

# บทที่ 5 การออกแบบวงจรดิจิทัล

1. บทนำ
2. มาตรฐานเอกสารสำหรับระบบดิจิทัล
3. หลักการออกแบบวงจรคอมพิวเตอร์ขั้นสูง
4. วงจรถอดรหัส(Decoder Circuits)
5. วงจรเข้ารหัส (Encoder Circuits)
6. วงจรมัลติเพลกซ์หรือวงจรถูกเลือกข้อมูล
7. ดีมัลติเพลกซ์ (DEMULTIPLEXER)
8. วงจรบวกเลขไบนารี (Binary addition Circuit)
9. วงจรคูณเลขฐานสอง
10. วงจรแปลงรหัส (CODE CONVERTER)
11. วงจรเปรียบเทียบเลขฐานสอง (Binary Comparator)



## บทนำ

!!!!???

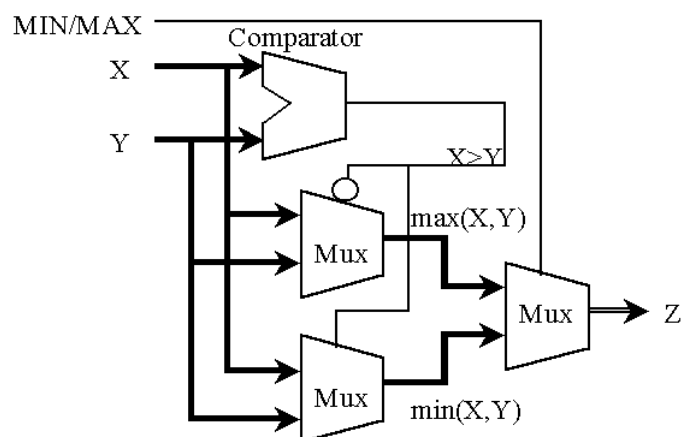
2

## มาตรฐานเอกสารสำหรับระบบดิจิทัล

- แผนภาพบล็อก (Block diagram)
- แผนภาพลอจิก (Logic Diagram)
- Schematic diagram

3

## แผนภาพบล็อก (BLOCK DIAGRAM)



4

## LOGIC DIAGRAM และ SCHEMATIC DIAGRAM

### สัญลักษณ์เกต (Gate Symbols)

นิพจน์ AND  $F = A \cdot B = \overline{\overline{A \cdot B}} = \overline{(\overline{A + \overline{B}})}$

นิพจน์ NAND  $F = \overline{A \cdot B} = \overline{A} + \overline{B}$

นิพจน์ OR  $F = A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}$

นิพจน์ NOR  $F = \overline{A + B} = \overline{A} \cdot \overline{B}$

AND



NAND



OR



NOR



INVERTER



BUFFER



5

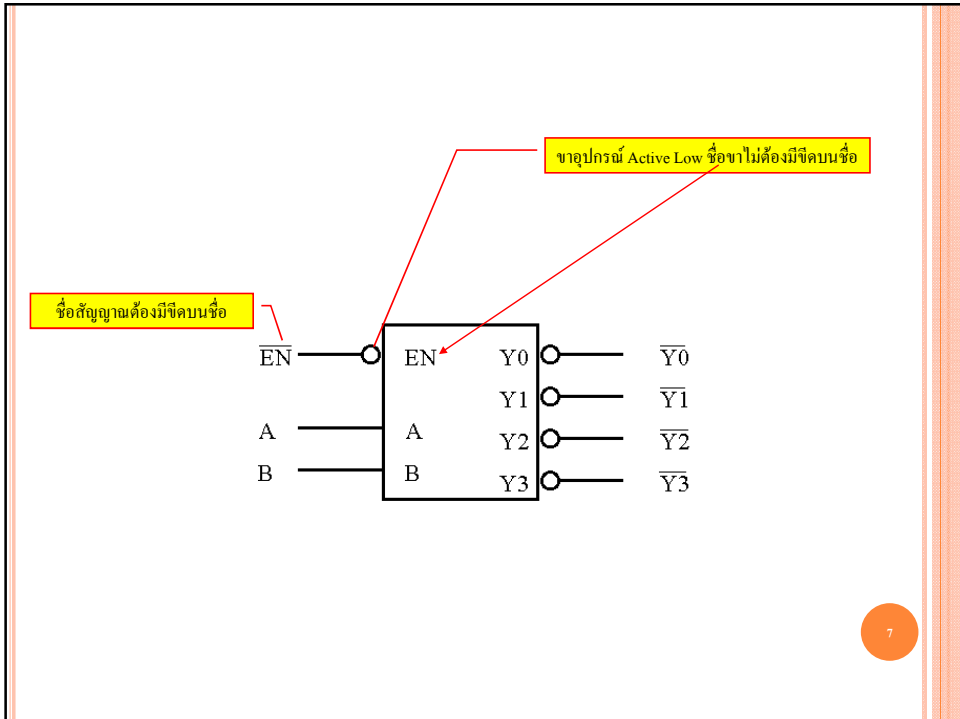
### ชื่อสัญญาณและระดับทำงาน (SIGNAL NAMES AND ACTIVE LEVELS)

- ชื่อสัญญาณอินพุตและสัญญาณเอาต์พุต ควรเป็นชื่อของตัวแปร และควรบ่งบอกสถานะทำงาน (Active) ว่าเป็น high หรือ low ถ้าทำงานด้วยสถานะ low ให้ชื่อสัญญาณมีขีดอยู่บนชื่อสัญญาณ เช่น A, B, F, Y, ENABLE, และ

- ระดับลอจิกทำงานของขา (PIN) อุปกรณ์

รูปแบบสัญลักษณ์ลอจิกจะใช้ bubble บอกระดับ active low เช่น วงจรถอดรหัส 2-to-4 line สัญญาณ EN, Y0, Y1, Y2 และ Y3 ทำงานด้วยระดับ 0

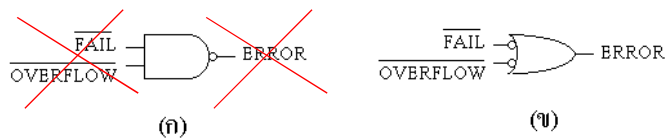
6



7

## BUBBLE-TO-BUBBLE LOGIC DESIGN

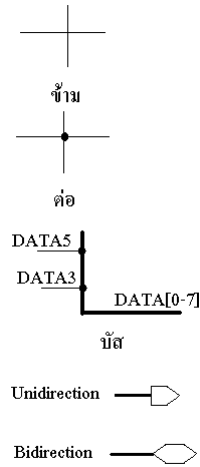
- มีกฎเกณฑ์ ระดับ active ของสัญญาณเอาต์พุตและสัญญาณอินพุตควรตรงกับระดับ active ของขาลูปกรณ เช่น สัญญาณอินพุต FAIL และ สัญญาณ OVERFLOW ทำงานด้วยระดับ low และเอาต์พุต ERROR ทำงานด้วยระดับ high รูป (ก) เขียนผิด รูป (ข) เขียนถูก



8

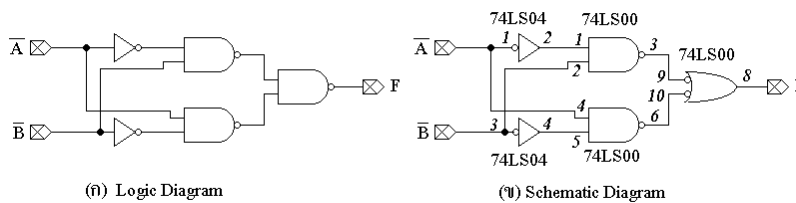
## DRAWING LAYOUTS

- ด้านอินพุตควรอยู่ด้านซ้าย ด้านเอาท์พุตควรอยู่ด้านขวา
- สัญญาณไหลจากซ้ายไปขวา
- สายข้าม
- สายต่อ
- ควรมีชื่อระบุบัส
- สัญญาณที่ดึงออกจากบัสต้องระบุชื่อ
- ทิศทางสัญญาณ - Unidirection  
- Bidirection
- Multiple pages schematics:
  - Flat Structure.
  - Hierarchical Structure.



## Schematic diagram/Logic Diagram

Schematic diagram ควรเขียนแบบ Bubble-to-Bubble Logic และควรประกอบด้วย IC-Type-Logic Family Pin numbers- Pin Diagram

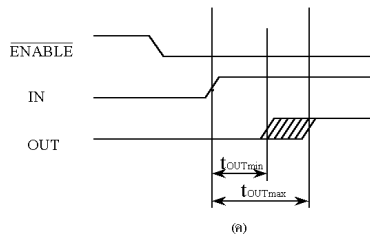
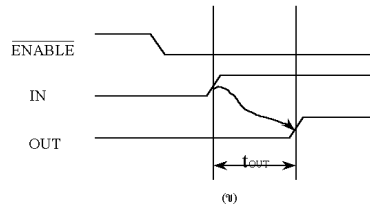
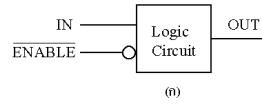


รูปที่ 5-6 การเขียนสัญญาณ

## แผนภาพเวลา (timing diagram)

Delay depends on

- Internal circuit structure
- Logic Family type
- Source Voltage
- Temperature



11

## Combinational Logic Design Structure

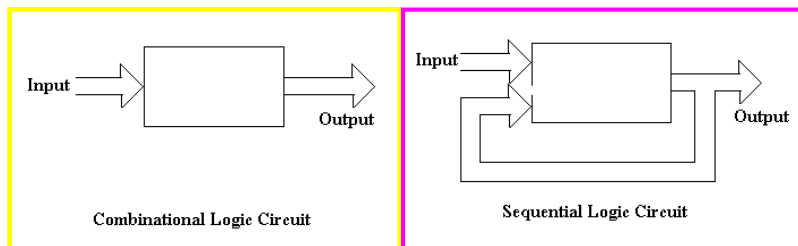
สามารถจัดประเภทของวงจรดิจิทัลได้เป็น 2 แบบใหญ่ๆคือ

**วงจรรอจิกแบบคอมไบเนชันนอล (Combinational Logic Circuit)**

วงจรรอจิกที่ค่าเอาต์พุตขึ้นอยู่กับค่าอินพุตเพียงอย่างเดียว

**วงจรรอจิกแบบซีควนเชียล (Sequential Logic Circuit)**

วงจรรอจิกที่ค่าเอาต์พุตขึ้นอยู่กับค่าอินพุตและสถานะเก่าของเอาต์พุต



12

## การออกแบบวงจรคอมไบเนชันลอจิก

- ขั้นตอนที่ 1 วิเคราะห์ปัญหา และกำหนดตัวแปรและค่าของตัวแปรต่างๆ
- ขั้นตอนที่ 2 เขียนตารางการทำงาน
- ขั้นตอนที่ 3 ลดทอนฟังก์ชัน และเขียนฟังก์ชันลอจิก
- ขั้นตอนที่ 4 เขียนไดอะแกรมวงจร(Circuit Diagram) จากฟังก์ชันในข้อ 3
- ขั้นตอนที่ 5 ทดสอบการทำงาน

13

## ตัวอย่างวงจรคอมไบเนชันลอจิก

- Decoders และ Encoders
- Multiplexers และ Demultiplexers
- Comparators
- Parity Circuits
- Adders/ Subtractors และ Arithmetic Logic Units ( ALUs)

14

## วงจรถอดรหัสและวงจรถ่ายรหัส (DECODER CIRCUIT)

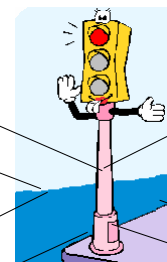
วงจรถอดรหัสไบนารี เป็นวงจรถอดรหัสสัญญาณอินพุตที่เป็นเลขไบนารี ให้เป็นสัญญาณแบบต่างๆ

- 2-to-4 line decoder เป็นวงจรถอดรหัสแบบ อินพุต 2 บิต เอาท์พุท 4 บิต หรือวงจรแปลงเลขฐาน 2 เป็นฐาน 4
- 3-to-8 line decoder เป็นวงจรถอดรหัสแบบ อินพุต 3 บิต เอาท์พุท 8 บิตหรือวงจรแปลงเลขฐาน 2 เป็นฐาน 8
- 4-to-16 line decoder เป็นวงจรถอดรหัสแบบ อินพุต 4 บิต เอาท์พุท 16 บิตหรือวงจรแปลงเลขฐาน 2 เป็นฐาน 8
- BCD TO DECIMAL DECODER วงจรแปลงรหัสบีซีดีเป็นรหัสฐานสิบ
- BCD TO 7 SEGMENT DECODER วงจรแปลงรหัสบีซีดีเป็นรหัสภาคแสดงผล 7 ส่วน

15

## ตัวอย่างการทำงานของระบบควบคุมไฟจราจร

เวลา	แดง	เหลือง	เขียว
0	00	1	0 0
1	01	0	1 0
2	10	0	0 1



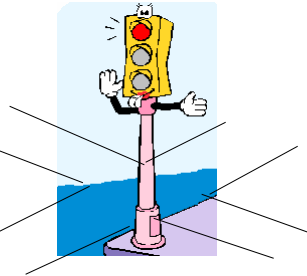
0 แทนไฟดับ

16



## ตัวอย่างการทำงานของระบบควบคุมไฟจราจร

เวลา		แดง	เหลือง	เขียว
0	00	1	0	0
1	01	0	1	0
2	10	0	0	1



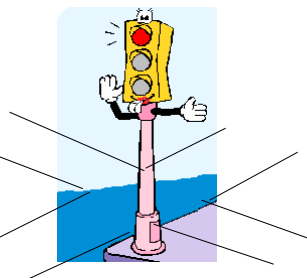
1 แทนไฟติด

0 แทนไฟดับ

17

## ตัวอย่างการทำงานของระบบควบคุมไฟจราจร

เวลา		แดง	เหลือง	เขียว
0	00	1	0	0
1	01	0	1	0
2	10	0	0	1

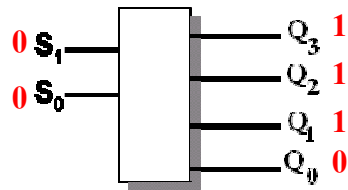


1 แทนไฟติด

0 แทนไฟดับ

18

## 2-TO-4 LINE DECODER

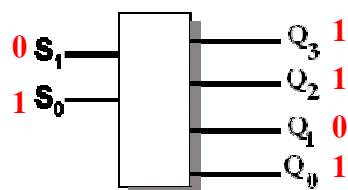


กำหนดให้ เอาต์พุต  $Q_3 - Q_0$  แลคทีฟ 0 โดยจะให้ค่าเป็น 0 เมื่ออินพุต  $S_1 S_0$  มีค่าตรงกับกับเอาต์พุตนั้น

Input		Output			
$S_1$	$S_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	1	1	1	0



## 2-TO-4 LINE DECODER

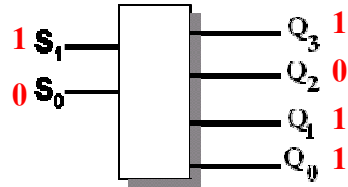


กำหนดให้ เอาต์พุต  $Q_3 - Q_0$  แลคทีฟ 0 โดยจะให้ค่าเป็น 0 เมื่ออินพุต  $S_1 S_0$  มีค่าตรงกับกับเอาต์พุตนั้น

Input		Output			
$S_1$	$S_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	1	1	1	0
0	1	1	1	0	1



## 2-TO-4 LINE DECODER

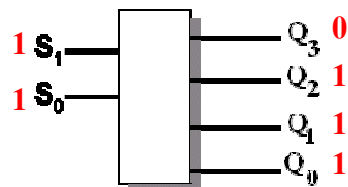


กำหนดให้ เอาท์พุท  $Q_3 - Q_0$  แลคทีฟ 0 โดยจะให้ค่าเป็น 0 เมื่ออินพุท  $S_1 S_0$  มีค่าตรงกับกับเอาท์พุทนั้น

Input		Output			
$S_1$	$S_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1



## 2-TO-4 LINE DECODER

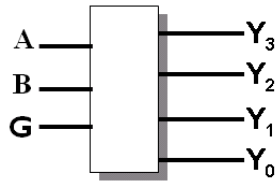


กำหนดให้ เอาท์พุท  $Q_3 - Q_0$  แลคทีฟ 0 โดยจะให้ค่าเป็น 0 เมื่ออินพุท  $S_1 S_0$  มีค่าตรงกับกับเอาท์พุทนั้น

Input		Output			
$S_1$	$S_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1



## 2-TO-4 LINE DECODER แบบมีสัญญาณควบคุม



กำหนดให้ เอาต์พุต  $Y_3 - Y_0$  แอคทีฟ 1 โดยจะให้ค่าเป็น 1 เมื่ออินพุต S1 S0 มีค่าตรงกับกับเอาต์พุตนั้น ทั้งนี้ อินพุต G ต้องเป็น 1 นอกจากนี้แล้วเอาต์พุตทั้งหมดจะเป็น 0

FUNCTION TABLE

INPUT			OUTPUT			
G	B	A	$Y_0$	$Y_1$	$Y_2$	$Y_3$
L	X	X	L	L	L	L
H	L	L	H	L	L	L
H	L	H	L	H	L	L
H	H	L	L	L	H	L
H	H	H	L	L	L	H

$$Y_0 = G\bar{B}\bar{A}$$

$$Y_1 = G\bar{B}A$$

$$Y_2 = GB\bar{A}$$

$$Y_3 = GBA$$

23

## 2-TO-4 LINE DECODER

FUNCTION TABLE

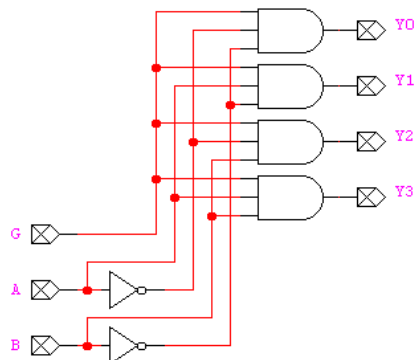
INPUT			OUTPUT			
G	B	A	$Y_0$	$Y_1$	$Y_2$	$Y_3$
L	X	X	L	L	L	L
H	L	L	H	L	L	L
H	L	H	L	H	L	L
H	H	L	L	L	H	L
H	H	H	L	L	L	H

$$Y_0 = G\bar{B}\bar{A}$$

$$Y_1 = G\bar{B}A$$

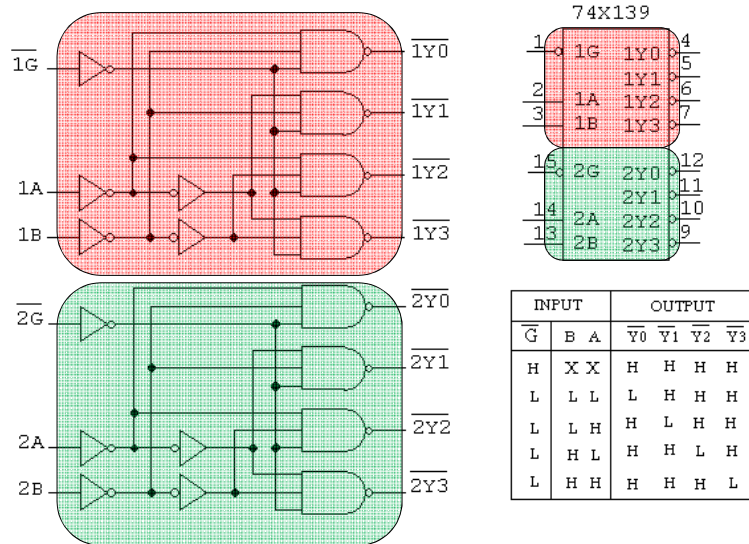
$$Y_2 = GB\bar{A}$$

$$Y_3 = GBA$$



24

## 74X139 2-TO-4 LINE DECODER



## 3-TO-8 DECODER

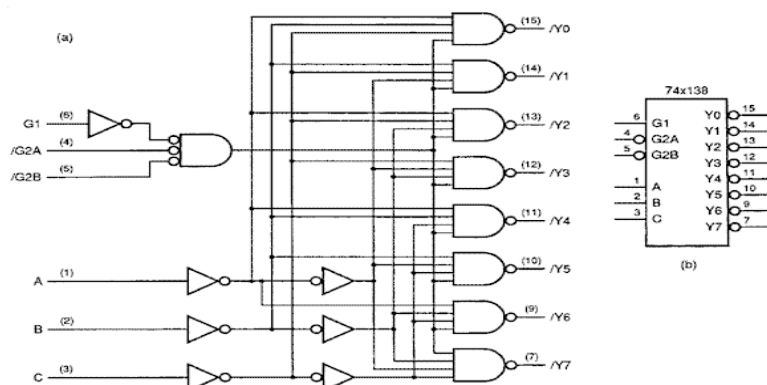


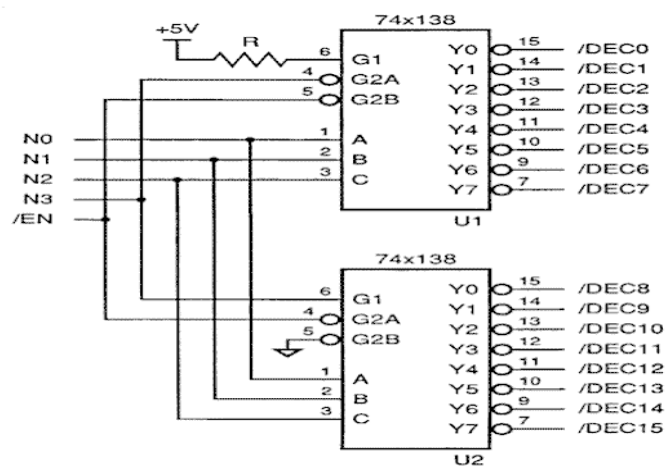
Figure 5-28 The 74x138 3-to-8 decoder: (a) logic diagram, including pin numbers for a standard 16-pin dual in-line package; (b) traditional logic symbol.

74x138 Truth table

Inputs						Outputs							
G1	/G2A	/G2B	C	B	A	/Y7	/Y6	/Y5	/Y4	/Y3	/Y2	/Y1	/Y0
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

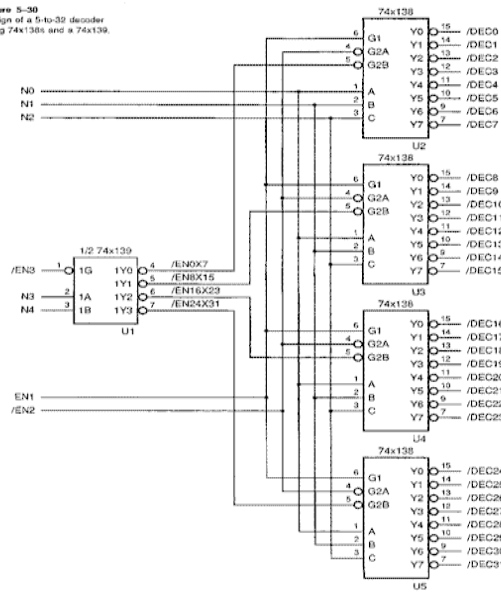
27

### CASCADING DECODERS: 4-TO-16 DECODER



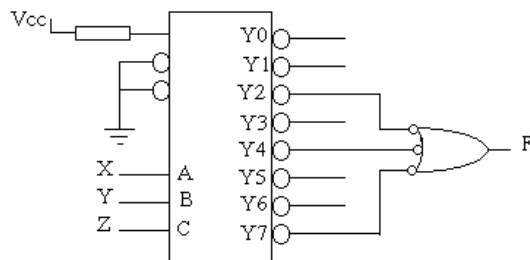
Cascading Decoders:  
5-to-32 Decoder

Figure 5-30  
Design of a 5-to-32 decoder  
using 74x138s and a 74x139.



IMPLEMENTING THE CANONICAL SUM

$$F = \sum_{X,Y,Z} (2,4,7)$$



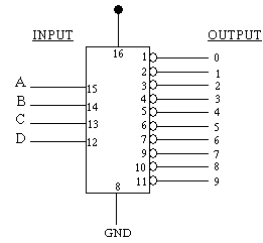
## THE BCD TO DECIMAL DECODER

การทำงาน

- สัญญาณในรูปรหัสบิิต (BCD) ที่อินพุท จะทำให้เอาต์พุท ที่สอดคล้องกับรหัสนั้น แอดตีฟคือมีลอจิกเป็น '0' เอาต์พุทอื่นๆ เป็นลอจิก '1' เช่นถ้าอินพุทเป็น "0101" เอาต์พุทจะเป็น "111101111"

เอาต์พุท 9

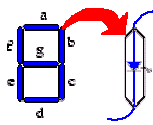
เอาต์พุท 0



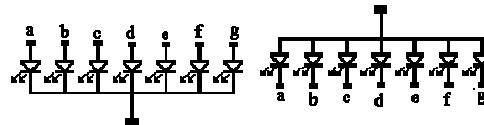
31

## ตัวถอดรหัสแอลอีดี 7 ส่วน

แอลอีดี 7 ส่วน (7 Segment LEDs)

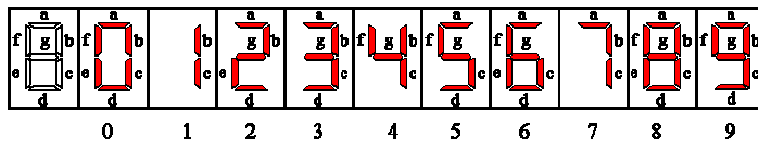


One segment enlarged



Common Cathode

Common Anode

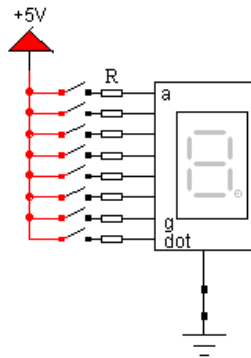


32



## แอลอีดี 7 ส่วน (7 SEGMENT LEDs )

รูปแบบ	a	b	b	d	e	f	g	dot
ดับ	0	0	0	0	0	0	0	0
ดับ	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
3	1	1	1	1	0	0	1	0
4	0	1	1	0	0	1	1	0
5	1	0	1	1	0	1	1	0
6	1	0	1	1	1	1	1	0
7	1	1	1	0	0	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	1	1	0	1	1	0

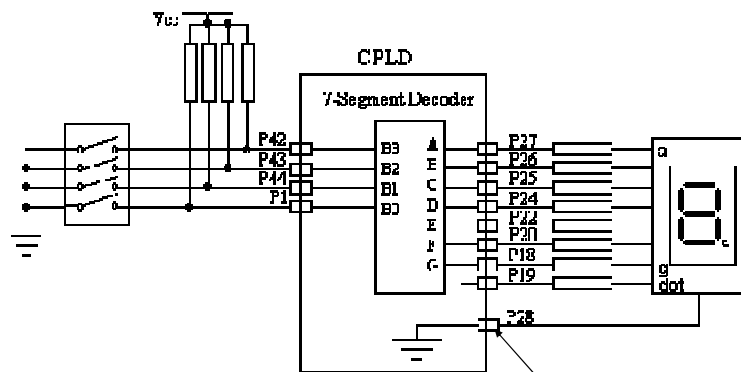


1 แทน LED ติด

0 แทน LED ดับ

ขาคาโอดไม่ต่อไปที่ไฟลบ

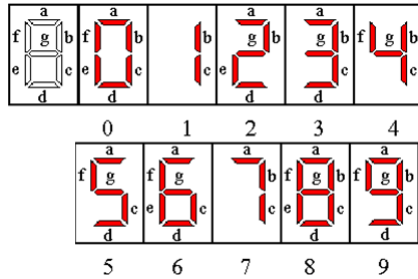
## ตัวถอดรหัสแอลอีดี 7 ส่วน



เนื่องจากบอร์ดทดลอง  
ต่ออยู่แล้ว

## ฟังก์ชันของตัวถอดรหัสแอลอีดี 7 ส่วน

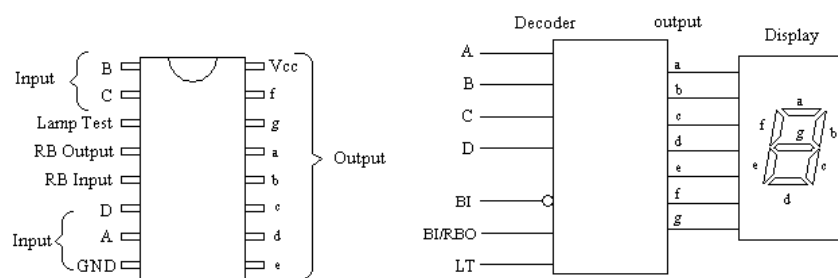
รูปแบบ	b3	b2	b1	b0	เซกเมนต์ a
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
ดับ	1	x	1	x	0



$$\begin{aligned}
 \text{seg}_a &= \sum m(0, 2, 3, 5, 6, 7, 8, 9) \\
 &= (b3.\overline{b2}.\overline{b1}) + (\overline{b3}.b2.b0) + (\overline{b2}.\overline{b1}.\overline{b0}) + (\overline{b3}.b1)
 \end{aligned}$$

35

## ไอซี 7447 BCD TO 7 SEGMENT DECODER



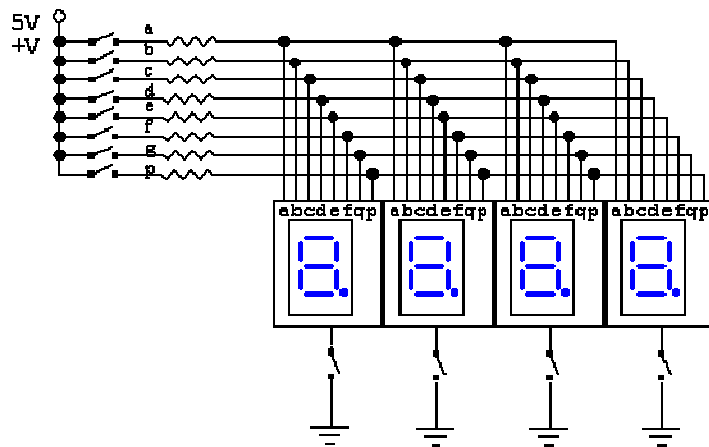
**Lamp Test (LT)** เมื่อเป็น 0 ทุกเซกเมนต์จะ ON และ BI/RBO จะเป็น 1

**ripple Blank Input (RBI)** ถ้า LT = 1, Rbi = 0 และ A = B = C = D = 0 ทุกเซกเมนต์จะ OFF

**Blanking Input/Ripple Blanking Output (BI/RBO)** สัญญาณนี้เป็นทั้งอินพุตและเอาต์พุตใช้สำหรับการต่อภาคแสดงผลหลายๆหลัก โดยต้องการจะ Blank ชุดตำแหน่งสูง

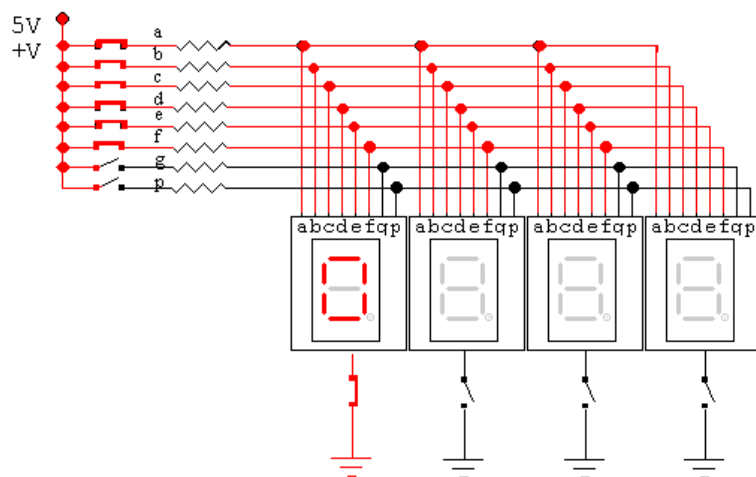
36

## ตัวอย่างการใช้งานแอลอีดี 7 ส่วน 4 หลัก



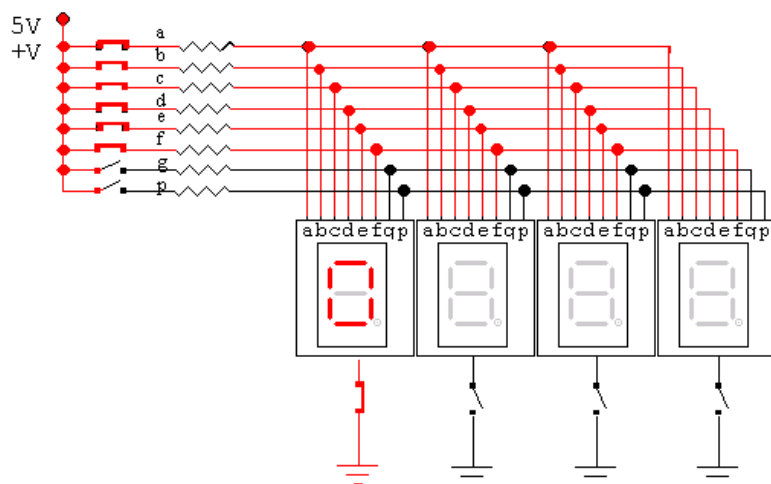
37

## ตัวอย่างการใช้งานแอลอีดี 7 ส่วน 4 หลัก

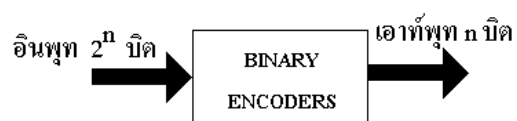


38

## ตัวอย่างการใช้งานแอลอีดี 7 ส่วน 4 หลัก

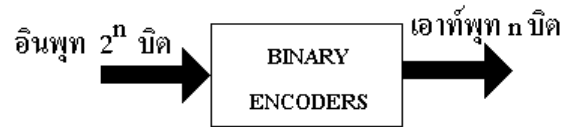


## วงจรเข้ารหัส(ENCODER CIRCUIT)



วงจรเข้ารหัสจะทำงานตรงข้ามกับวงจรถอดรหัส ใน  
ที่นี่จะกล่าวถึงวงจรเข้ารหัสแบบไบนารี (Binary  
Encoder) โดยสัญญาณอินพุตจะเป็นสัญญาณอะไรก็ได้  
แต่สัญญาณเอาต์พุตจะออกมาเป็นสัญญาณไบนารี  
จำนวนบิตของอินพุต จะมีค่าเท่ากับ  $2$  โดย  $n$  คือ  
จำนวนบิตของเอาต์พุต

## วงจรเข้ารหัส(ENCODER CIRCUIT)

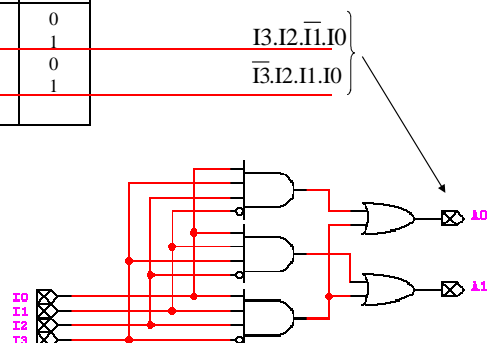
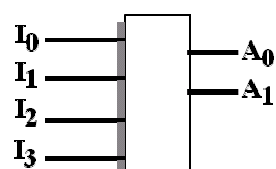


- 4-to-2 line encoder เป็นวงจรเข้ารหัสแบบ อินพุต 4 บิต เอาต์พุต 2 บิต หรือวงจรแปลงเลขฐาน 4 เป็นเลขฐาน 2
- 4-to-2 line priority encoder เป็นวงจรเข้ารหัสแบบ อินพุต 4 บิต เอาต์พุต 2 บิต เหมือนแบบแรกแต่มีการจัดลำดับความสำคัญของอินพุต
- วงจรเข้ารหัสแบบอื่นๆ วงจรเข้ารหัสคีย์ (The Key Encoder)

41

## วงจรเข้ารหัสเข้า 4 ออก 2 (4-TO-2 LINE BINARY ENCODER)

INPUT				OUTPUT	
I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>	A <sub>1</sub>	A <sub>0</sub>
1	1	1	0	0	0
1	1	0	1	0	1
1	0	1	1	1	0
0	1	1	1	1	1



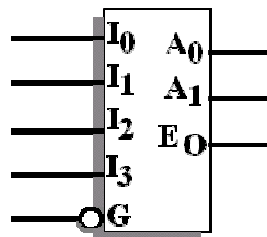
## วงจรเข้ารหัสแบบไพโรอริตี้(PRIORITY ENCODER)

Input					Output		
G	I3	I2	I1	I0	EO	A1	A0
0	1	1	1	0	1	0	0
0	1	1	0	X	1	0	1
0	1	0	X	X	1	1	0
0	0	X	X	X	1	1	1
1	X	X	X	X	0	1	1

$$A_0 = \overline{G}(I_3.I_2.\overline{I_1} + \overline{I_3}) + G$$

$$A_1 = \overline{G}(I_3.\overline{I_2} + \overline{I_3}) + G$$

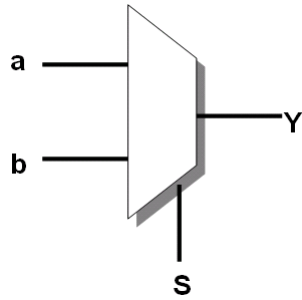
$$EO = \overline{G}$$



## วงจรมัลติเพลกซ์หรือวงจรเลือกข้อมูล

- วงจรมัลติเพลกซ์หรือวงจรเลือกข้อมูลจะทำหน้าที่ในการเลือกข้อมูลจากข้อมูลอินพุตแล้วส่งออกไปยังเอาต์พุต การจะเลือกเอาข้อมูลจากอินพุตใด กระทำโดยการกำหนดค่าแอดเดรสที่สายเลือก (selector)
- วงจรมัลติเพลกซ์นั้นมีอยู่หลายแบบ ขึ้นอยู่กับจำนวนของอินพุต ในที่นี้เราจะศึกษาเพียง 2 แบบ คือแบบ 2 อินพุต กับแบบ 4 อินพุต

## 2-TO-1 MULTIPLEXER



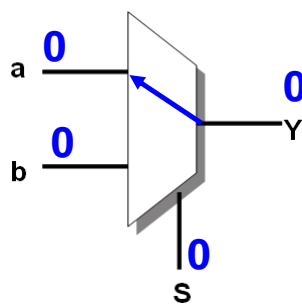
A b เป็นสัญญาณอินพุต  
S เป็นสัญญาณเลือก  
Y เป็นสัญญาณเอาต์พุต

ถ้า  $S = 0$  ให้  $Y = a$

ถ้า  $S = 1$  ให้  $Y = b$

45

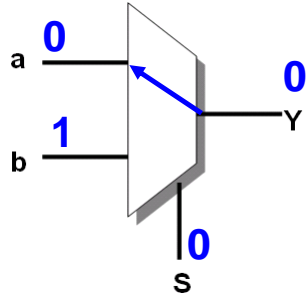
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0

46

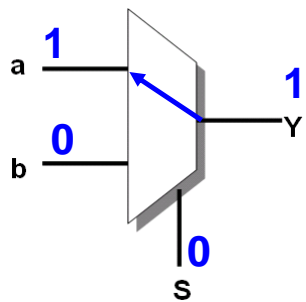
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0

47

## 2-TO-1 MULTIPLEXER

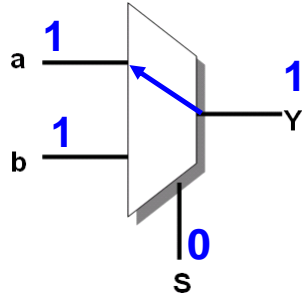


S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1

48



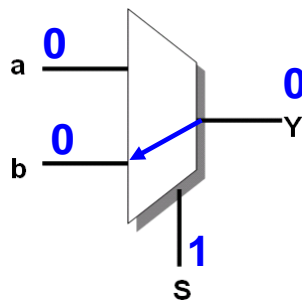
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1

49

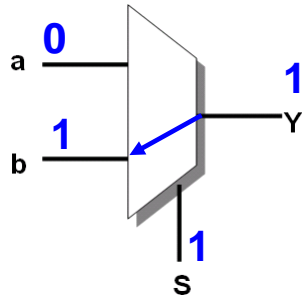
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0

50

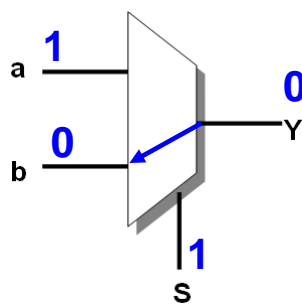
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

51

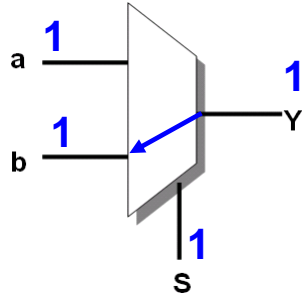
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0

52

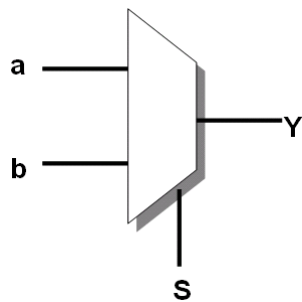
## 2-TO-1 MULTIPLEXER



S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

53

## 2-TO-1 MULTIPLEXER

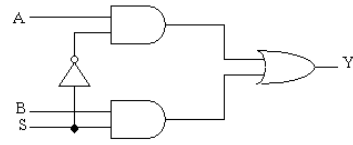


S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

54

## 2-TO-1 MULTIPLEXER

S	a	b	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



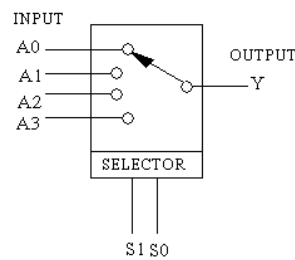
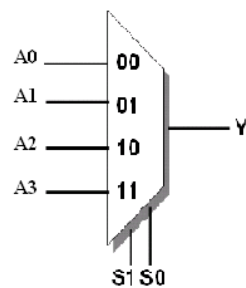
$$Y = \bar{S}a + Sb$$

ถ้า  $S = 0$  ให้  $Y = a$

ถ้า  $S = 1$  ให้  $Y = b$

55

## 4-TO-1 MULTIPLEXER



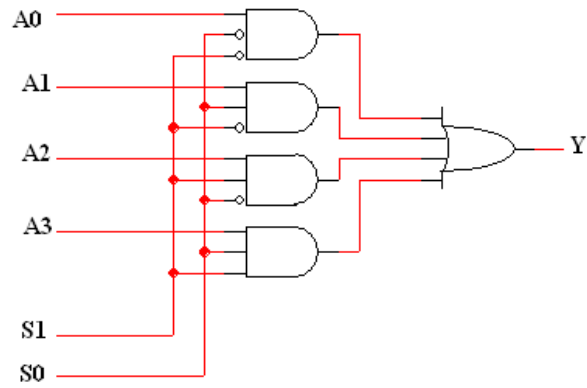
Input		Output
S1	S0	Y
0	0	Y = A0
0	1	Y = A1
1	0	Y = A2
1	1	Y = A3

$$Y = \bar{S}_1 \bar{S}_0 A_0 + \bar{S}_1 S_0 A_1 + S_1 \bar{S}_0 A_2 + S_1 S_0 A_3$$

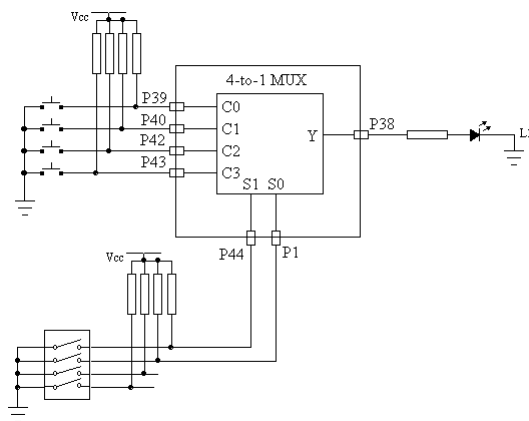
56

## 4-TO-1 MULTIPLEXER

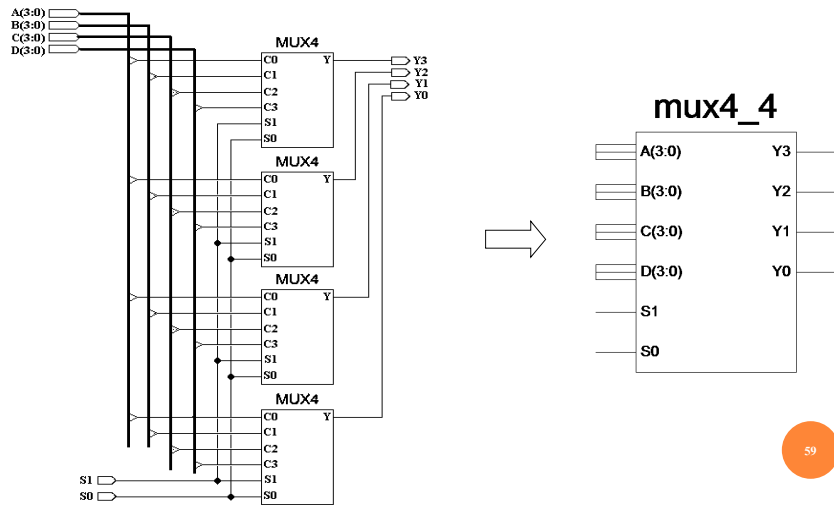
$$Y = \overline{S1.S0}.A0 + \overline{S1}.S0.A1 + S1.\overline{S0}.A2 + S1.S0.A3$$



## 4-TO-1 MULTIPLEXER



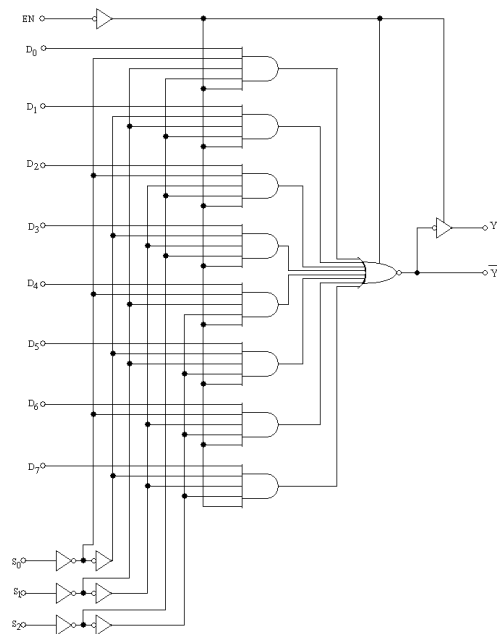
## 4-TO-1 MULTIPLEXER 4 ชุด



59

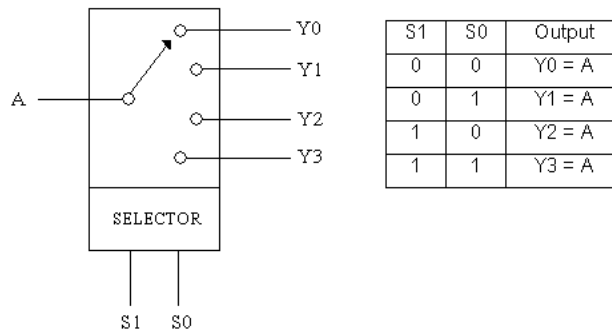
## วงจรมัลติเพล็กซ์แบบ 8 อินพุต

แผนผังโลจิก (LOGIC DIAGRAM)  
ของ 74251

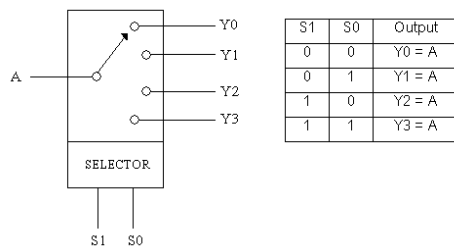


# ดีมัลติเพลกซ์ (DEMULTIPLEXER)

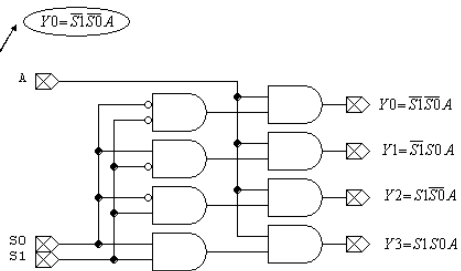
วงจรดีมัลติเพลกซ์แบบ 4 เอาท์พุท (1-To-4 Demultiplexer)



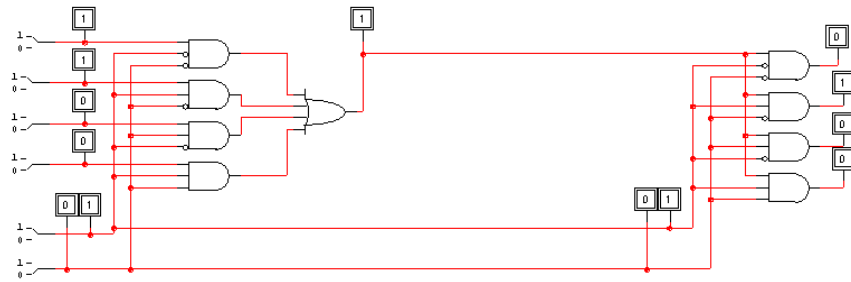
61



Input			Output			
S1	S0	A	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

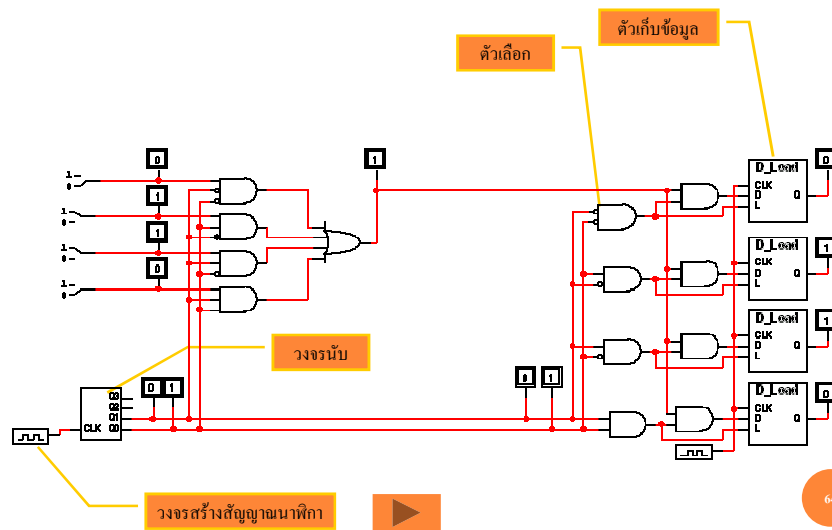


## ตัวอย่างการใช้งาน MUX และ DEMUX



63

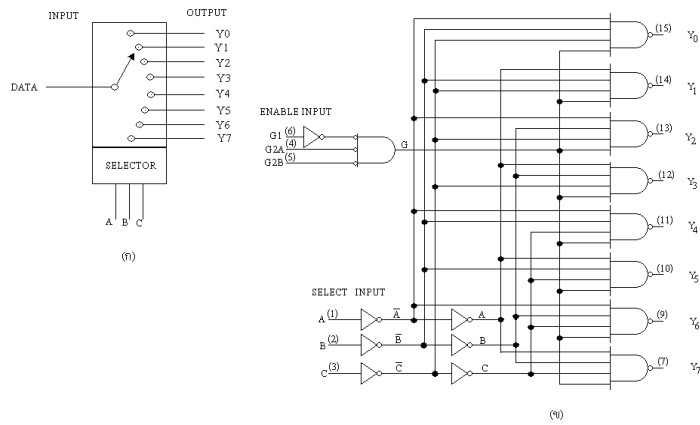
## ตัวอย่างการใช้งาน MUX และ DEMUX (ต่อ)



64



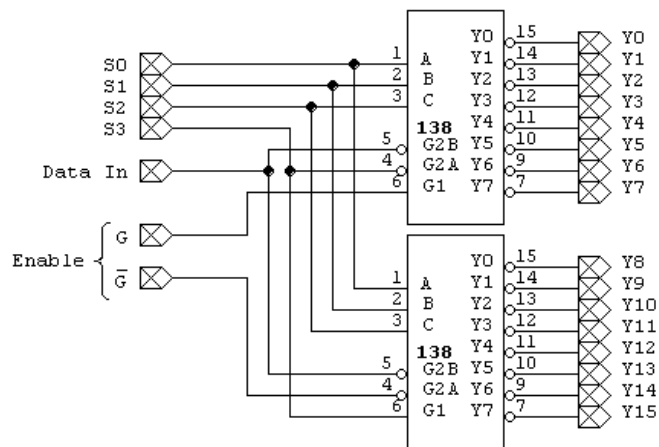
## ดีมีคิตีเพลกซ์แบบ 8 เอาท์พุท



แผนผังบล็อกแสดงการทำงาน และแผนผัง โลจิกของ 74LS138

65

ตัวอย่างการประยุกต์ใช้ ดีมีคิตีเพลกซ์ แบบ 8 เอาท์พุททำเป็นดีมีคิตีเพลกซ์แบบ 16 เอาท์พุท



66