yang berubah berdasarkan masukan 1 ... 9.

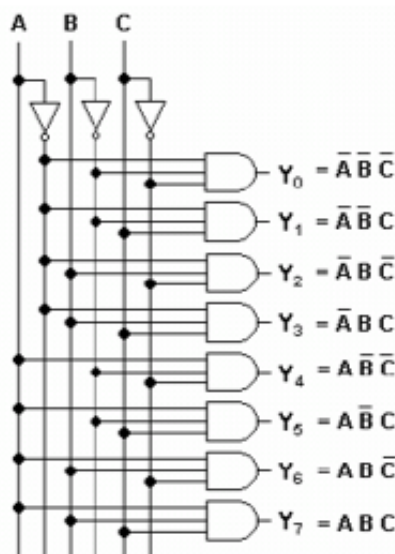


Gambar 1.3: Konfigurasi IC 74LS147

### 1.3.2 Decoder

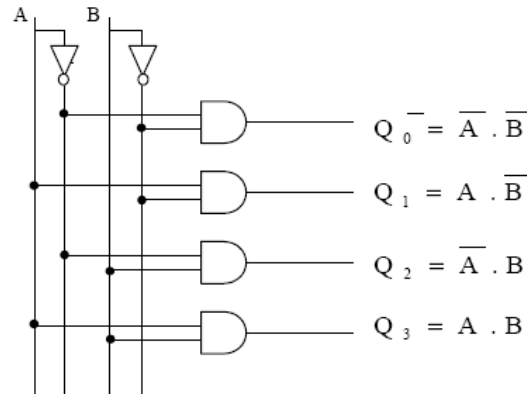
*Decoder* merupakan rangkaian logika yang berfungsi mengkode ulang atau menafsirkan kode-kode biner yang ada pada *input*-nya menjadi data asli pada *output*-nya, dan fungsinya merupakan kebalikan dari fungsi *encoder*.

Setiap  $n$  masukan dapat berisi logika 1 atau 0, ada  $2^N$  kemungkinan kombinasi dari masukan atau kode-kode. Untuk setiap kombinasi masukan ini hanya satu dari  $m$  keluaran yang akan aktif (berlogika 1), sedangkan keluaran yang lain adalah berlogika 0. Beberapa *decoder* didesain untuk menghasilkan keluaran *low* pada keadaan aktif, dimana hanya keluaran *low* yang dipilih akan aktif sementara keluaran yang lain adalah berlogika 1. Dari keadaan aktif keluarannya, *decoder* dapat dibedakan atas “*non inverted output*” dan “*inverted output*”.

Gambar 1.4: Rangkaian *Decoder*

### Decoder 2 ke 4

Dalam kemasan IC TTL, *decoder* 2 ke 4 disediakan oleh seri 74139 yang juga menyediakan fungsi *demultiplexer* 1 ke 4. Pada umumnya *input-input decoder* dalam kemasan IC berjenis *active-high*, sedangkan *output*-nya bervariasi di antara *active-high* dan *active-low*.



Gambar 1.5: *Decoder* 2-4

Table 1.1: Tabel kebenaran *Decoder* 2-4 dengan *enable* dan *output* jenis *active-low*

INPUT			OUTPUT			
$\bar{I}$	$S_1$	$S_0$	$\bar{Y}_0$	$\bar{Y}_1$	$\bar{Y}_2$	$\bar{Y}_3$
$\bar{E}$	B	A				
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

### Decoder Biner ke Oktal

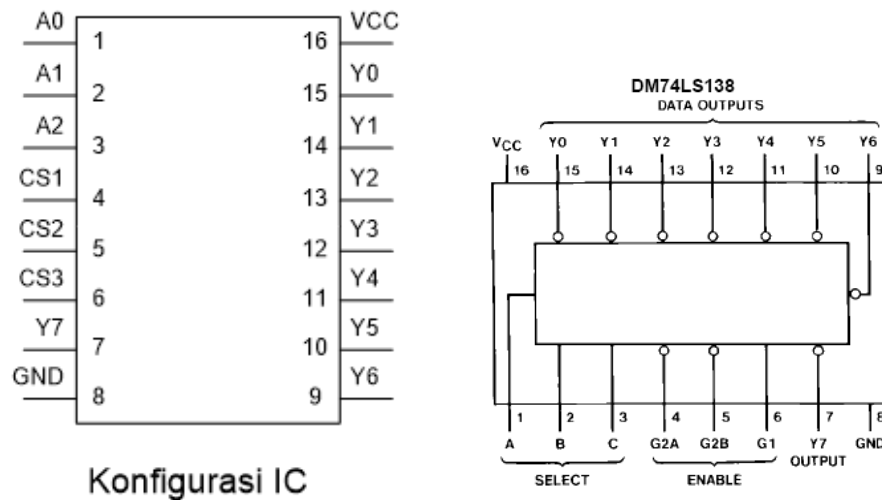
Rangkaian *decoder* biner ke oktal atau *decoder* 3 ke 8 dalam kemasan disediakan oleh IC 74138 dengan spesifikasi pin seperti ditunjukkan pada Gambar 1.6.

### Decoder BCD ke Decimal

*Decoder* yang berfungsi menafsirkan kode-kode BCD ke nilai desimal atau dinamakan pula *decoder* 4 ke 10. Disediakan oleh IC TTL dengan seri 7442, 7445, 74145, 74445, dan 74141.

### Decoder 4 ke 16

*Decoder* 4 ke 16 menyediakan 16 saluran *output* sebagai saluran-saluran yang menampilkan hasil tafsiran terhadap kode 4 bit yang dimasukkan melalui *input*-nya. Dalam kemasan IC, *decoder* ini disediakan oleh IC dengan nomor



Gambar 1.6: IC seri 74138 (a) Konfigurasi IC (b) Spesifikasi pin

seri 74154 dan 74159. Kedua IC tersebut menyediakan fungsi decoder 4 ke 16 dengan *output* jenis *active-low*.

### 1.3.3 Multiplexer

*Multiplexer* berarti “dari banyak ke dalam (menjadi) satu”. Sebuah *multiplexer* adalah rangkaian yang memiliki banyak masukan tetapi hanya satu keluaran. *Multiplexer* sering disingkat dengan MUX atau MPX.

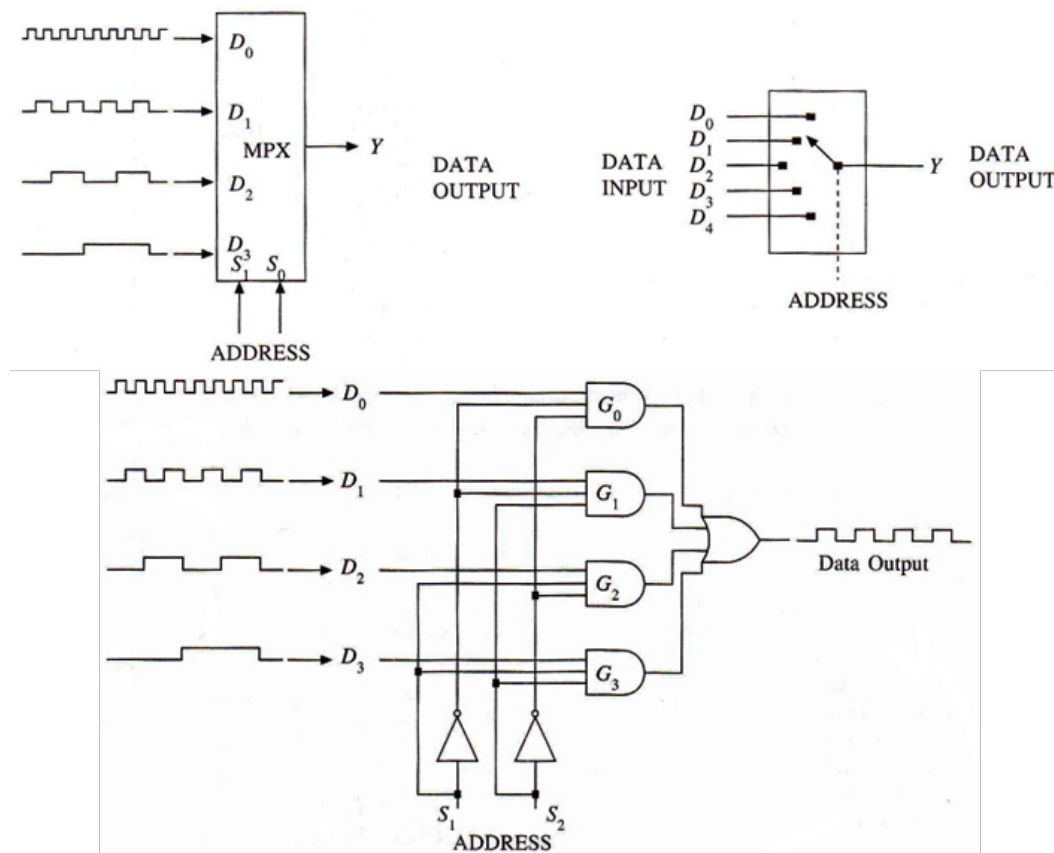
Dengan menggunakan sinyal kendali kita dapat mengatur penyaluran masukan tertentu menuju keluarannya. Sinyal kendali ini akan mengatur bagian mana atau alamat (*address*) mana yang akan diaktifkan atau dipilih. Perangkat *multiplexer* atau disebut juga pemilih data (*data selector*) adalah sebuah rangkaian logika yang menerima beberapa masukan data dan hanya satu di antara mereka yang dilewatkan ke keluaran pada suatu waktu.

Jalur data masukan yang diharapkan keluaran dikendalikan oleh sinyal kendali alamat (*address*) atau disebut juga masukan SELECT. Lambang dari *multiplexer* dapat dilihat pada Gambar 1.7.

Gambar 1.7 memperlihatkan sebuah *multiplexer* yang dasar dan sederhana, terdiri hanya 4 masukan data digital dengan keluaran 1 jalur digital. *Multiplexer* tersebut sering diartikan sebagai *multiplexer* 4 ke-1. Bit-bit data masukan terdiri dari D0 sampai D3. Hanya satu di antara masukan-masukan tersebut yang diteruskan ke bagian keluaran. Sinyal kendali address S0 dan S1, menentukan jalur data yang berisi bit-bit biner mana yang akan dikeluarkan, sebagai contoh, jika:

$$S_0 S_1 = 0 \ 1$$

maka gerbang AND atau gerbang G1, menjadi aktif (*enable*) dan semua gerbang AND yang lain dalam keadaan tak aktif atau dilumpuhkan (*disable*).

Gambar 1.7: *Multiplexer*

Karena itu, bit data D1, diteruskan ke bagian keluaran dan memberikan hasil:

$$Z = D_1$$

Jadi, data apapun yang lewat pada jalur D1 menjadi G1, akan dikirimkan melalui gerbang OR dan dikeluarkan ke keluaran Z.

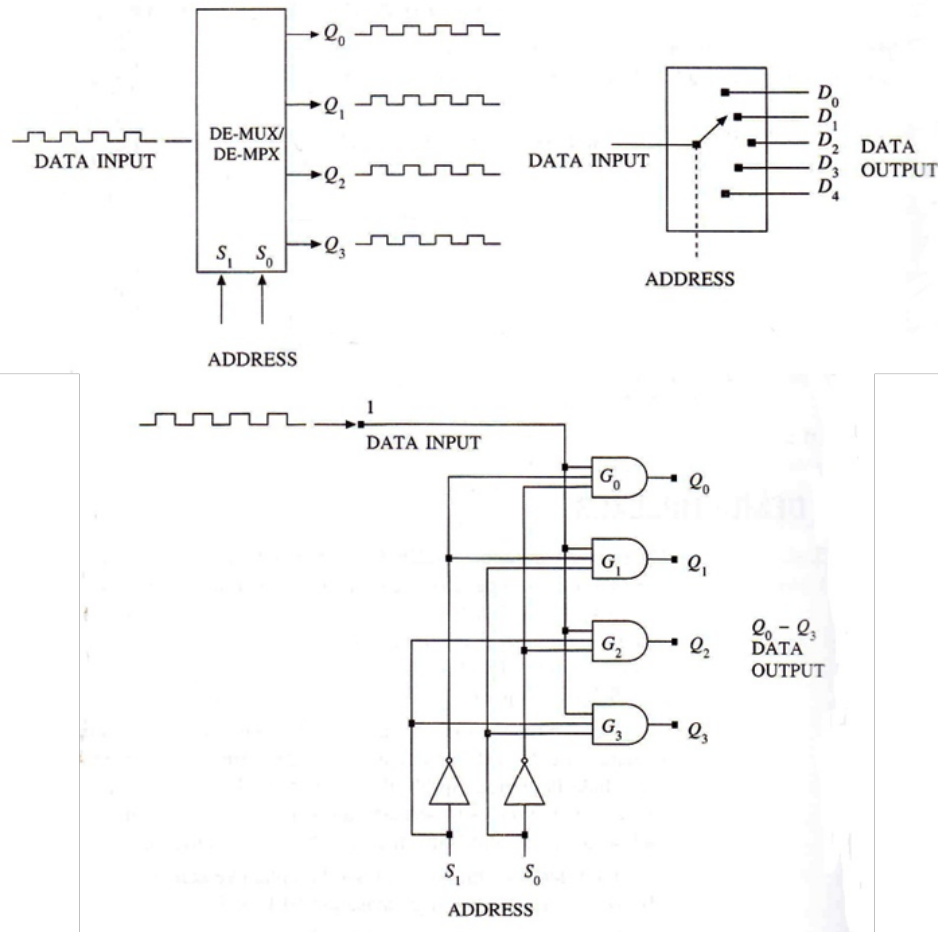
### 1.3.4 Demultiplexer

*Demultiplexer* merupakan kebalikan dari *multiplexer*, yang berarti “dari satu menjadi banyak”. *Demultiplexer* merupakan rangkaian yang memiliki satu masukan tetapi memiliki banyak keluaran. *Demultiplexer* sering disingkat dengan DEMUX atau DEMPX.

Dengan menggunakan sinyal kendali, kita dapat mengatur penyaluran masukan pada keluaran tertentu yang diinginkan. Sinyal kendali ini akan mengatur bagian mana atau alamat (*address*) mana yang akan diaktifkan atau dipilih. Perangkat *demultiplexer* disebut juga distribusi data atau penyalur data (*data distributor*) yaitu sebuah rangkaian logika yang menerima hanya satu masukan data dan melewatkan ke salah satu di antara beberapa keluaran.



Jalur dari data masukan yang diharapkan ke keluaran dikendalikan oleh sinyal kendali alamat (*address*) atau disebut juga masukan SELECT.



Gambar 1.8: Lambang *demultiplexer*

Gambar 1.8 memperlihatkan sebuah *demultiplexer* yang dasar dan sederhana, yang terdiri dari hanya satu masukan digital dengan keluaran 4 jalur digital. *Demultiplexer* tersebut sering diartikan dengan *demultiplexer* 1 ke-4. Bit-bit data masukan hanya satu jalur. Jalur ini pada umumnya berasal dari *multiplexer*. Kemudian melewati perangkat *demultiplexer*, yang kemudian akan dipilih oleh sinyal kendali *address*. Jalur yang dipilih untuk menyalurkan data, diteruskan ke bagian keluaran yang dipilih. Jalur yang dipilih adalah antara  $D_0$  sampai  $D_3$ . Sinyal kendali *address*  $S_0$  dan  $S_1$ , menentukan jalur data yang berisi bit-bit biner, jalur mana yang akan dikeluarkan. Sebagai contoh jika:

$$S_1 S_0 = 01$$

maka gerbang AND atau gerbang  $G_1$ , menjadi aktif (*enable*) dan semua gerbang AND yang lain dalam keadaan tidak aktif atau dilumpuhkan (*disable*). Karena itu, bit data yang berasal dari masukan diteruskan ke bagian keluaran,

dan memberikan hasil:

$$Data_{input} = G_1 \quad \text{dengan keluaran } Q_1$$

Jadi, apapun data yang lewat pada jalur D1 melalui G1, akan dikirimkan melalui gerbang AND yang kedua atau G1.

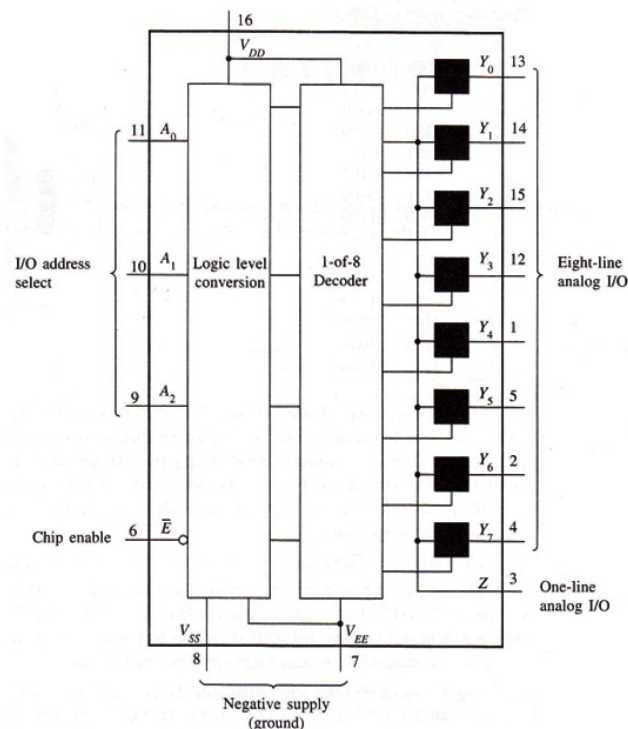
#### *Multiplexer-Demultiplexer Analog*

IC perangkat *multiplexer-demultiplexer* juga tersedia di pasaran, dengan teknologi CMOS. Contohnya: IC 4051, 4052, atau 4053 (secara umum diberi simbol 405X) dan juga 4067 yang merupakan gabungan antara perangkat *multiplexer* dan *demultiplexer* yang dibuat dengan teknologi CMOS.

IC-IC tersebut dapat berfungsi sebagai salah satu dari 2 susunan *multiplexer-demultiplexer* secara bergantian, karena masukan-keluarannya adalah 2 atau sering disebut dengan *Input/Output Birectional*. Masukan keluaran *bidirectional*, berarti bahwa aliran data dapat menuju ke salah satu dari dua arah atau dapat bolak-balik secara bergantian.

Isyarat masukan dan keluaran dalam bentuk analog, artinya level masukan keluarannya bervariasi. Tidak harus 0 atau 1 seperti level logika yang telah dipelajari. Level tegangan analog masukan dan keluaran harganya antara catu level negatif dan positif.

Diagram fungsi untuk IC 405X *multiplexer-demultiplexer* digambarkan dalam Gambar 1.9. Dalam IC CMOS 405X terdapat 8 kotak yang dalam diagram



Gambar 1.9: IC 405X *multiplexer-demultiplexer*

fungsi dinyatakan bahwa jalur masukan/keluarannya dalam 2 arah atau I/O *bidirectional*.

Prosedur pengoperasiannya adalah sebagai berikut, ketika digunakan sebagai *multiplexer*, sinyal analog datang menuju jalur Y0 sampai Y7, dan *decoder* akan memilih melalui masukan yang mana (Y0 sampai Y7) dan keluar menuju ke jalur Z.

Sebagai *demultiplexer*, hubungan tersebut dibalik, dengan masukan sinyal analog datang ke dalam jalur Z dan *decoder* akan memilih keluaran mana yang dituju (salah satu dari keluaran Y0 sampai Y7).

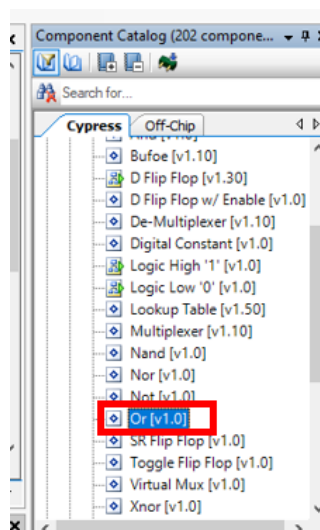
## 1.4 Alat dan Bahan

- a. Buku praktikum
- b. Modul PSoC
- c. Kabel *jumper*
- d. Kabel USB *Male to Female*

## 1.5 Percobaan

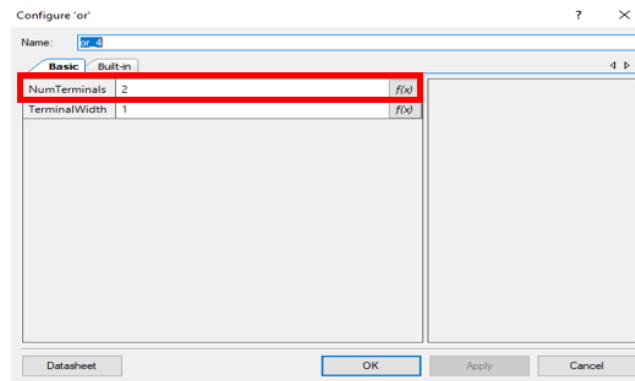
### 1.5.1 Encoder

1. Buka PSoC Creator
2. Pada Workspace TopDesign.cysch masukkan 4 buah gerbang OR yang didapat dari Component Catalog seperti pada Gambar 1.10



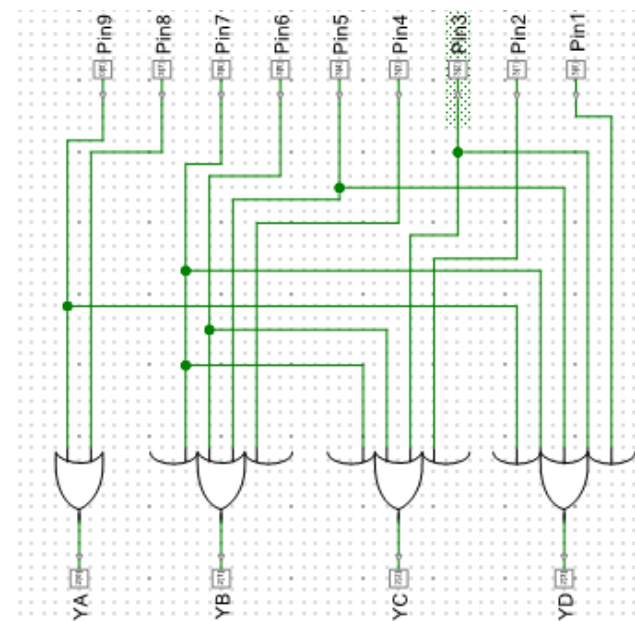
Gambar 1.10: *Component Catalog*

- Ubah jumlah *input* gerbang logika OR dengan cara *double click* komponen gerbang OR, pada *tab* Basic, bagian NumTerminals, ubah sesuai jumlah input yang diinginkan, kemudian klik OK.



Gambar 1.11: Mengubah jumlah pin input gerbang logika

- Masukkan Digital Input Pin sebanyak 9 buah, ubah nama setiap *input* menjadi Input1 hingga Input9 dan ubah mode Digital Input Pin menjadi Resistive Pull Down dengan cara *double click* pada komponen → General → Drive Mode → Resistive Pull Down → OK.
- Masukkan Digital Output Pin sebanyak 4 buah dan ubah nama menjadi YA-YD.
- Hubungkan semua komponen hingga menjadi seperti gambar berikut



Gambar 1.12: Skema Rangkaian Encoder

7. Pada *tab* Workspace Explorer, klik Pins untuk pengaturan *port*. Definisikan semua *port* seperti pada Gambar 1.13

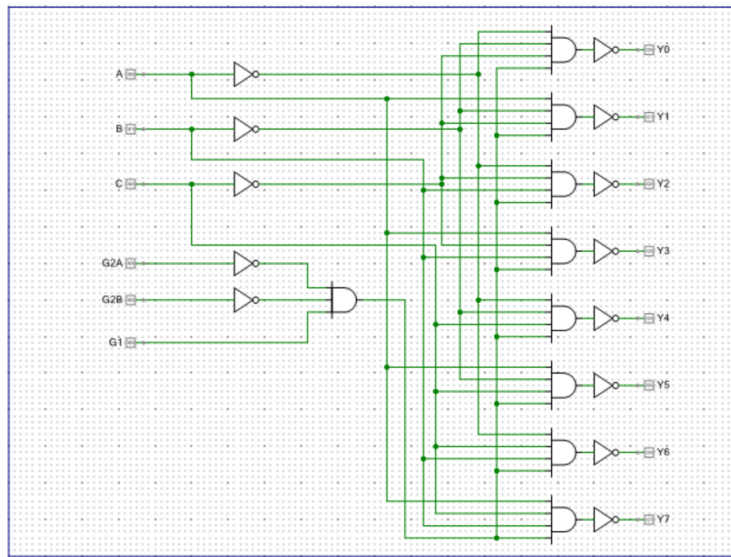
	Name	Port	Pin	Lock
<input type="checkbox"/>	Pin1	P3[0]	29	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin2	P3[1]	30	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin3	P3[2]	31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin4	P3[3]	32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin5	P3[4]	33	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin6	P3[5]	34	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin7	P3[6]	36	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin8	P3[7]	37	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Pin9	P0[0]	48	<input checked="" type="checkbox"/>
<input type="checkbox"/>	YA	P2[0]	62	<input checked="" type="checkbox"/>
<input type="checkbox"/>	YB	P2[1]	63	<input checked="" type="checkbox"/>
<input type="checkbox"/>	YC	P2[2]	64	<input checked="" type="checkbox"/>
<input type="checkbox"/>	YD	P2[3]	65	<input checked="" type="checkbox"/>

Gambar 1.13: *Setting Port*

8. Lakukan *compile* desain dengan cara klik Build → Build (nama *project*). Kemudian, unduh program ke modul PSoC dengan cara klik Debug → Program.
9. Isi hasil percobaan pada Tabel Hasil Pengamatan.

### 1.5.2 Decoder

1. Lakukan prosedur seperti yang disebutkan pada percobaan *encoder* untuk mendesain seperti pada Gambar 1.14.
2. Atur *port* seperti pada Gambar 1.15
3. Unduh program ke modul PSoC dengan cara yang telah disebutkan pada percobaan *encoder*
4. Pastikan rangkaian pada PSoC *Board Kit* sesuai dengan desain skema pada *Top Desain PSoC Creator*
5. Isi hasil percobaan pada Tabel Hasil Pengamatan



Gambar 1.14: Skema Rangkaian Decoder

	Name	Port	Pin	Lock
<input checked="" type="checkbox"/>	A	P3[0]	29	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	B	P3[1]	30	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	C	P3[2]	31	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	G1	P3[3]	32	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	G2A	P3[4]	33	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	G2B	P3[5]	34	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y0	P2[0]	62	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y1	P2[1]	63	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y2	P2[2]	64	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y3	P2[3]	65	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y4	P2[4]	66	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y5	P2[5]	68	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y6	P2[6]	1	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y7	P2[7]	2	<input checked="" type="checkbox"/>

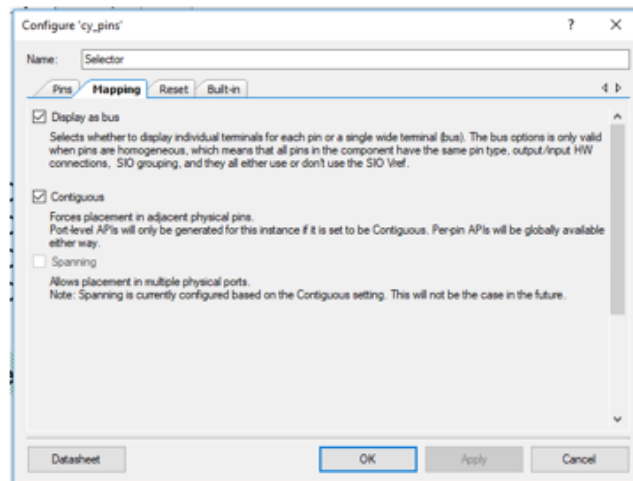
Gambar 1.15: *Setting Port*

### 1.5.3 Multiplexer

1. Pada Component Catalog, pilih Digital, kemudian Logic, kemudian pilih Multiplexer dan *drag* blok tersebut ke TopDesign.cysch
2. Pilih Ports and pins, kemudian pilih Digital Input Pin, kemudian *drag* blok tersebut ke TopDesign.cysch. Untuk percobaan ini, dibutuhkan 5 buah *input* untuk kaki D0-D3 dan *selector*.
3. Atur konfigurasi *input pin* menjadi Resistive Pull Down dengan cara seperti

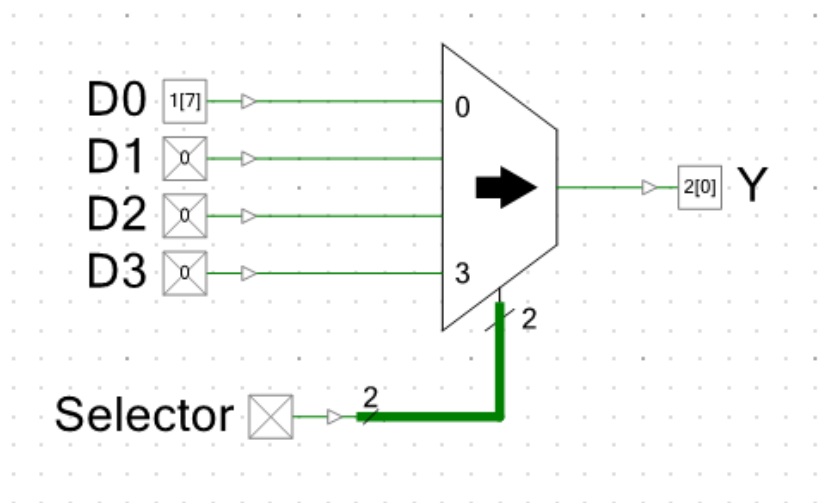
yang telah disebutkan pada percobaan *encoder*. Ulangi untuk seluruh *pin*.

4. Khusus untuk *pin* yang difungsikan sebagai *selector*, klik 2 kali pada *pin* tersebut, kemudian klik pada *tab* Mapping, kemudian *check* Display as bus. Kemudian, klik pada *tab* Pins, lalu ubah Number of pins menjadi 2 atau sesuai jumlah *number of selector* yang digunakan pada blok *multiplexer* seperti Gambar 1.16.



Gambar 1.16: blok Skema Multiplexer

5. Masukkan pula Digital Output Pin ke dalam TopDesign.cysch
6. Hubungkan semua blok hingga membentuk rangkaian seperti Gambar 1.17.



Gambar 1.17: blok Skema Multiplexer

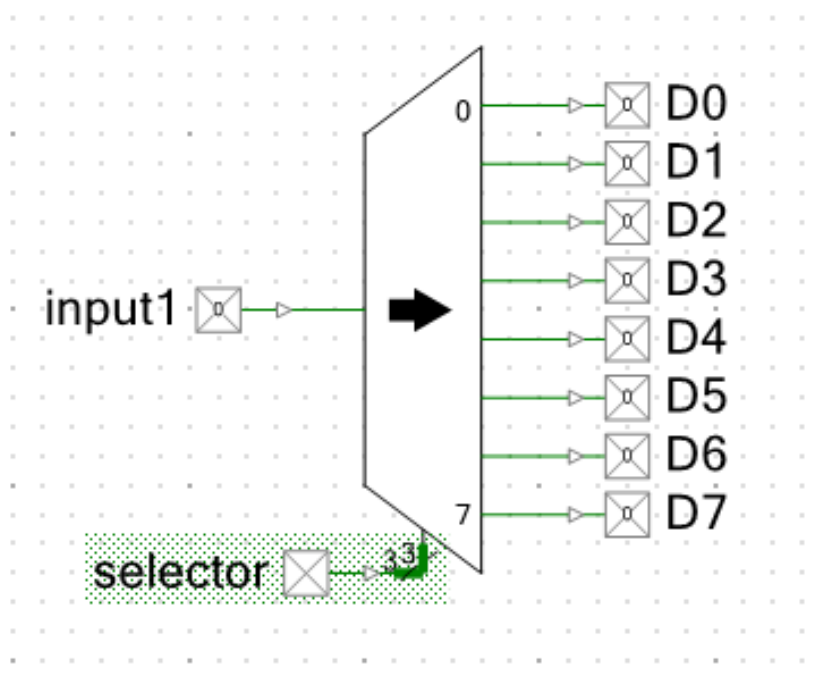
	Name	Port	Pin	Lock
	\Selector[1:0]\			<input type="checkbox"/>
<input checked="" type="checkbox"/>	D0	P3[0]	29	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	D1	P3[1]	30	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	D2	P3[2]	31	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	D3	P3[3]	32	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Y	P2[0]	62	<input checked="" type="checkbox"/>

Gambar 1.18: *Setting Port*

7. Atur konfigurasi *port* seperti pada Gambar 1.18
8. Lakukan *build* kemudian unduh program ke modul PSoC
9. Isi hasil percobaan pada Tabel Hasil Pengamatan

#### 1.5.4 Demultiplexer











1. Lakukan prosedur seperti yang disebutkan pada percobaan *multiplexer* untuk menghasilkan rangkaian seperti Gambar 1.19



Gambar 1.19: Blok Skema Demultiplexer

2. Atur *port* seperti pada Gambar 1.20



	Name /	Port	Pin	Lock
	\selector[2:0]\	P0[2:0] ▼	50...48 ▼	<input type="checkbox"/>
	D0	P0[3] ▼	51 ▼	<input type="checkbox"/>
	D1	P0[6] ▼	55 ▼	<input type="checkbox"/>
	D2	P15[4] ▼	60 ▼	<input type="checkbox"/>
	D3	P0[7] ▼	56 ▼	<input type="checkbox"/>
	D4	P1[5] ▼	16 ▼	<input type="checkbox"/>
	D5	P0[4] ▼	53 ▼	<input type="checkbox"/>
	D6	P0[5] ▼	54 ▼	<input type="checkbox"/>
	D7	P12[2] ▼	46 ▼	<input type="checkbox"/>
	input1	P12[3] ▼	47 ▼	<input type="checkbox"/>

Gambar 1.20: *Setting Port*

3. Lakukan *build* kemudian unduh program ke modul PSoC
4. Pastikan rangkaian pada PSoC *Board Kit* sesuai dengan desain skema pada *Top Desain* PSoC Creator
5. Isi hasil percobaan pada Tabel Hasil Pengamatan

## 1.6 Tabel Percobaan

### 1. Encoder

Input									Output			
1	2	3	4	5	6	7	8	9	A	B	C	D
1	1	1	1	1	1	1	1	1				
1	1	1	0	0	0	0	0	0				
0	1	0	1	0	1	0	0	1				
1	0	1	0	1	0	0	1	1				
1	0	1	0	1	0	1	1	1				
0	1	0	1	0	1	1	1	1				
1	0	1	0	1	1	1	1	1				
0	1	0	1	1	1	1	1	1				
1	0	1	1	1	1	1	1	1				
0	1	1	1	1	1	1	1	1				
0	1	0	1	0	1	0	1	1				
0	0	0	0	0	0	0	0	0				
1	1	1	1	1	1	0	1	1				
1	1	1	1	1	1	1	1	0				
0	0	0	0	0	0	0	1	1				

Gambar 1.21: Tabel Percobaan Encoder

### 2. Decoder

INPUT						OUTPUT							
G1	G2A	G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	1	1	0	1	1								
1	0	1	0	1	1								
1	0	0	0	0	0								
1	0	0	0	0	1								
1	0	0	0	1	0								
1	0	0	0	1	1								
1	0	0	1	0	0								
1	0	0	1	0	1								
1	0	0	1	1	0								
1	0	0	1	1	1								

Gambar 1.22: Tabel Percobaan Decoder

## 3. Multiplexer

INPUT				ADDRESS		OUTPUT
D3	D2	D1	D0	B	A	Y (V)
0	0	0	0	0	0	
0	0	0	1			
0	0	1	0			
0	1	0	0			
1	0	0	0			
0	0	0	0	0	1	
0	0	0	1			
0	0	1	0			
0	1	0	0			
1	0	0	0			
0	0	0	0	1	0	
0	0	0	1			
0	0	1	0			
0	1	0	0			
1	0	0	0			
0	0	0	0	1	1	
0	0	0	1			
0	0	1	0			
0	1	0	0			
1	0	0	0			

Gambar 1.23: Tabel Percobaan Multiplexer

## 4. Demultiplexer

INPUT	ADDRESS			OUTPUT							
	C	B	A	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

Gambar 1.24: Tabel Percobaan Demultiplexer



## DAFTAR PUSTAKA

David Bucchlah, Wayne McLahan, “Applied Electronic Instrumentation And Measurement”, MacMilian Publishing Company, 1992.

Eggebrecht, Lewis C., Interfacing to The IBM PC, Howard W. Sams & Co., Indianapolis, 1987.

Hall, Douglas V., Microprocessor and Interfacing : Programming and Hardware, McGraw-Hill Book Company, Singapore, 1987.

Hodges D. , Jacson, Nasution S. “Analisa dan Desain Rangkaian Terpadu Digital”, Erlangga, Jakarta, 1987.

Ian Robertson Sinclair, Suryawan, “Panduan Belajar Elektronik Digital”, ElexMedia Komputindo, Jakarta, 1993.

K.F. Ibrahim, “Teknik Digital”, Andi Offset , Jakarta, 1996.

Sendra, Smith, Keneth C., “ Rangkaian Mikroelektronika”, Penerbit Erlangga, Jakarta, 1989.

Singh, Avtar & Walter A. Triebel, The 8088 Microprocessor : Programming, Interfacing, Software, Hardware and Applications, Prentice-Hall International Inc., New Jersey, 1987.

Sofyan H. Nasution, “Analisa dan Desain Rangkaian Terpadu Digital”, Penerbit Erlangga, Jakarta, 1987.

Sutrisno, “Rangkaian Digital dan Rancangan Logika”, Erlangga, Jakarta, 1990.

Tokheim. R., “Elektronika Digital”, Edisi Kedua, Erlangga, Jakarta, 1995.

Wijaya Widjanarka N., “Teknik Digital”, Erlangga, Jakarta, 2006.