UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SKEE1223: Digital Electronics

9 – Medium Scale Integrated (MSI) Circuits

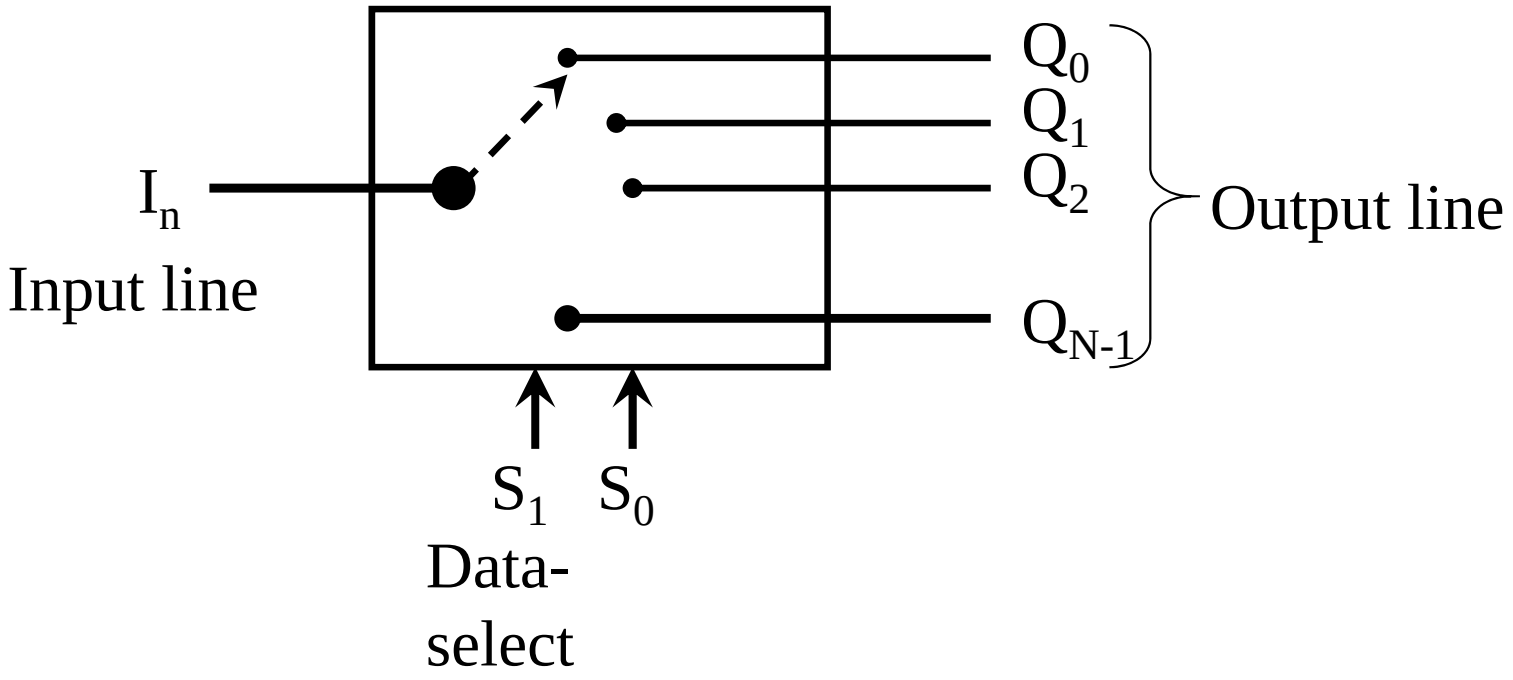
Dr Michael Tan Loong Peng
PhD (Cambridge)
Senior Lecturer
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MSI Circuits

- Multiplexers (Mux)
  - 2x1, 4x1, and 8x1 muxes
  - 74x151, 74x153, 74x157 devices
- Demultiplexers (Demux), Decoders, and Encoders
  - 74x138 and 74x139 decoders
  - Encoder, priority encoder and the 75x147 devices
  - BCD to 7-segment decoder and the 74x247 devices
  - Logic functions using muxes and decoders
- Adders and Comparators
  - Half, full and ripple carry adders
  - The 74x83 devices
  - Comparator and the 74x85 devices

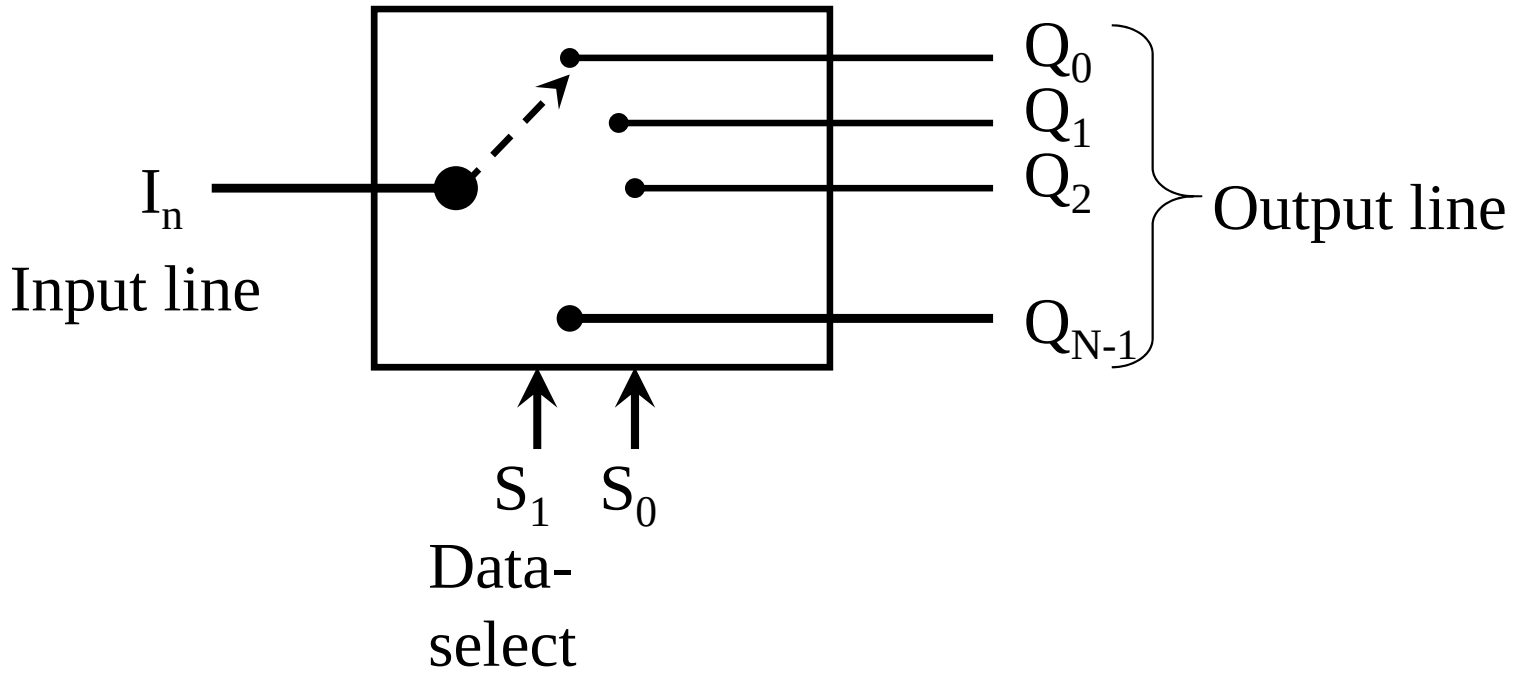
Demultiplexer

- Demultiplexer (DEMUX) function is opposite to the MUX.
- It allows the single input line data to be routed to any output line.



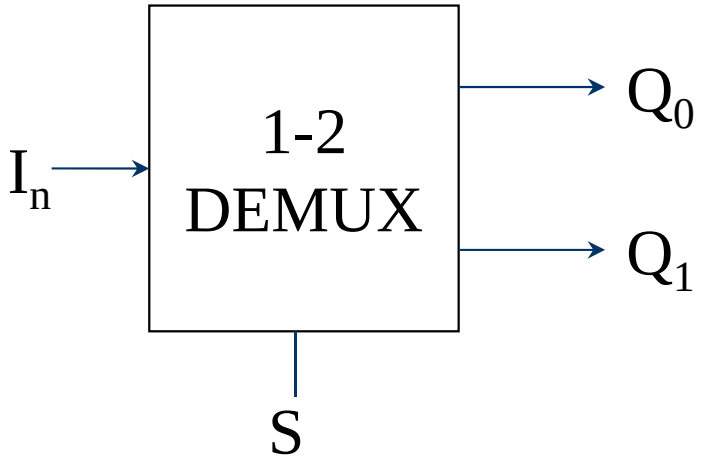
Demultiplexer

- DEMUX has only one input line (I_n) and several output lines (Q_0, Q_1, Q_2.....Q_{N-1}).
- Data-select input (S_1 and S_0) is used to select which output line should be switched from the input line.



Demultiplexer

- 2-output Demultiplexer (1-2 DEMUX)



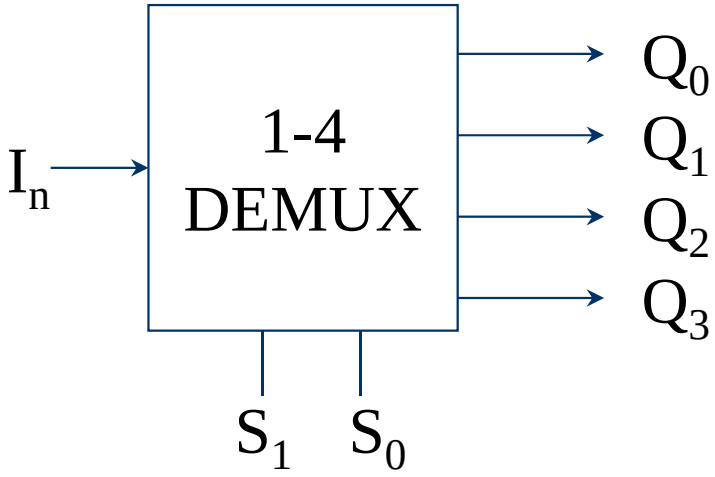
S	Q_1	Q_0
0	0	I_n
1	I_n	0

$$Q_0 = \overline{S} \cdot I_n$$

$$Q_1 = S \cdot I_n$$

Demultiplexer

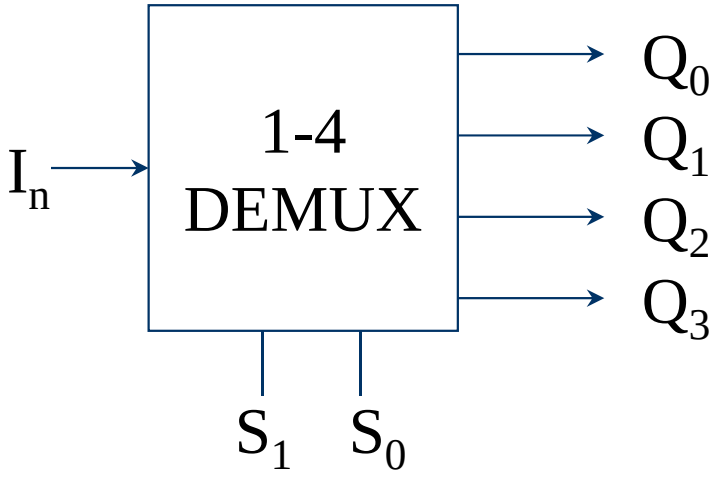
- 4-output Demultiplexer (1-4 DEMUX)



$S_1 S_0 I_n$	Q_3	Q_2	Q_1	Q_0
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

Demultiplexer

- 4-output Demultiplexer (1-4 DEMUX)-in simplified truth table



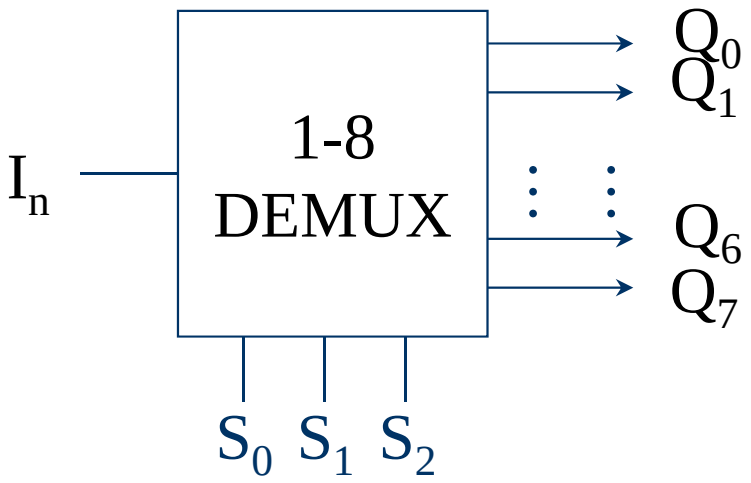
S_1S_0	Q_3	Q_2	Q_1	Q_0
0 0	0	0	0	I_n
0 1	0	0	I_n	0
1 0	0	I_n	0	0
1 1	I_n	0	0	0

$$Q_0 = \overline{S}_1\overline{S}_0 I_n \qquad Q_2 = S_1\overline{S}_0 I_n$$

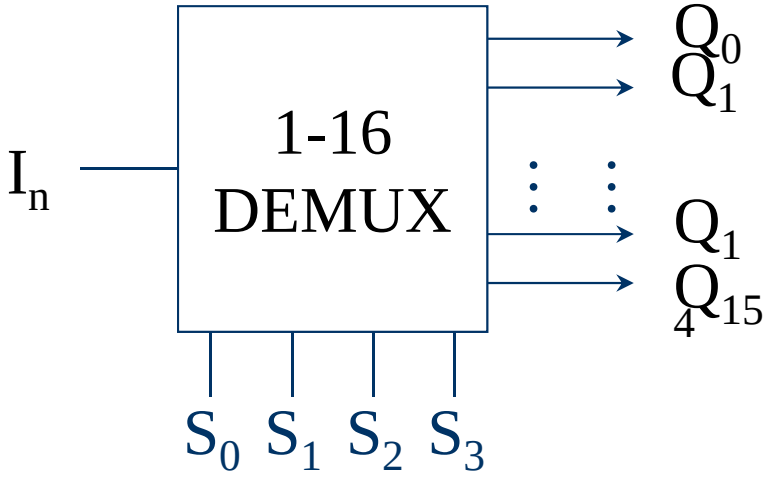
$$Q_1 = \overline{S}_1 S_0 I_n \qquad Q_3 = S_1 S_0 I_n$$

Demultiplexer

- Design 8-output Demultiplexer (1-to-8 DEMUX)



- Design 16-output Demultiplexer (1-to-16 DEMUX)



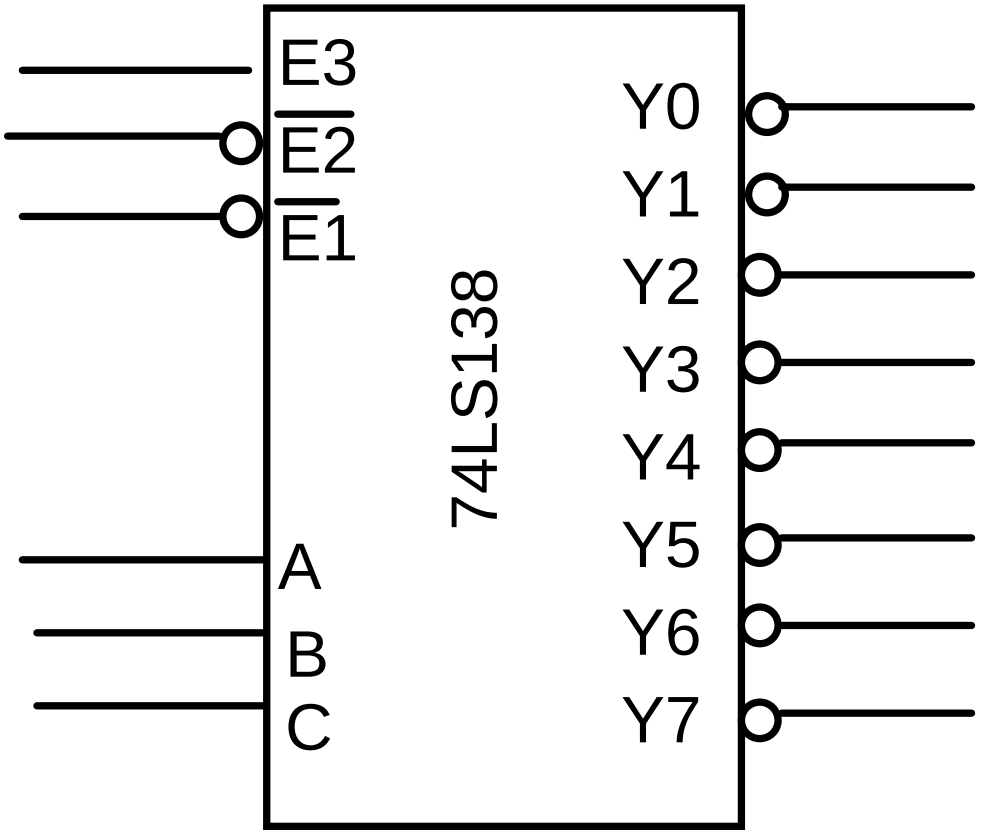
Demultiplexer

- Use 1-2 DEMUX to implement 1-4 DEMUX.
- Use 1-2 DEMUX to implement 1-8 DEMUX.
- Use 1-2 DEMUX to implement 1-16 DEMUX.

- Use 1-2 DEMUX and 1-4 DEMUX to implement 1-8 DEMUX.
- Use 1-4 DEMUX to implement 1-16 DEMUX.

Demultiplexer

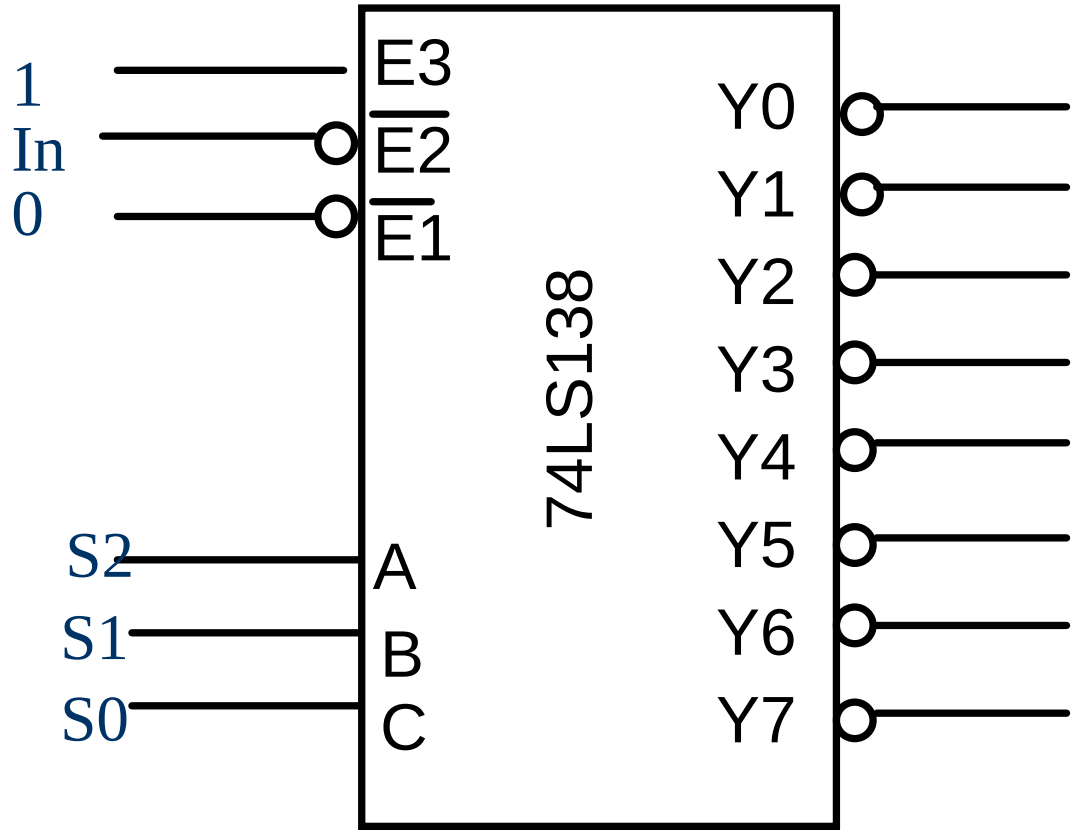
- Use 74138 Decoder to Implement 1-8 DEMUX



- Use 74138 Decoder and NOT gate to Implement 1-16 DEMUX

Solution Demultiplexer

- Use 74138 Decoder to Implement 1-8 DEMUX



Decoder

- Circuit that accepts a set of input (in binary number/code) and execute the output or a set of outputs corresponding to that particular input.
- For example, the 3-line-to-8-line ‘1-of-8’ Decoder. For 1-of-8 Decoder only one output is activated at one time, corresponding to the input code. This decoder has an active-HIGH input and output.

Decoder

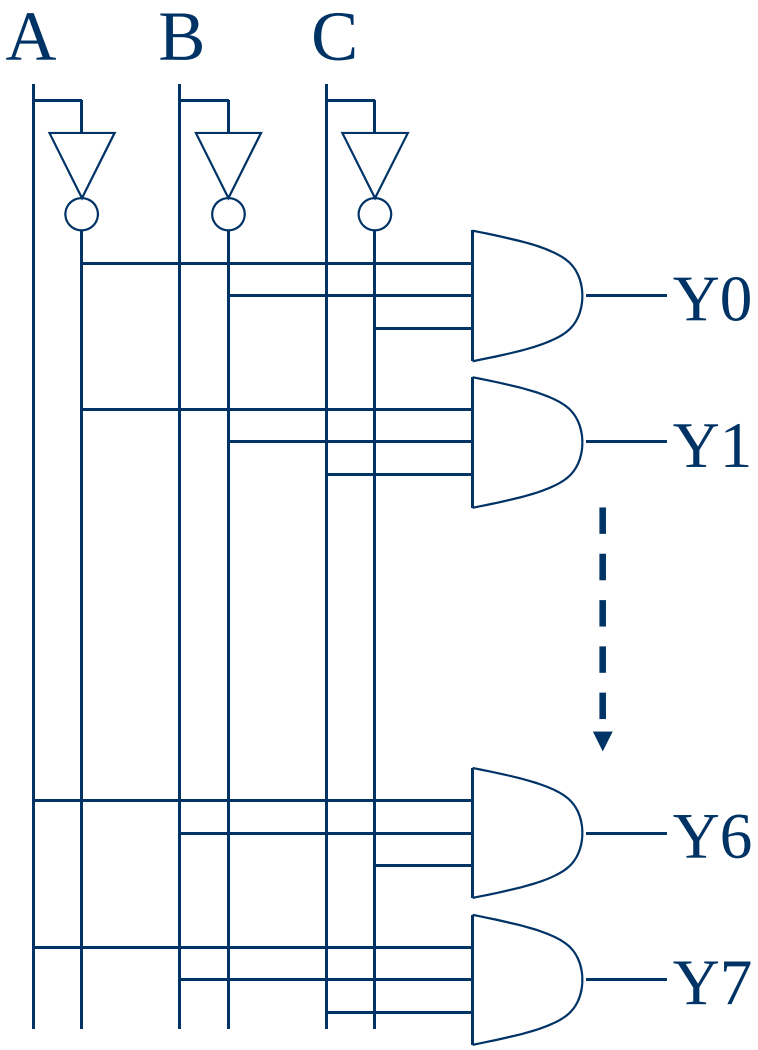
- Truth table for 1-of-8 Decoder

A	B	C	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
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								1

- $Y0 = \bar{A}\bar{B}\bar{C}$
- $Y1 = \bar{A}\bar{B}C$

Decoder

- 1-of-8 Decoder logic circuit

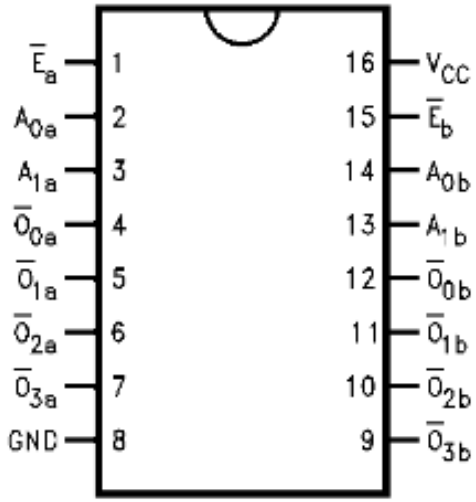


Decoder

- The 74138 Decoder (1-of-8 Decoder)
- Enable the input pin.
- Active low Enable, active low output

Dual 2-to-4 Decoder

Connection Diagram



Pin Description

Pin Names	Description
A_0, A_1	Address Inputs
$\overline{E}$	Enable Inputs
$\overline{O}_0$–$\overline{O}_3$	Outputs

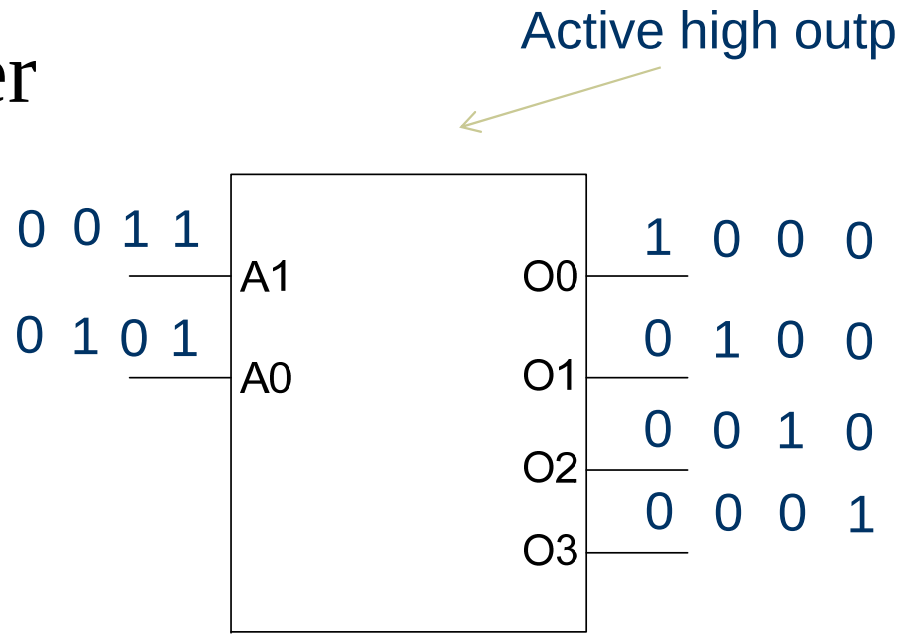
Truth Table

Inputs			Outputs			
$\overline{E}$	A_0	A_1	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

Decoders

- 2x4 Decoder



Function:

$O_3O_2O_1O_0$ = 0001 when A_1A_0 = 00 $\longrightarrow$ $O0 = \overline{A1} \cdot \overline{A0}$

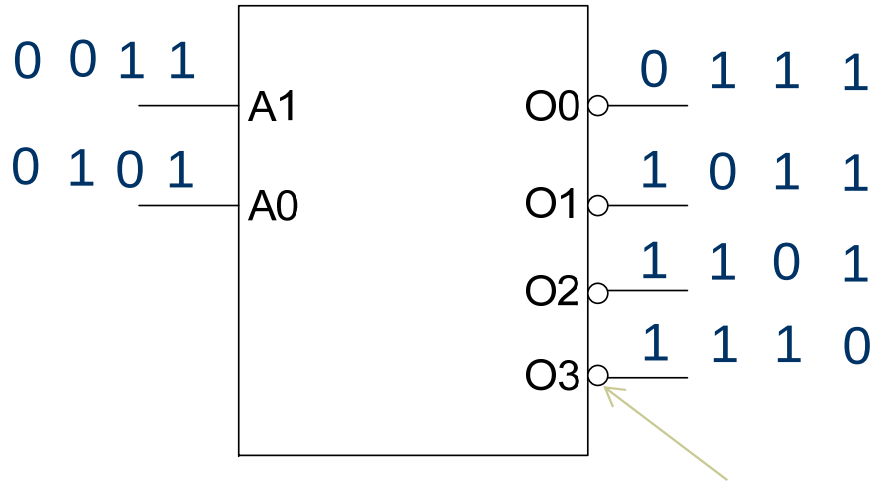
$O_3O_2O_1O_0$ = 0010 when A_1A_0 = 01 $\longrightarrow$ $O1 = \overline{A1} \cdot A0$

$O_3O_2O_1O_0$ = 0100 when A_1A_0 = 10 $\longrightarrow$ $O2 = A1 \cdot \overline{A0}$

$O_3O_2O_1O_0$ = 1000 when A_1A_0 = 11 $\longrightarrow$ $O3 = A1 \cdot A0$

Decoders (cont.)

- Decoders are typically designed as active low



Bubble at output denotes active low output

Function:

$O_3O_2O_1O_0 = 1110$ when $A_1A_0 = 00$ $\rightarrow$ $O0 = \overline{\overline{A1} \cdot \overline{A0}}$

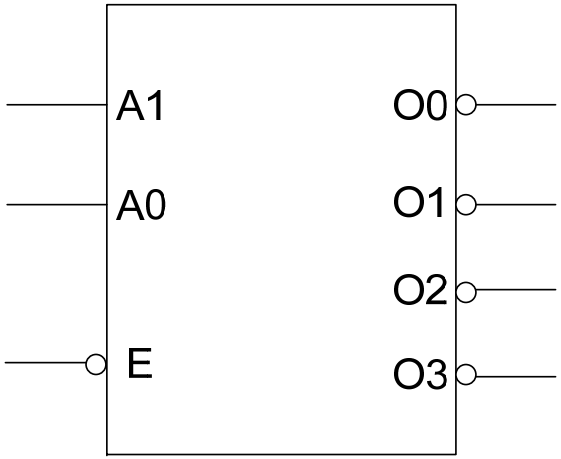
$O_3O_2O_1O_0 = 1101$ when $A_1A_0 = 01$ $\rightarrow$ $O1 = \overline{A1 \cdot \overline{A0}}$

$O_3O_2O_1O_0 = 1011$ when $A_1A_0 = 10$ $\rightarrow$ $O2 = \overline{\overline{A1} \cdot A0}$

$O_3O_2O_1O_0 = 0111$ when $A_1A_0 = 11$ $\rightarrow$ $O3 = \overline{A1 \cdot A0}$

Decoders (cont.)

- 2x4 Decoder with active low enable



Function:

If (E = 1),

$O_3O_2O_1O_0 = 1111$, $A_1A_0 = xx$

If (E = 0),

$O_3O_2O_1O_0 = 1110$ when $A_1A_0 = 00$

$O_3O_2O_1O_0 = 1101$ when $A_1A_0 = 01$

$O_3O_2O_1O_0 = 1011$ when $A_1A_0 = 10$

$O_3O_2O_1O_0 = 0111$ when $A_1A_0 = 11$

Can you draw the logic circuit?

$$O0 = \overline{\overline{E} \cdot \overline{A1 \cdot A0}}$$

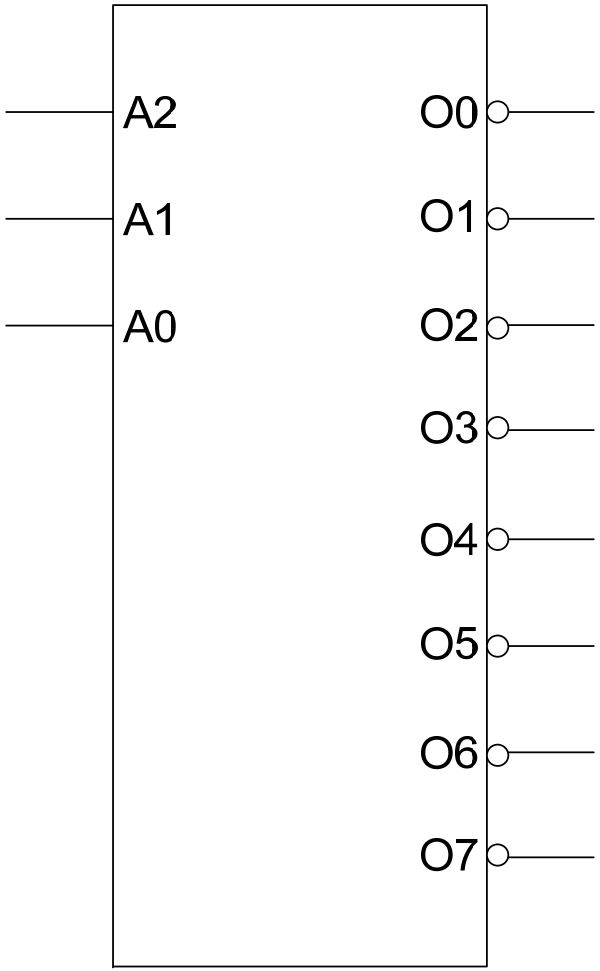
$$O1 = \overline{\overline{E} \cdot \overline{A1 \cdot A0}}$$

$$O2 = \overline{\overline{E} \cdot \overline{A1 \cdot A0}}$$

$$O3 = \overline{\overline{E} \cdot \overline{A1 \cdot A0}}$$

Decoders (cont.)

- 3x8 Decoder



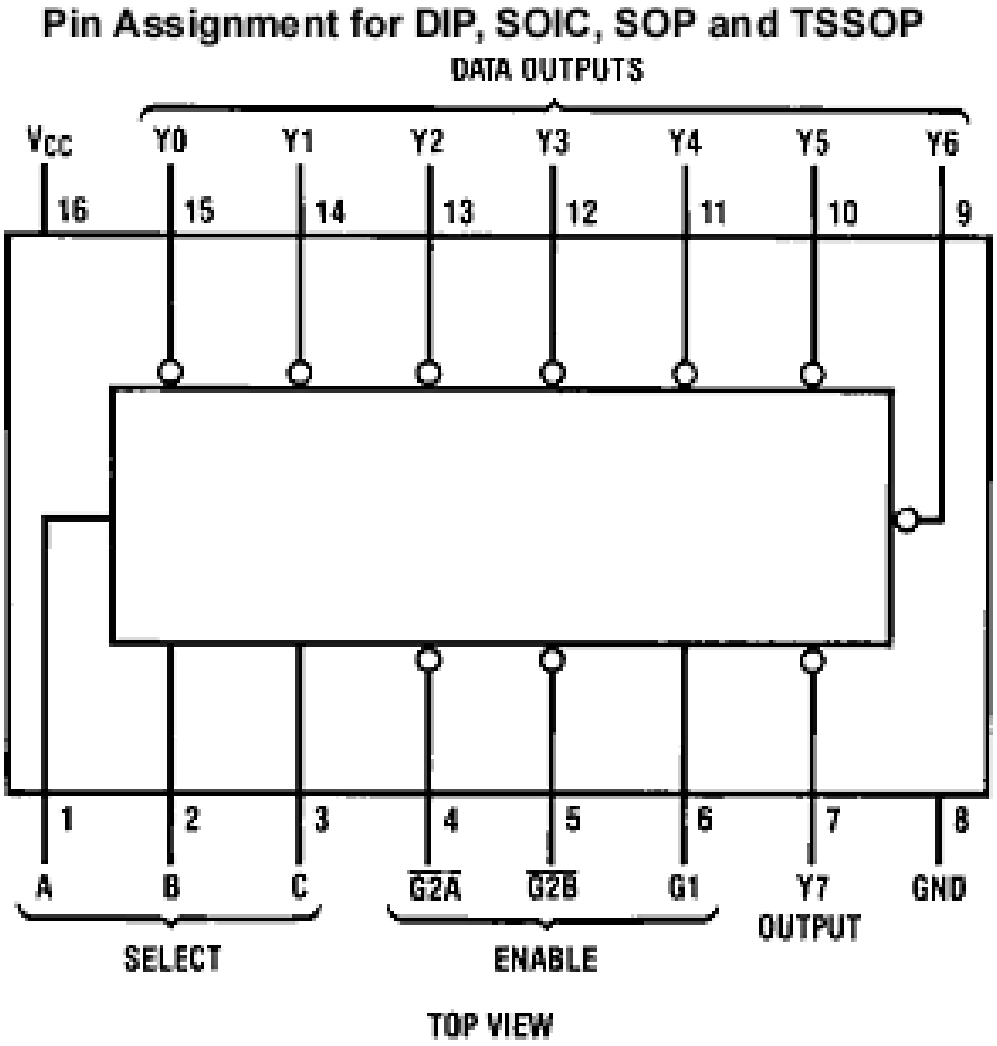
Function:

$O_7O_6O_5O_4O_3O_2O_1O_0 = 1111\ 1110$ when $A_2A_1A_0 = 000$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1111\ 1101$ when $A_2A_1A_0 = 001$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1111\ 1011$ when $A_2A_1A_0 = 010$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1111\ 0111$ when $A_2A_1A_0 = 011$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1110\ 1111$ when $A_2A_1A_0 = 100$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1101\ 1111$ when $A_2A_1A_0 = 101$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 1011\ 1111$ when $A_2A_1A_0 = 110$
$O_7O_6O_5O_4O_3O_2O_1O_0 = 0111\ 1111$ when $A_2A_1A_0 = 111$

How does the logic circuit looks like?
- Can we use K-Map?

Decoders (cont.)

- 74138 Single 3x8 Decoder



Contains one active low 3x8 decoder with three enable inputs (G2A, G2B, G1)

Decoders (cont.)

- 74138 Single 3x8 Decoder (cont.)

Truth Table

Inputs					Outputs							
Enable		Select										
G1	$\overline{G2}$ (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

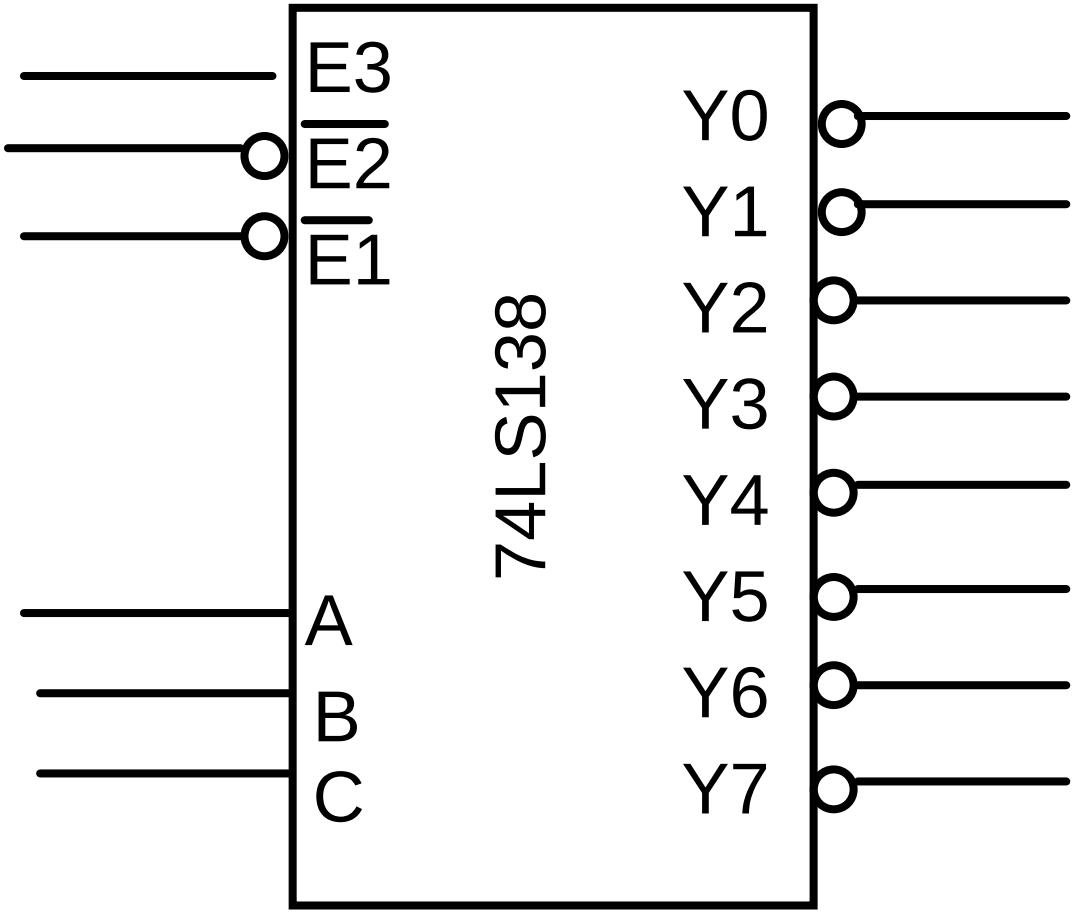
H = HIGH Level, L = LOW Level, X = don't care

Note 1: $\overline{G2} = G2A + G2B$

Output is enabled only when G1 = 1, $\overline{G2A}$ = 0, $\overline{G2B}$ = 0

Decoder

- 74138 Decoder chip



Decoder

- 74138 Decoder

Enabled I/P			Input			Output							
$\overline{E_1}$	$\overline{E}$	E_3	A	B	C	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$	$\overline{Y_4}$	$\overline{Y_5}$	$\overline{Y_6}$	$\overline{Y_7}$
1	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	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Decoders (cont.)

- 74x139 Dual 2x4 Decoder

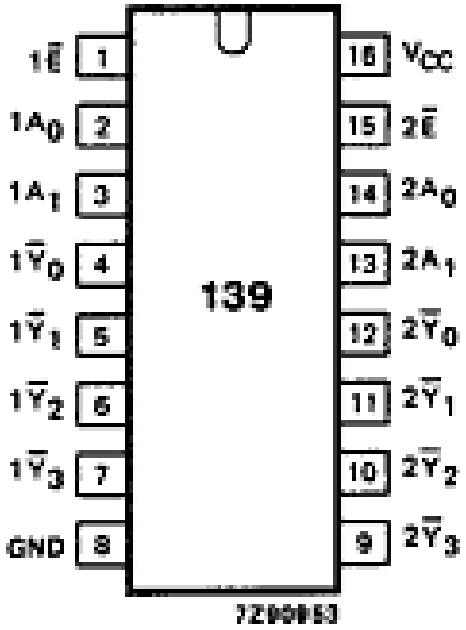


Fig.1 Pin configuration.

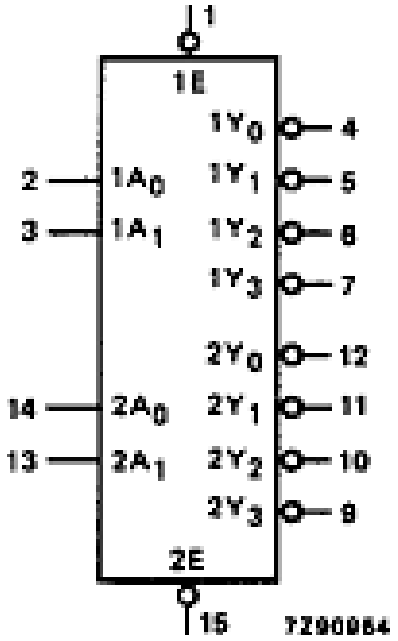


Fig.2 Logic symbol.

Contains two 2x4 decoder with separate enable

Decoders (cont.)

- 74x139 Dual 2x4 decoder (cont.)

FUNCTION TABLE

INPUTS			OUTPUTS			
$n\overline{E}$	nA_0	nA_1	$n\overline{Y}_0$	$n\overline{Y}_1$	$n\overline{Y}_2$	$n\overline{Y}_3$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

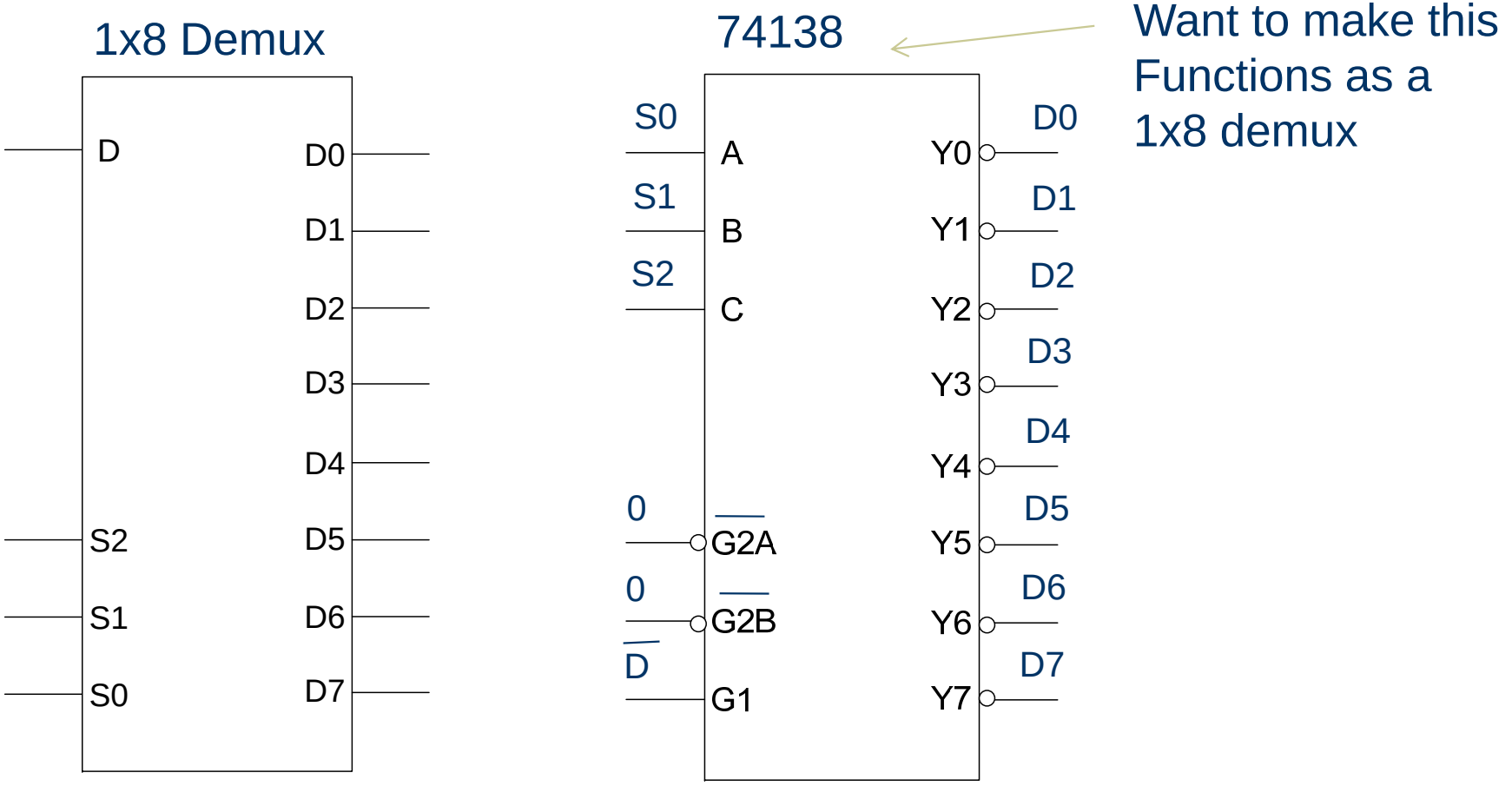
Notes

1. H = HIGH voltage level
   L = LOW voltage level
   X = don't care

Output depends on inputs A with $n\overline{E} = 0$

Decoders (cont.)

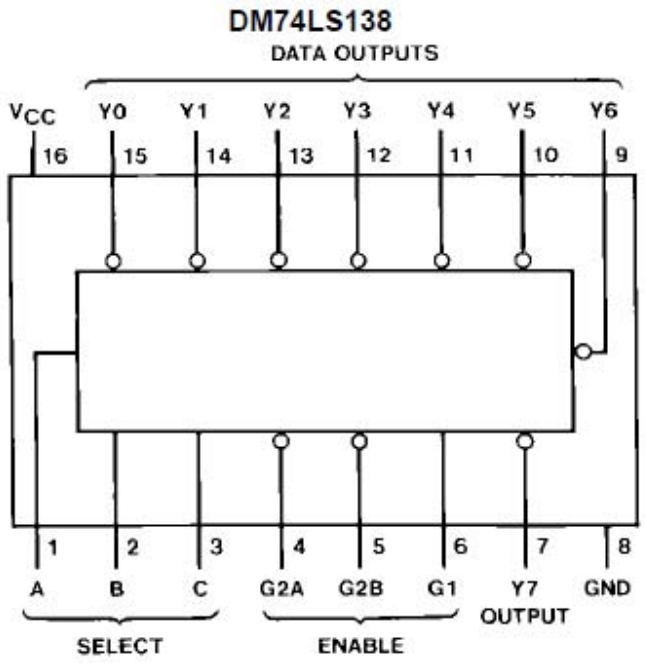
- How to design 1x8 demux using the 74138?

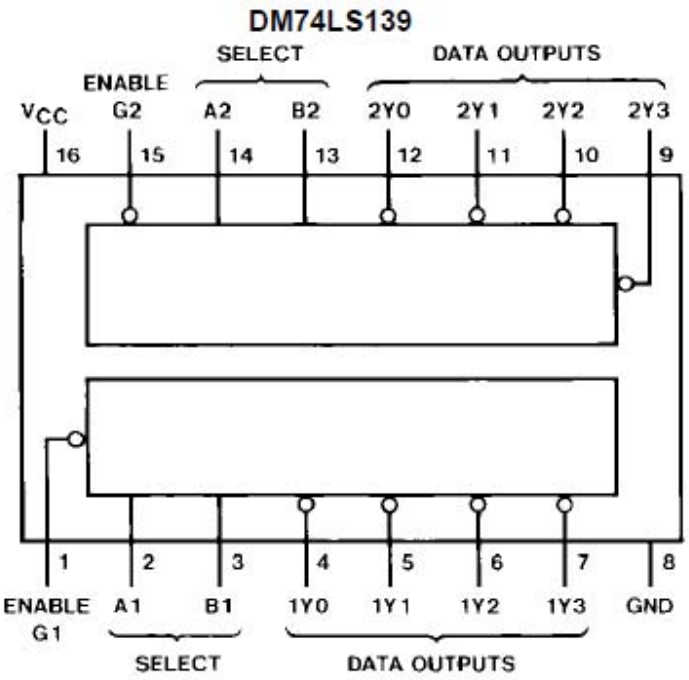


Decoders (In General)

DM74LS138 • DM74LS139

Connection Diagrams





Decoders (LS138 & LS139)

Function Tables

DM74LS138

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	YO	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

DM74LS139

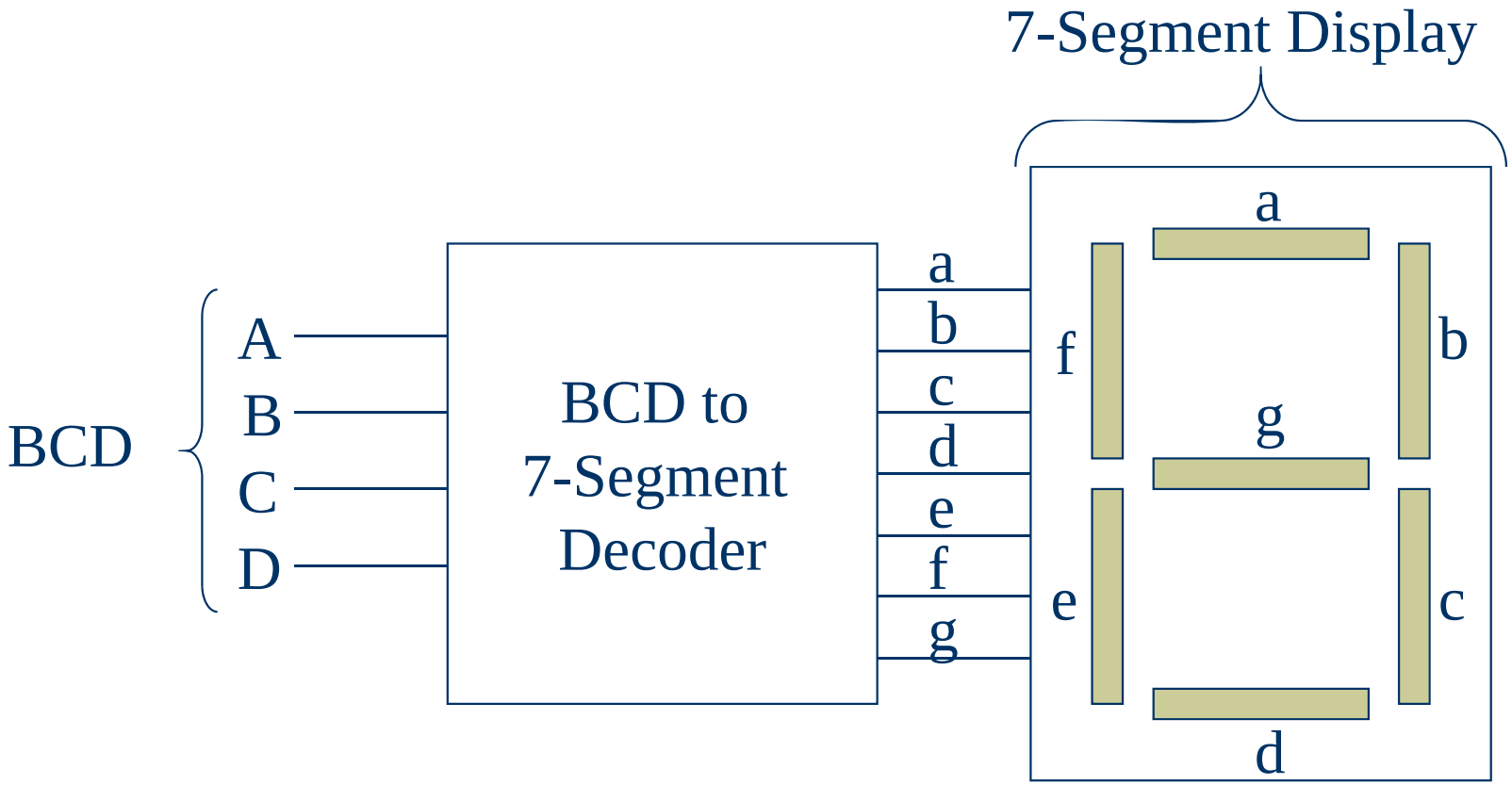
Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Level
L = LOW Level
X = Don't Care

Note 1: G2 = G2A + G2B

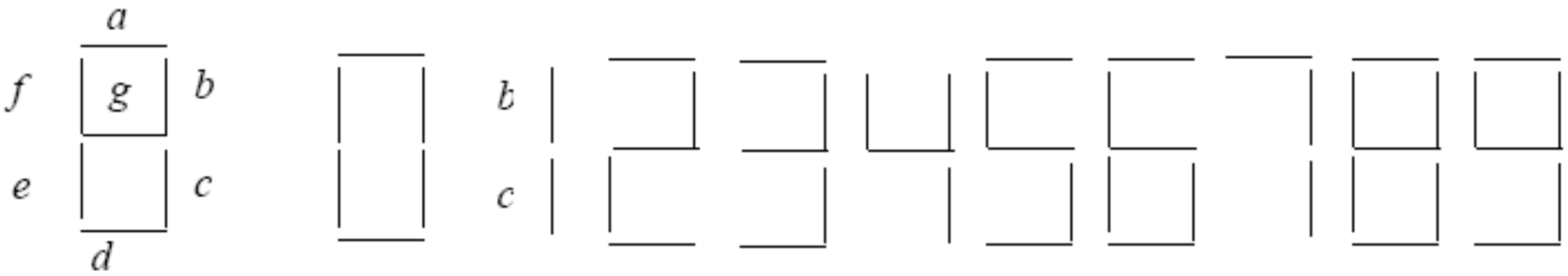
Decoder

- BCD to 7-Segment Decoder



Decoder

- 7-Segment Display



- Common Anode :
  - Output of decoder must be active low to light up.
- Common Cathode :
  - Output of decoder must be active high to light up.

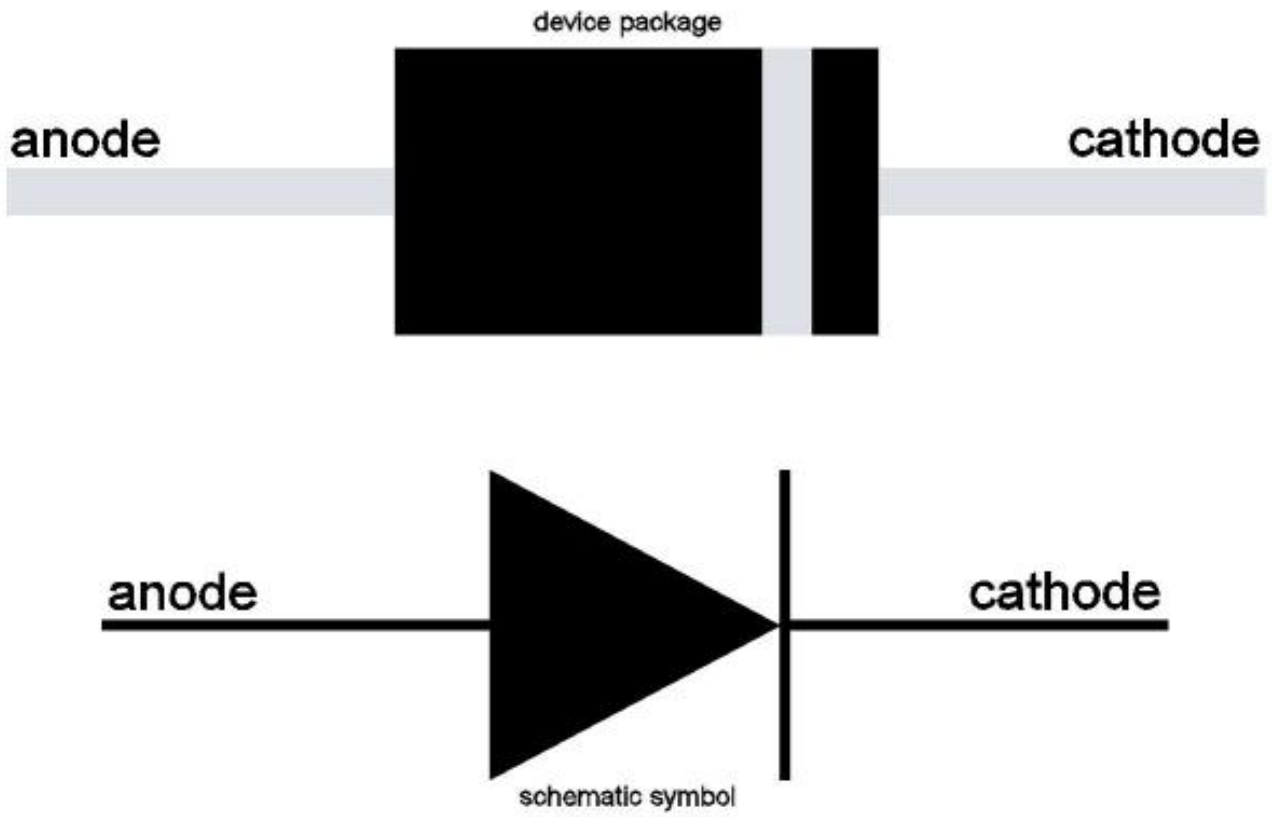
BCD to seven segments decoders-drivers 7447A (74LS47) and 7448 (74LS48)

- The BCD to 7 Segments Decoder is used exclusively for the drive the display of decimal digits; therefore it is usually called Decoder Driver.

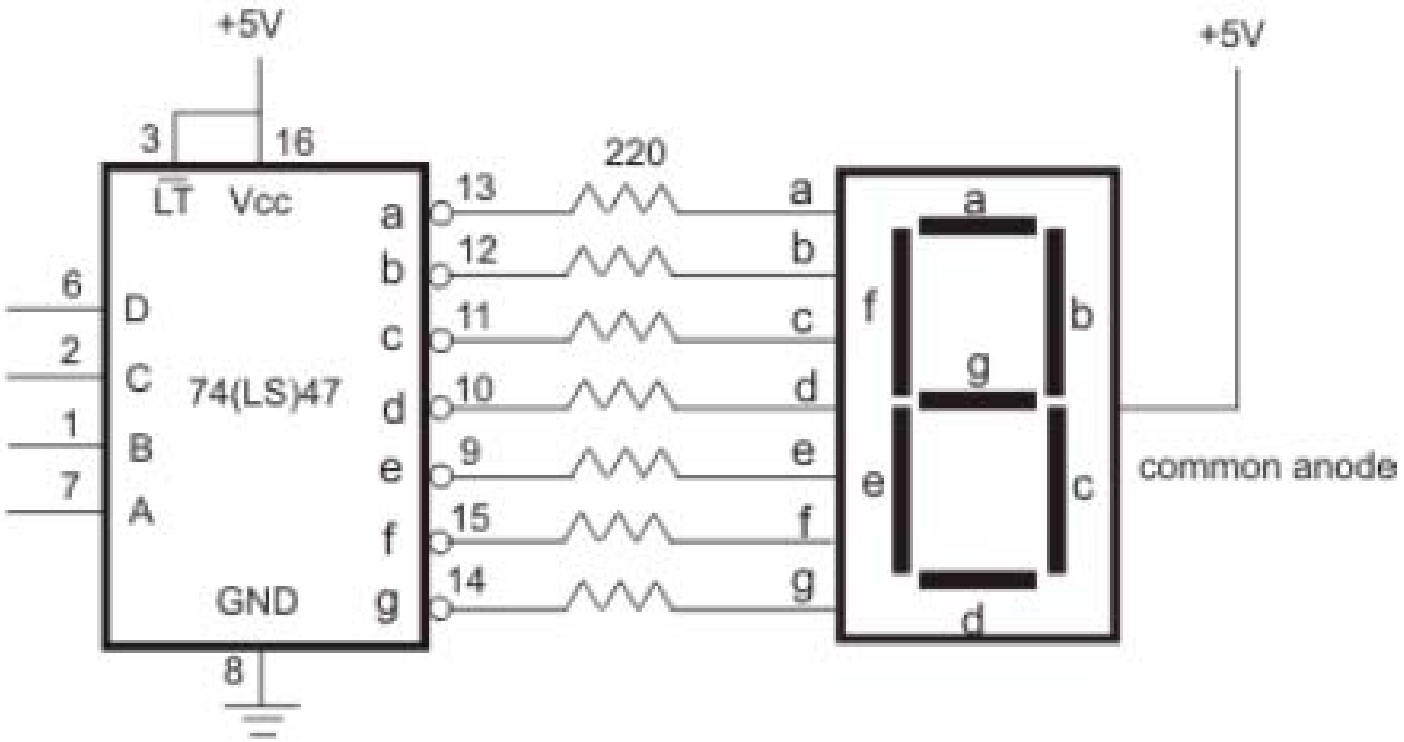
BCD to seven segments decoders-drivers 7447A (74LS47) and 7448 (74LS48)

- BCD to 7 Segments Decoders are the TTL the 74(LS)47 which drives the common anode displays and the 74(LS)48 which drives the common cathode displays.

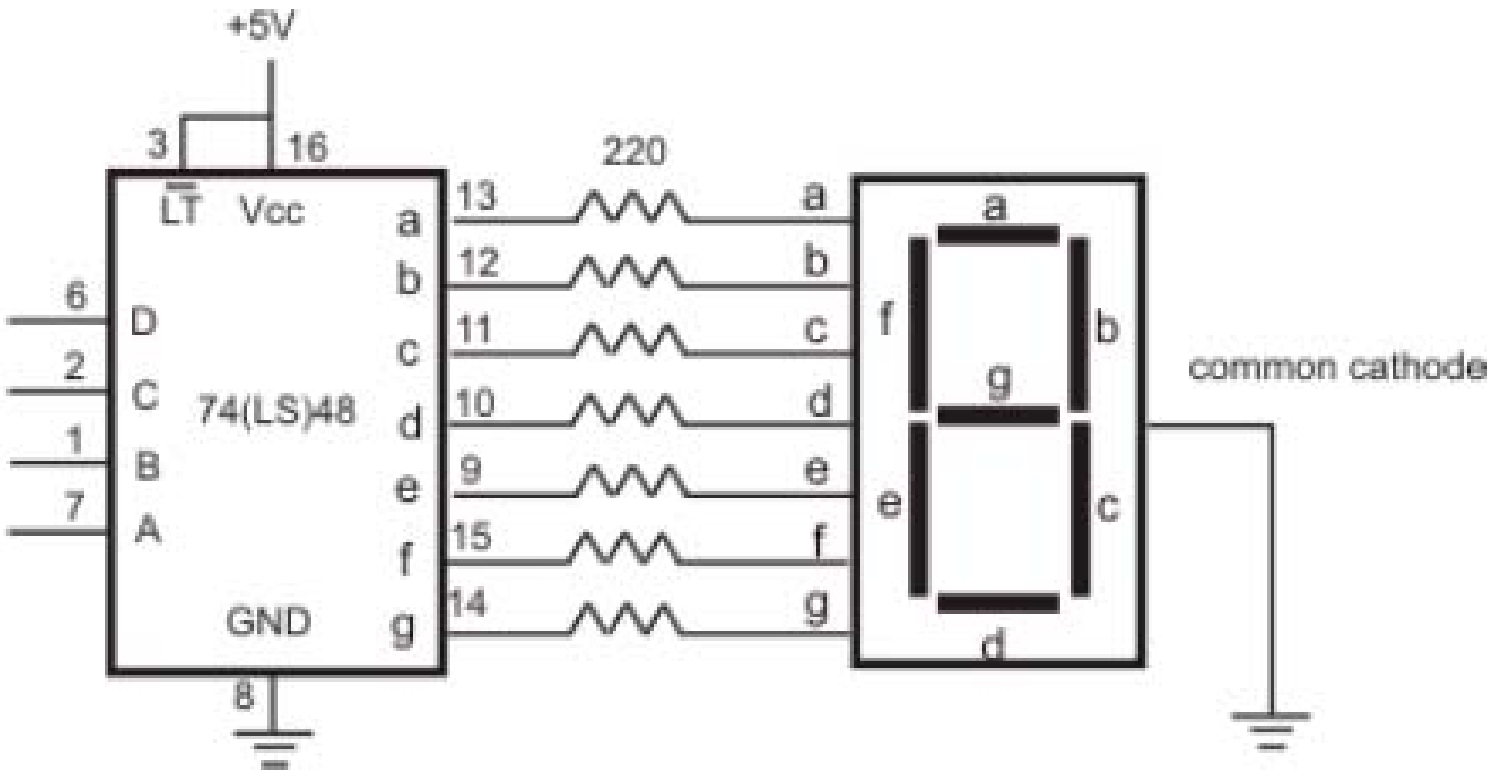
Anode and Cathode



BCD to seven segments decoders-drivers 7447A (74LS47)



BCD to seven segments decoders-drivers 7448 (74LS48)

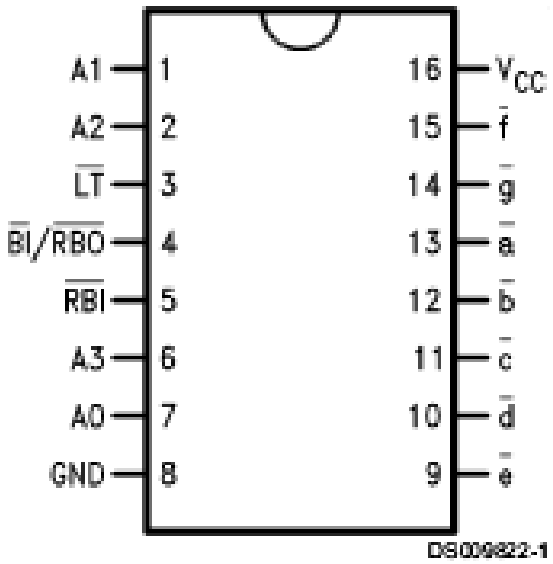


BCD to 7 Segment Decoder

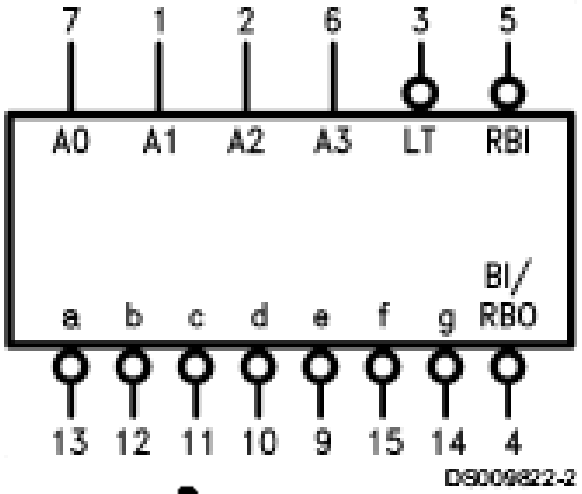
- 7447 BCD to 7 Segment Decoder

Connection Diagram

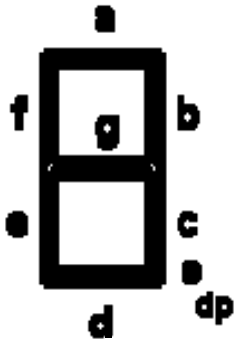
Dual-In-Line Package



Logic Symbol



V_{CC} = Pin 16
GND = Pin 8

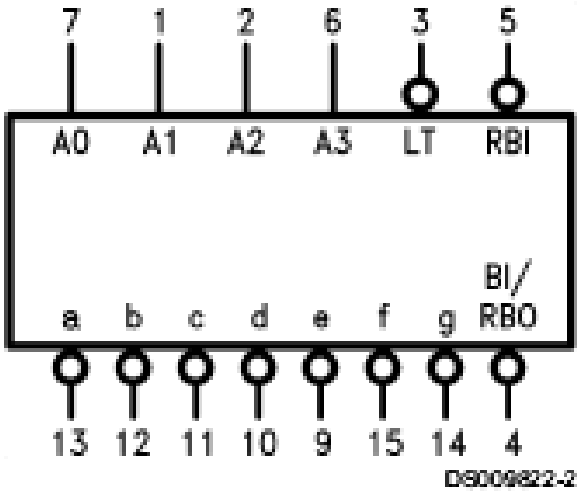


Converts 4-bit BCD (A3, A2, A1, A0) to 7-segment LED (a,b,c,d,e,f,g)

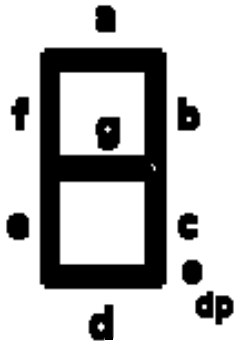
BCD to 7 Segment Decoder (cont.)

- What is the output of 7447 if input is 0110?

Logic Symbol



V_{CC} = Pin 16
GND = Pin 8



a b c d e f g
0 1 0 0 0 0 0

Exercise 1

- Show how the outputs of the truth table below can be implemented using:
  - 3 to 8 (1-of-8) active high decoder
  - 2 to 4 (1-of-4) active high decoder

E’	A	B	C	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1

Solution Exercise 1

- 3 to 8 (1-of-8) active high decoder
  - Answer : label a 3 to 8 decoder correctly
- 2 to 4 (1-of-4) active high decoder
  - Answer : Cascade two 2-to-4 decoders, assume both decoders have active low enable bit E' (E-bar). The first decoder's enable bit connected to A and the second decoder's enable bit connected to A-bar. The two inputs of each decoder is connected to B and C, with B as MSB. Eight outputs can be obtained, with output O0-O3 from first decoder, and O4-O7 from second decoder.

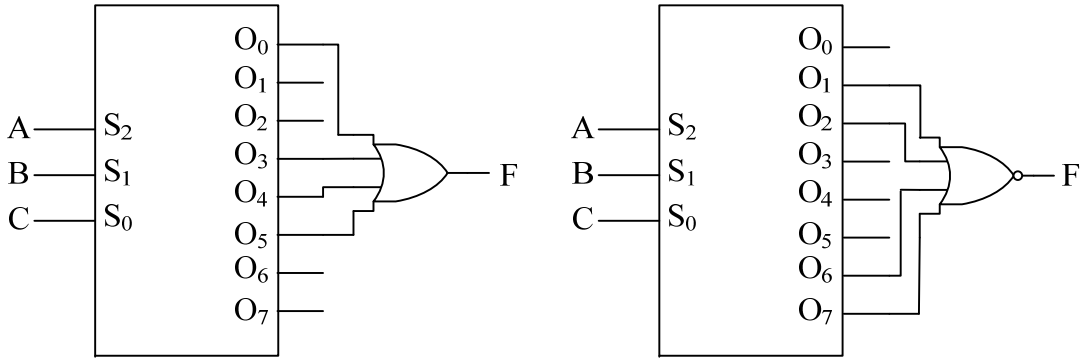
Exercise 2

- Implement the output of the truth table using a 3 to 8 (or 1-of-8) decoder with :
  - active high output
  - active low output

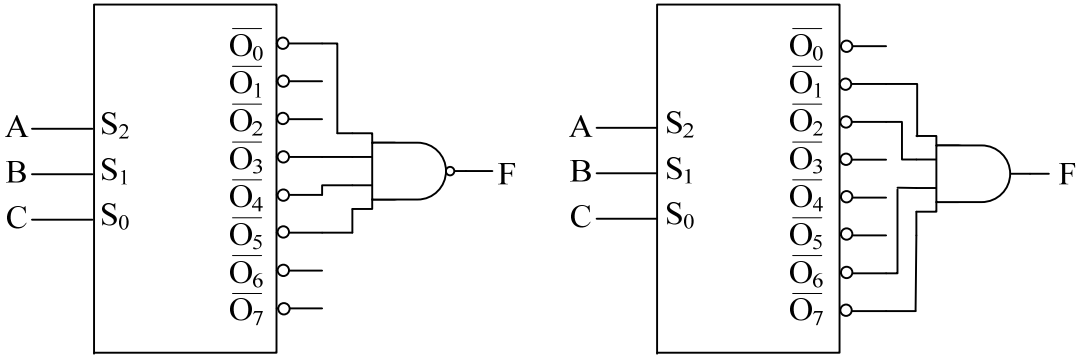
A	B	C	F	
0	0	0	1	O_0 or $\overline{O_0}$
0	0	1	0	O_1 or $\overline{O_1}$
0	1	0	0	O_2 or $\overline{O_2}$
0	1	1	1	O_3 or $\overline{O_3}$
1	0	0	1	O_4 or $\overline{O_4}$
1	0	1	1	O_5 or $\overline{O_5}$
1	1	0	0	O_6 or $\overline{O_6}$
1	1	1	0	O_7 or $\overline{O_7}$

Solution Exercise 2

- Implement the output of the truth table using a 3 to 8 (or 1-of-8) decoder with :
  - active high output

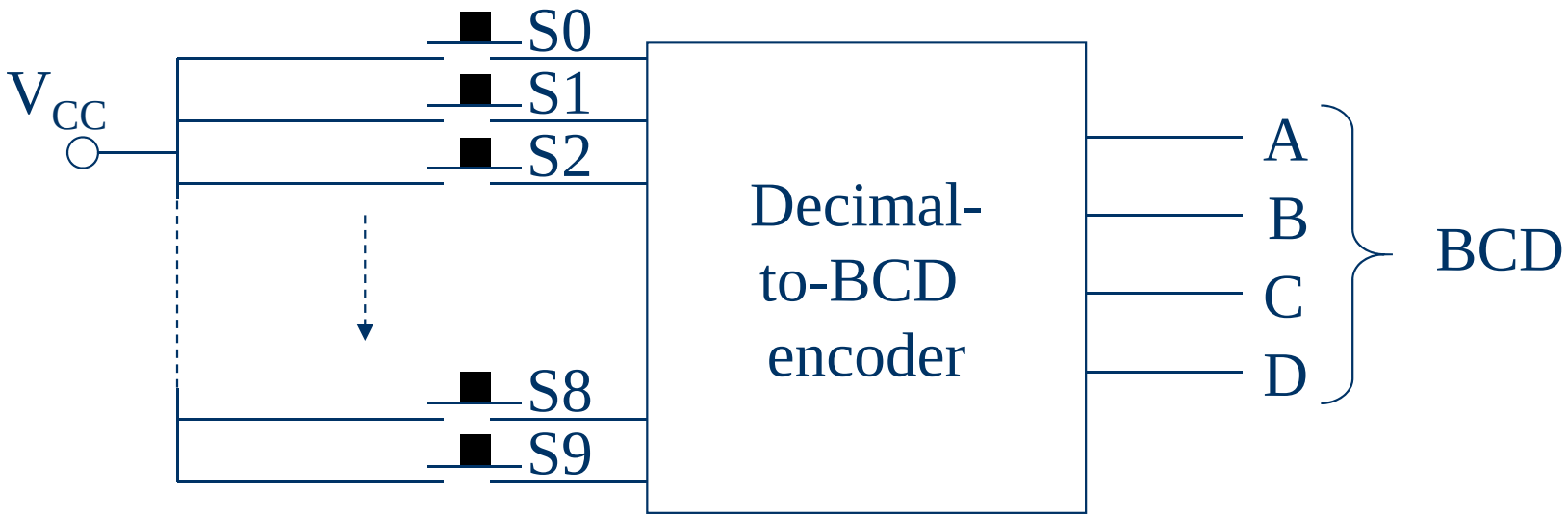


  - active low output



Encoder

- An encoder accepts an active level on ONE of its inputs, and converts it to a coded output.
- For example, the Decimal to BCD encoder as shown in figure below. This encoder has 10 inputs and 4-bit output. Input and output are ACTIVE HIGH (normal state is LOW).



Encoder

- Truth table for Decimal-to-BCD encoder

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	A	B	C	D
1										0	0	0	0
	1									0	0	0	1
		1								0	0	1	0
			1							0	0	1	1
				1						0	1	0	0
					1					0	1	0	1
						1				0	1	1	0
							1			0	1	1	1
								1		1	0	0	0
									1	1	0	0	1

Encoder

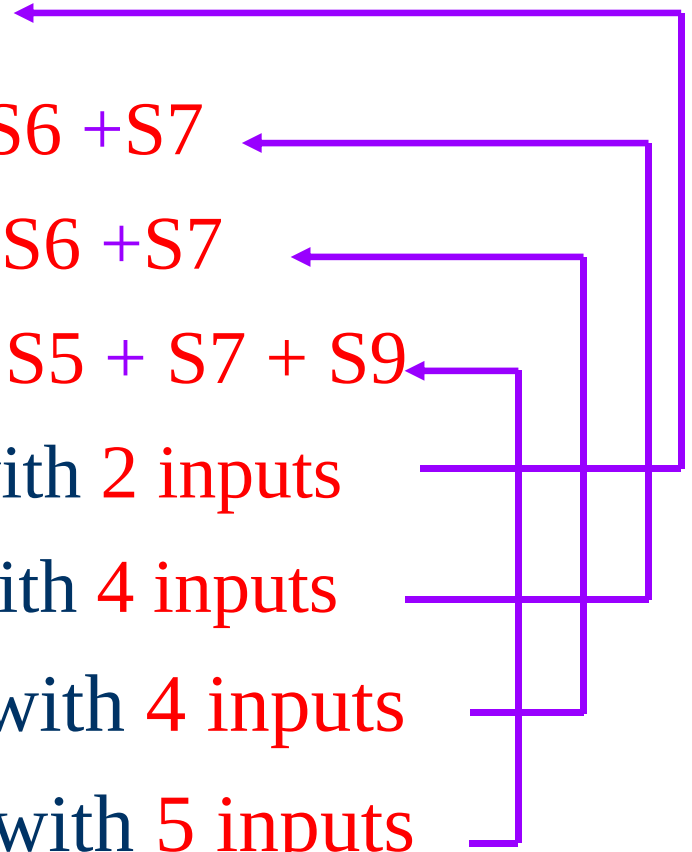
- Obtain the Boolean expression for each output, and draw the logic circuit

A = S8 +S9

B = S4 +S5 + S6 +S7

C = S2 + S3 + S6 +S7

D = S1 + S3 + S5 + S7 + S9



A has one OR gate with 2 inputs

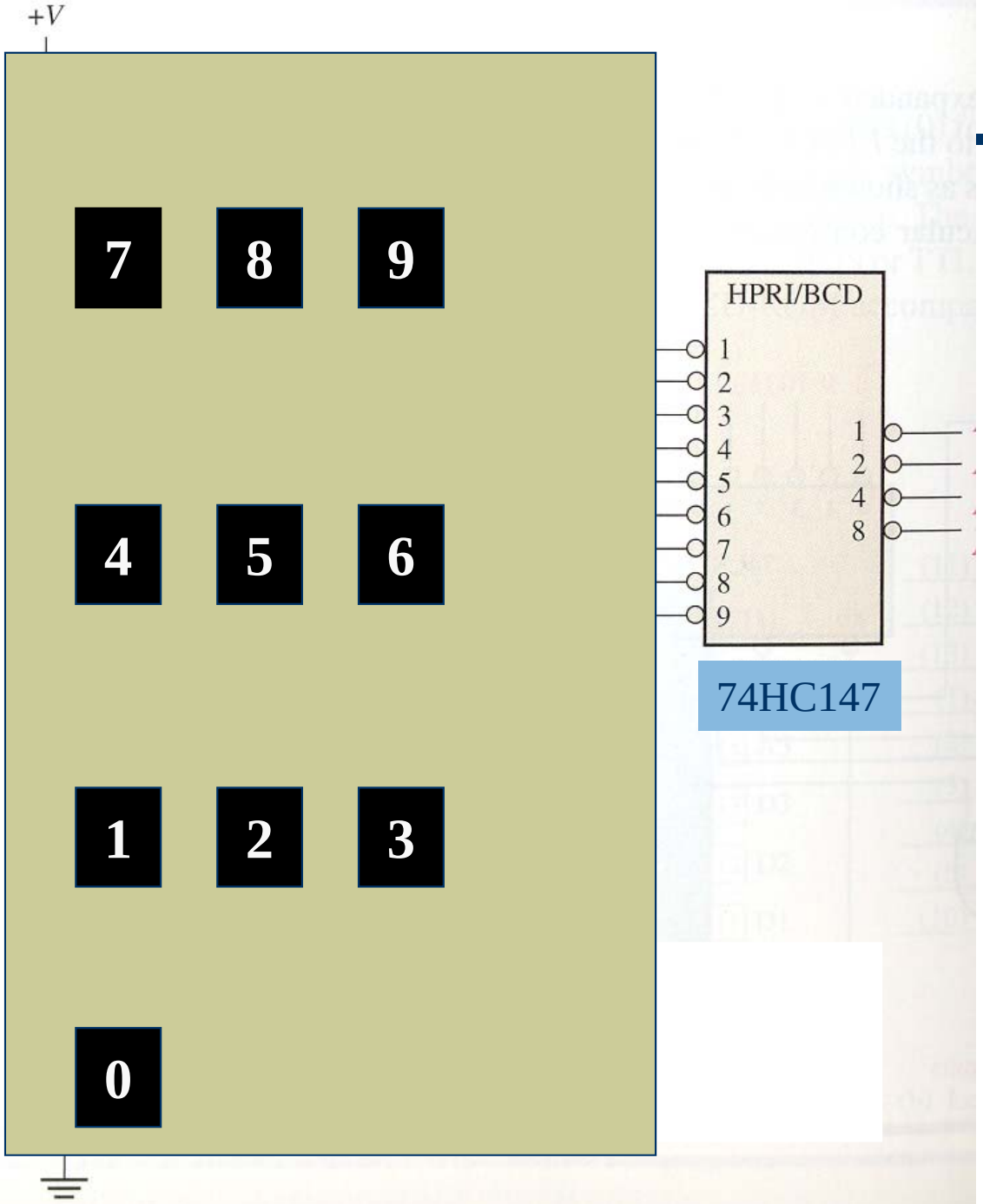
B has one OR gate with 4 inputs

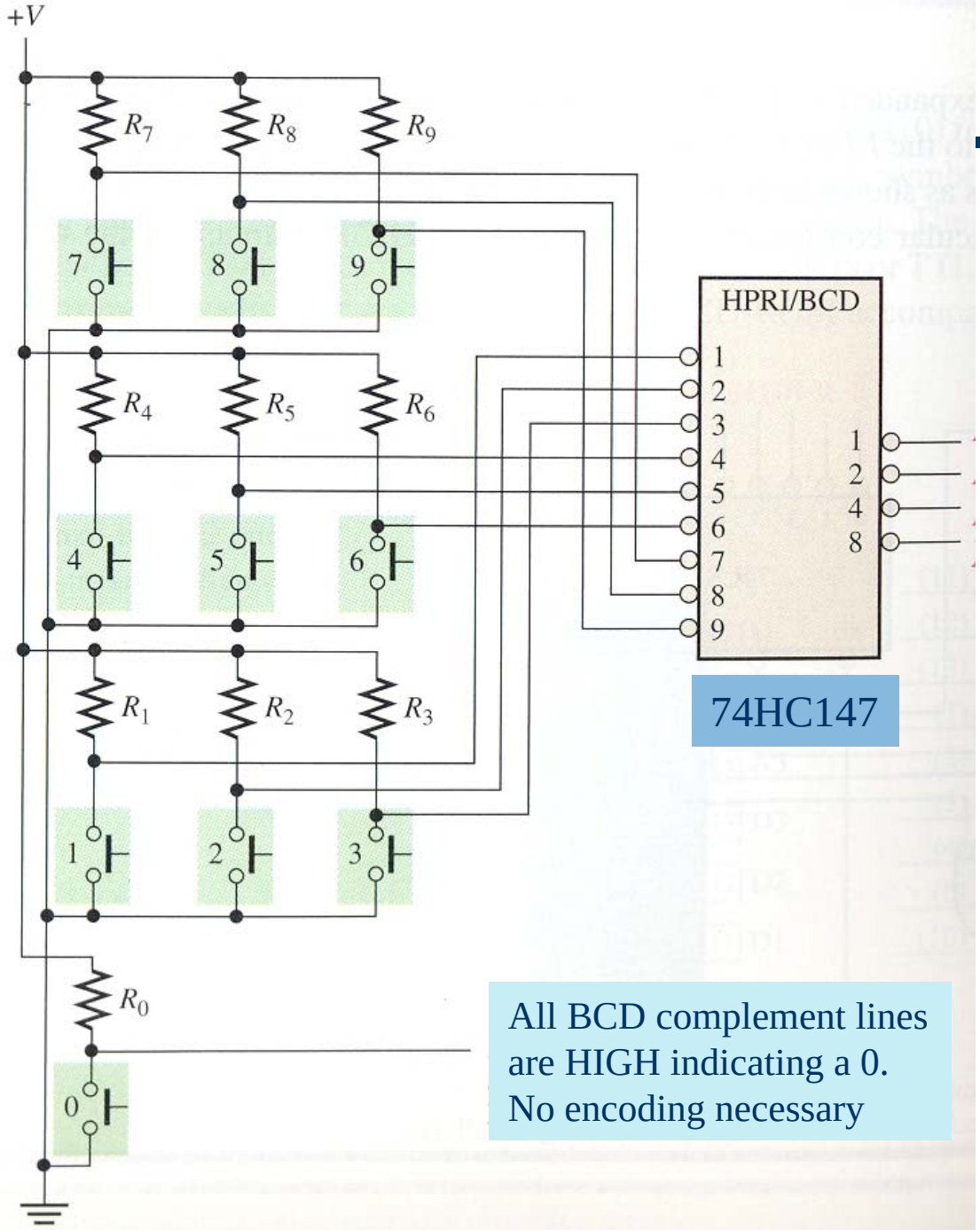
C has one OR gate with 4 inputs

D has one OR gate with 5 inputs

Encoder

- The 74HC147 Decimal-to-BCD **Priority Encoder**.
- Active-LOW input, active-LOW output
- HPRI /BCD means highest value input has priority.
- Example: If both key 3 and 6 are pressed, the BCD output will be 0110 (representing the decimal 6, which has higher value).


+V
7
8
9
4
5
6
1
2
3
0
HPRI/BCD
1
2
3
4
5
6
7
8
9
1
2
4
8
74HC147



74HC147

All BCD complement lines are HIGH indicating a 0. No encoding necessary

Encoder

- Design a 3 inputs and 2 outputs LOWEST-PRIORITY ENCODER? (**active HIGH** input and output)
- Ex: if all S_1, S_2, and S_3 is activated then S_1 will be considered.
- If S_1 =1, Y_1Y_0=01; if S_2 =1, Y_1Y_0=10; if S_3 =1, Y_1Y_0=11; no inp active, Y_1Y_0=00

$S_3S_2S_1$	Y_1Y_0
000	00
001	01
010	10
011	01
100	11
101	01
110	10
111	01

Encoder

- The truth table for
  - **lowest priority** encoder
  - active high input
  - active high output

  can be simplified

S_3	S_2	S_1	Y_1 Y_0
0	0	0	00
X	X	1	01
X	1	0	10
1	0	0	11

Encoder

- Design a 3 inputs and 2 outputs HIGHEST-PRIORITY ENCODER? (active LOW input and output)
- Eg, if all A, B, and C is activated then A will be considered.
- If A=0, Y_1Y_0=00; if B=0, Y_1Y_0=01; if C=0, Y_1Y_0=10; no inp active, Y_1Y_0=11

ABC	Y_1Y_0
000	00
001	00
010	00
011	00
100	01
101	01
110	10
111	11

Encoder

- The truth table for
  - highest priority encoder
  - **active low** input
  - active low output

  can be simplified to:

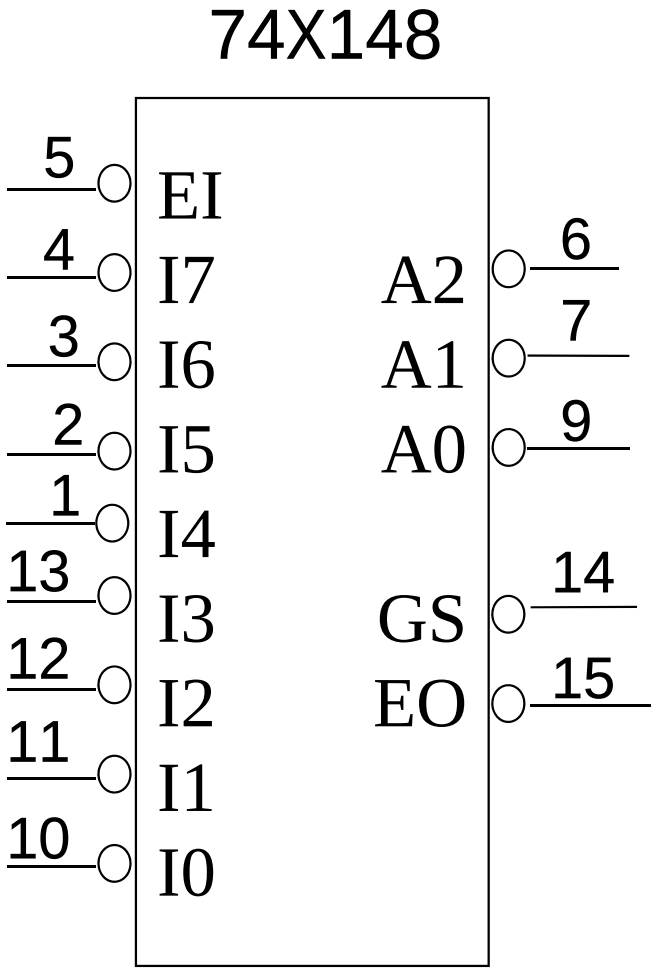
$S_3S_2S_1$	Y_1Y_0
0 X X	00
1 0 X	01
1 1 0	10
1 1 1	11

5.5 Encoders

- Priority Encoders

The solution is to assign priority to the input lines, so that when multiple inputs are asserted, the encoding device produces the number of the highest-priority input. Such a device is called a priority encoder.

- The 74x148 Priority Encoders



5.5 Encoders

- Input data $\overline{I0}$-$\overline{I7}$
- Enable input $\overline{EI}$
- Output $\overline{GS}$ is asserted when the device is enabled and one or more of the request are asserted.
- Output $\overline{EO}$ is asserted if $\overline{EI}$ is assert but no request input is asserted, it designed to be connected to the $\overline{EI}$ input of another ‘148 that handles lower-priority requests.

5.5 Encoders

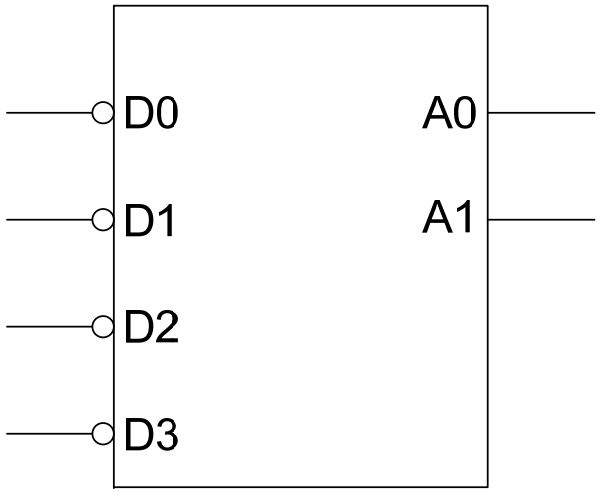
- Truth table for a 74x148 8-input priority encoder

Inputs									Outputs				
$\overline{EI}$	$\overline{I0}$	$\overline{I1}$	$\overline{I2}$	$\overline{I3}$	$\overline{I4}$	$\overline{I5}$	$\overline{I6}$	$\overline{I7}$	$\overline{A2}$	$\overline{A1}$	$\overline{A0}$	$\overline{GS}$	$\overline{EO}$
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

Encoders

- Performs reverse decoder function

4x2 Encoder



Function:

$A_1A_0 = 00$ when $D_3D_2D_1D_0 = 1110$

$A_1A_0 = 01$ when $D_3D_2D_1D_0 = 1101$

$A_1A_0 = 10$ when $D_3D_2D_1D_0 = 1011$

$A_1A_0 = 11$ when $D_3D_2D_1D_0 = 0111$

Which implies,

$$A_1 = D_3\overline{D_2}D_1D_0 + \overline{D_3}D_2D_1D_0$$

$$A_0 = D_3D_2\overline{D_1}D_0 + \overline{D_3}D_2D_1D_0$$

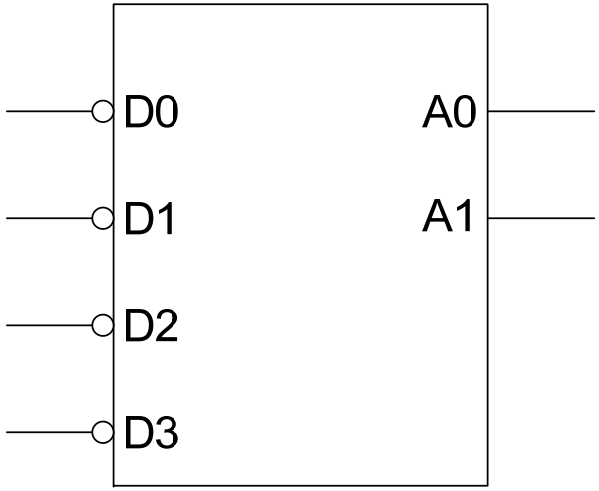
What happens if more than 1 input is '0' (D0 = 0 and D1 = 0)?

- We need a priority encoder

Encoders (cont.)

- Priority Encoder: Output depends on the largest active input

4x2 Priority Encoder



Function:

$A_1A_0 = 00$ when $D_3D_2D_1D_0 = 1110$

$A_1A_0 = 01$ when $D_3D_2D_1D_0 = 110x$

$A_1A_0 = 10$ when $D_3D_2D_1D_0 = 10xx$

$A_1A_0 = 11$ when $D_3D_2D_1D_0 = 0xxx$

Can you derive the truth table of the priority encoder?

What happens if more than 1 input is '0' (D0 = 0 and D1 = 0)?

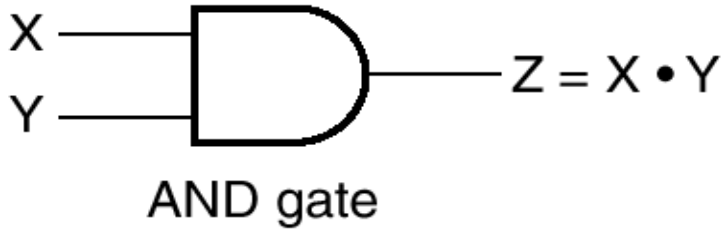
- output A1A0 = 01

Logic function using Mux

- Any logic function (AND, OR, decoder, encoder, etc) can be realized using mux. All we need is a truth table
- This is the concept widely being used today in FPGA (Field Programmable Gate Array), where any logic functions can be rapidly implemented using Mux

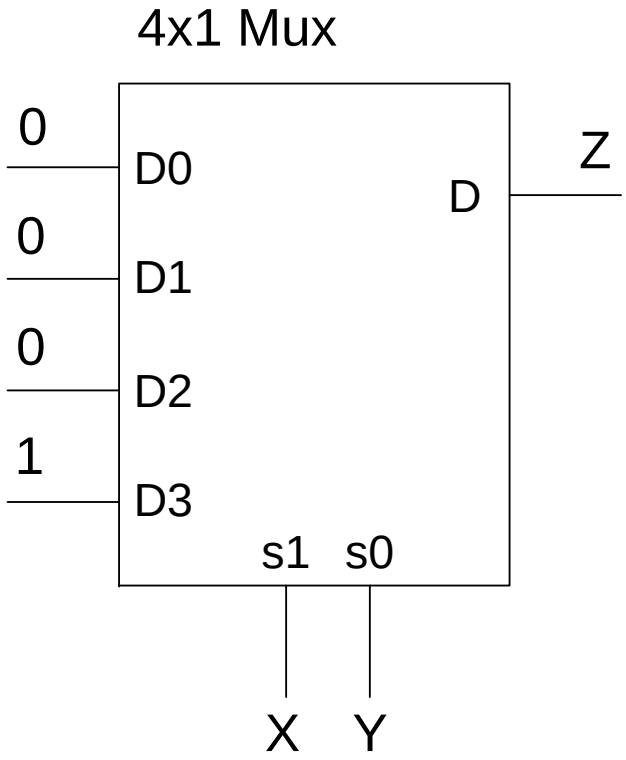
Logic function using Mux (cont.)

- 2-input AND gate using 4x1 Mux



AND gate

X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

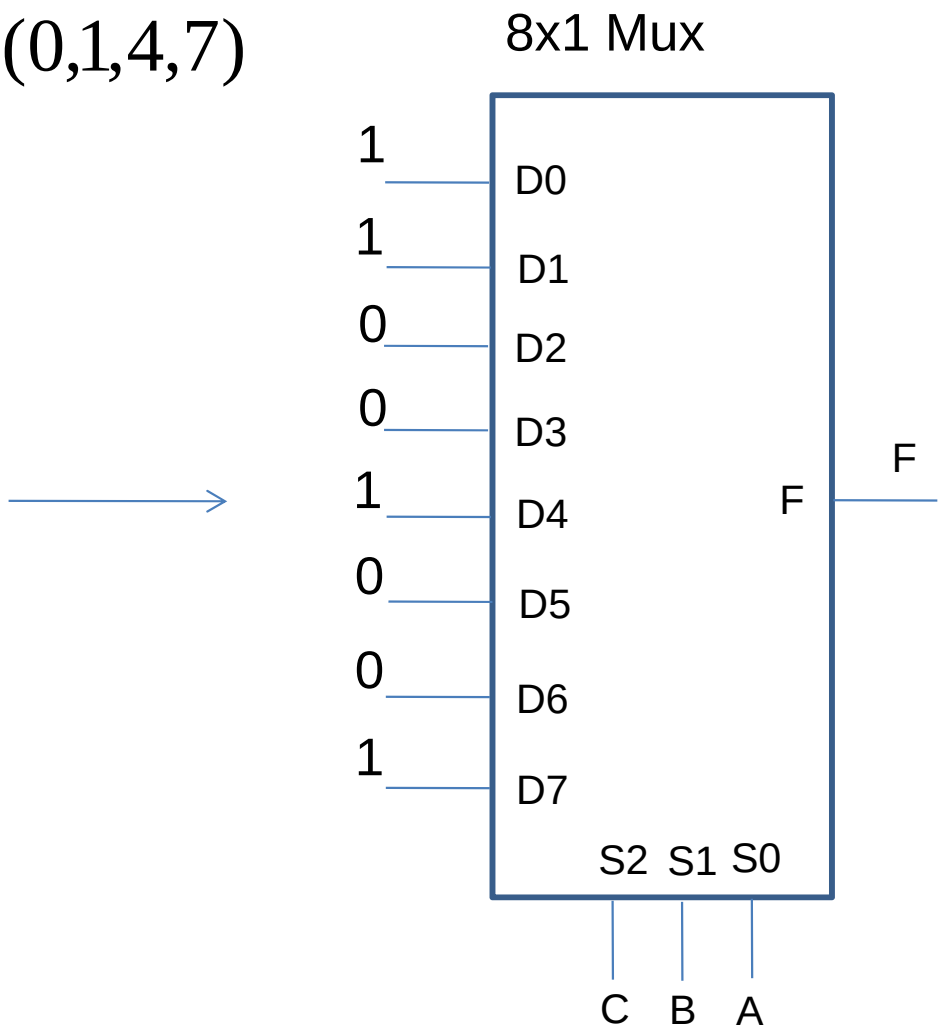


Logic function using Mux (cont.)

- Implement the following using Mux

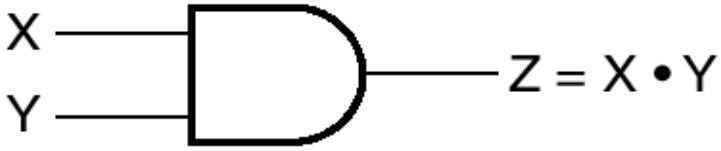
$$F(A,B,C) = \sum m(0,1,4,7)$$

C B A	F
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1



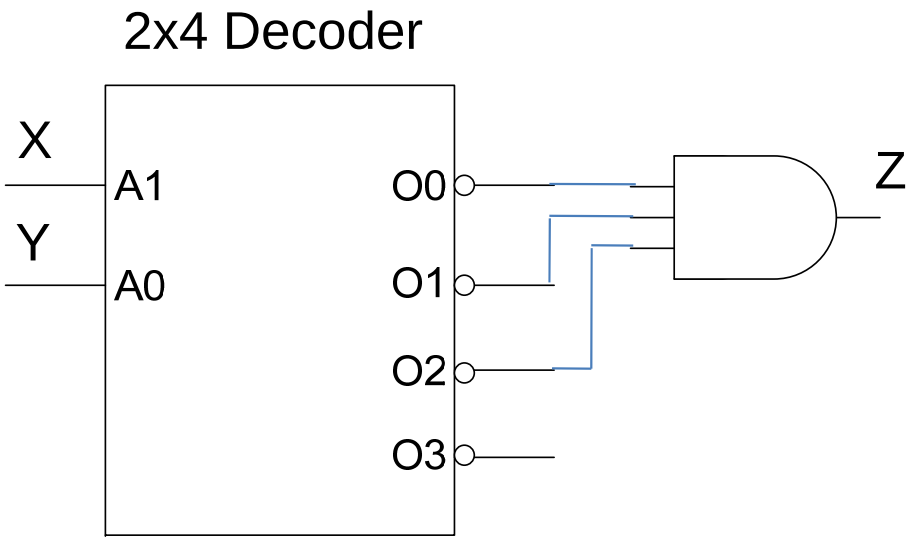
Logic function using Decoder

- Similar to Mux, any logic function can be realized using decoders
- 2-input AND gate using 2x4 decoder



AND gate

X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1



Logic function using Decoder (cont.)

- Implement the following using 74138

$$F(A,B,C) = \prod M(2,3,5,6)$$

C B A	F
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1