

Department of
Electronics, Computer, and Information
Technology

ECT 213 Digital Electronics

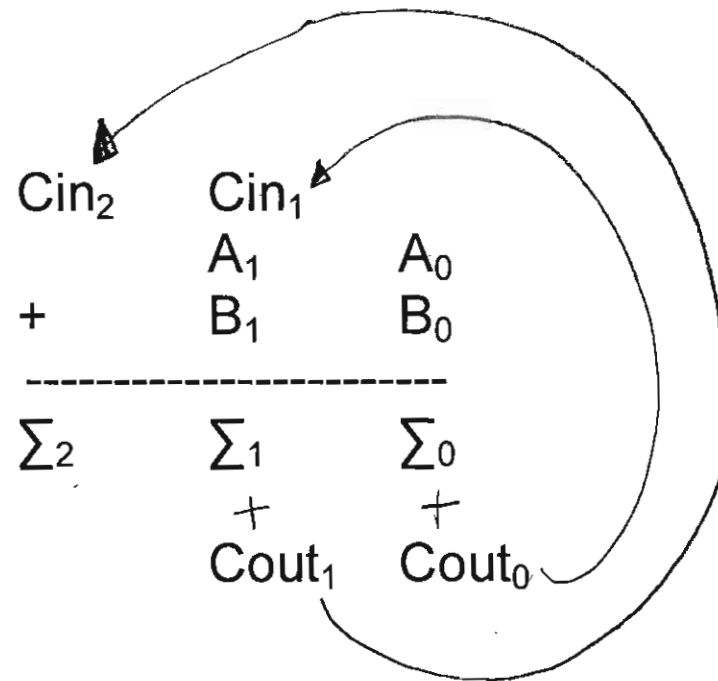
Lecture 10: Chapter 7
Adder Circuits
Chapter 8
Comparator
Decoder
Encoder
Multiplexer
DeMultiplexer

Chapter 7

Adder Circuits:

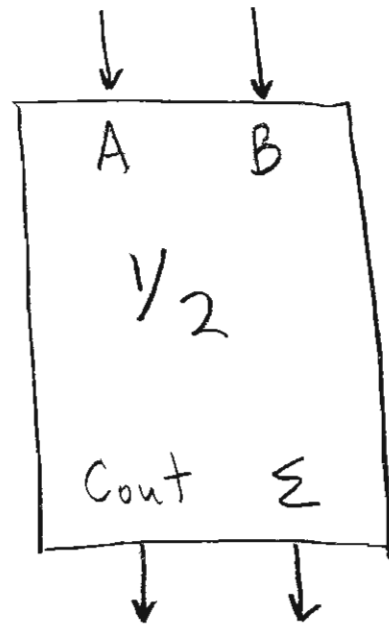
A_0	B_0	Σ_0	C_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A_1	B_1	C_{in}	Σ_1	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



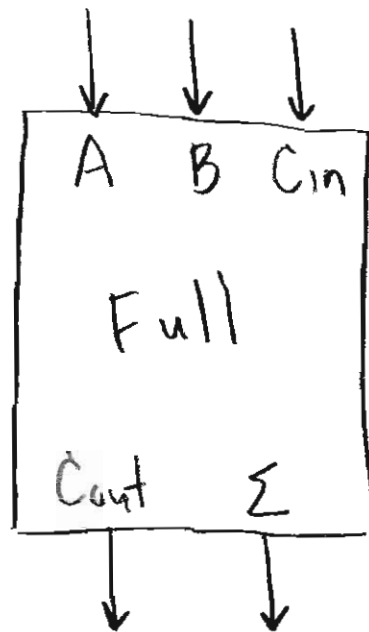
Half-adder

Block Diagram:

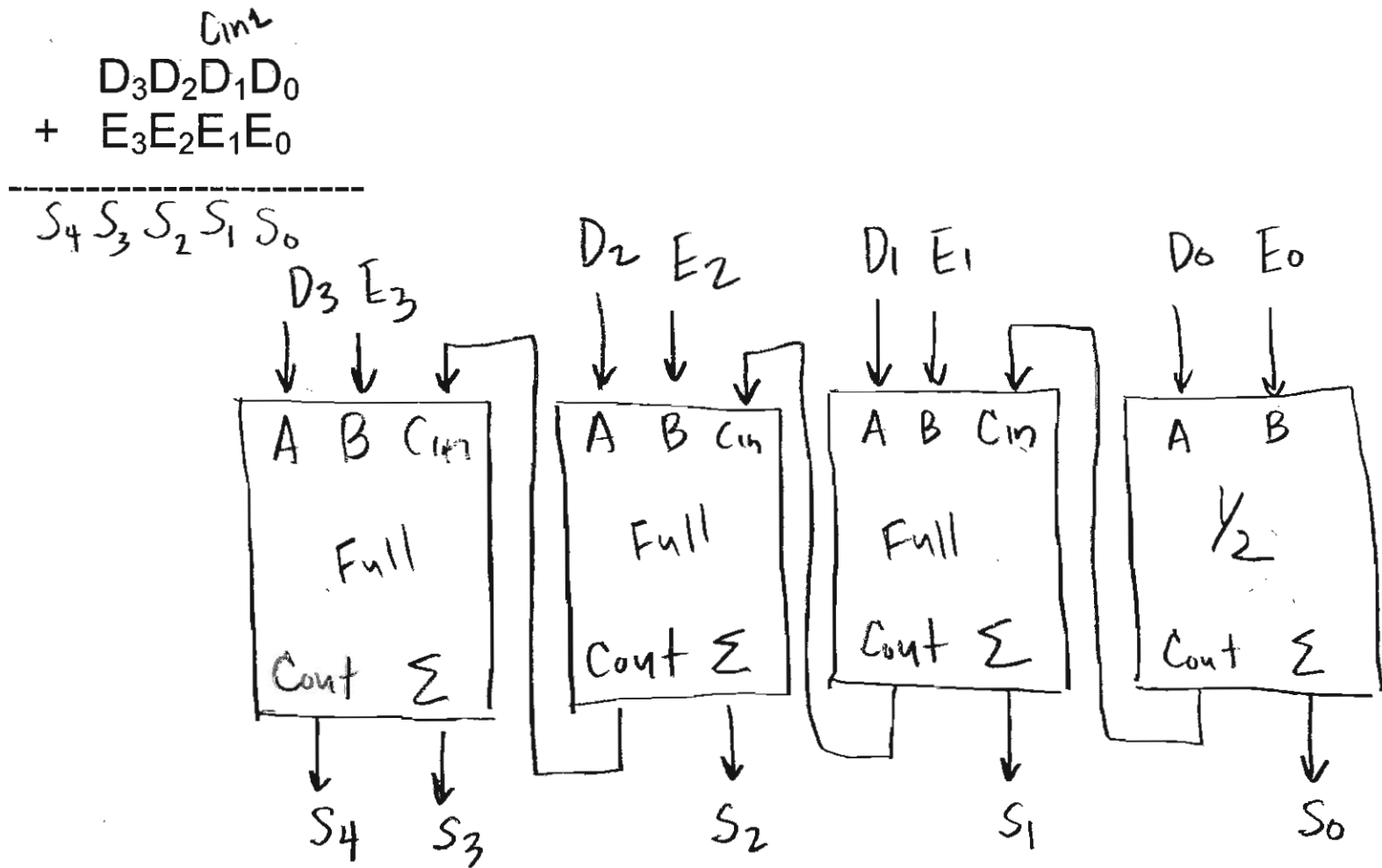


Full Adder:

Block Diagram

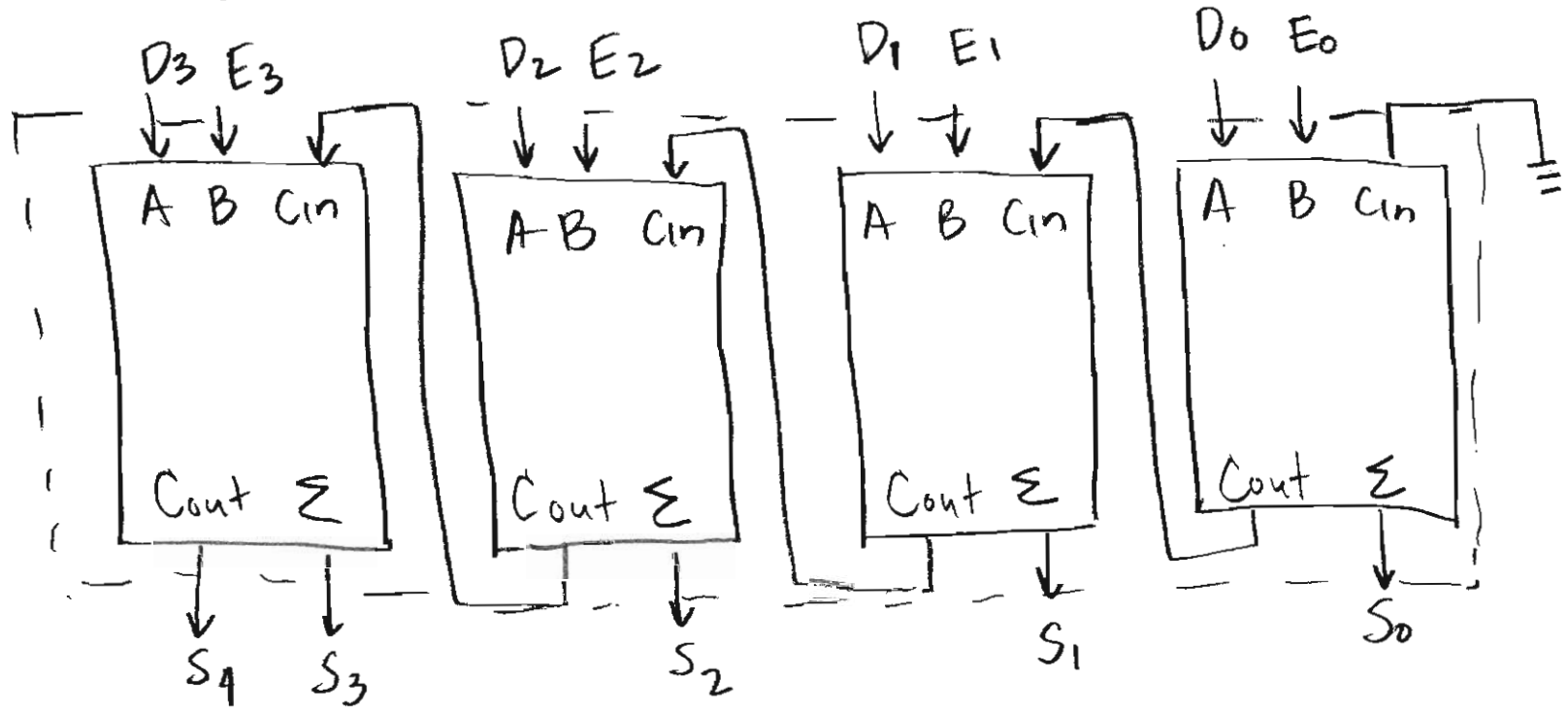


Example 1: Draw the block diagram of the following Arithmetic expression using half and full adders

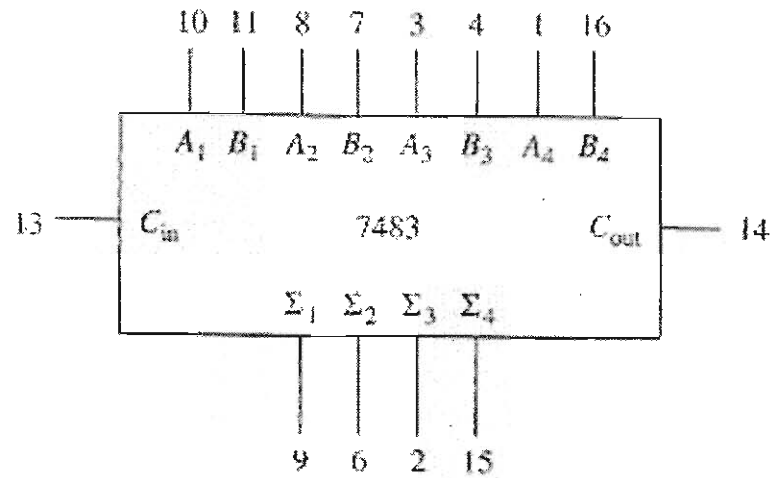


Example 2: Draw the block diagram of the following Arithmetic expression using only full-adders

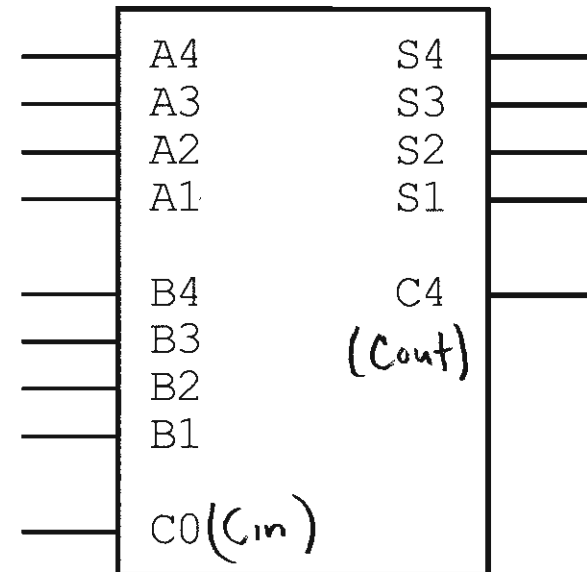
$$\begin{array}{r}
 D_3 D_2 D_1 D_0 \\
 + E_3 E_2 E_1 E_0 \\
 \hline
 S_4 S_3 S_2 S_1 S_0
 \end{array}$$



Logic Diagram of 7483 4-bit adder full adder



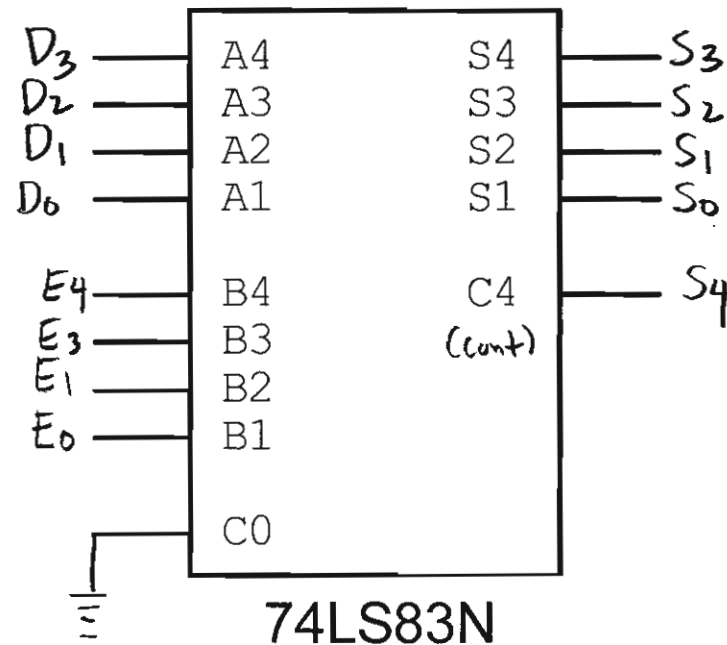
V_{CC} = Pin 5
 \bar{GND} = Pin 12



74LS83N

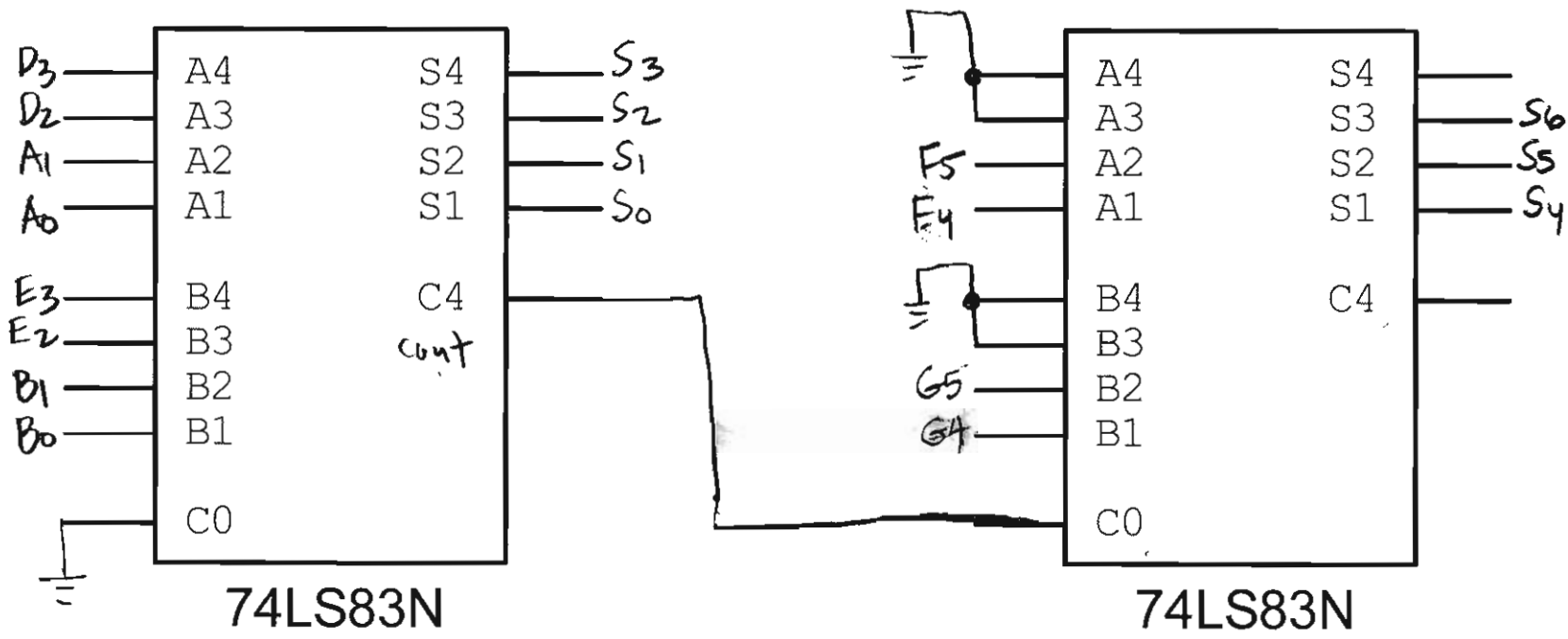
Example 3: Draw the block diagram of the following Arithmetic expression using the 7483 IC.

$$\begin{array}{r}
 D_3D_2D_1D_0 \\
 + E_3E_2E_1E_0 \\
 \hline
 S_4S_3S_2S_1S_0
 \end{array}$$



Example 4: Draw the block diagram of the following Arithmetic expression using only 74LS83 ICs

$$\begin{array}{r}
 F_3 F_2 D_3 D_2 A_1 A_0 \\
 + G_3 G_2 E_3 E_2 B_1 B_0 \\
 \hline
 S_6 S_5 S_4 S_3 S_2 S_1 S_0
 \end{array}$$



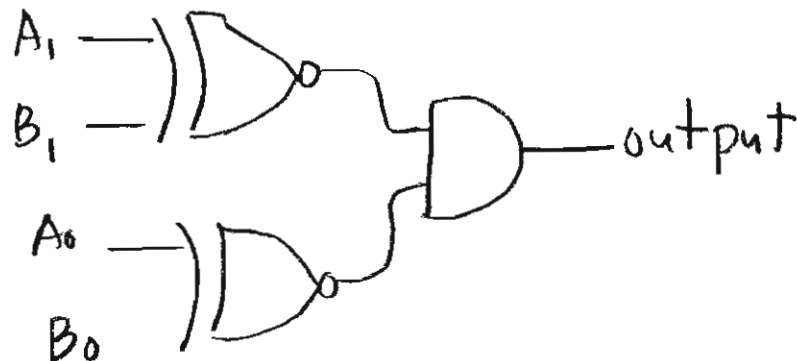
Chapter 8

Comparator:

- Compares two binary strings.
- Outputs 1 if both strings equal
- Outputs 0 if both strings do not equal

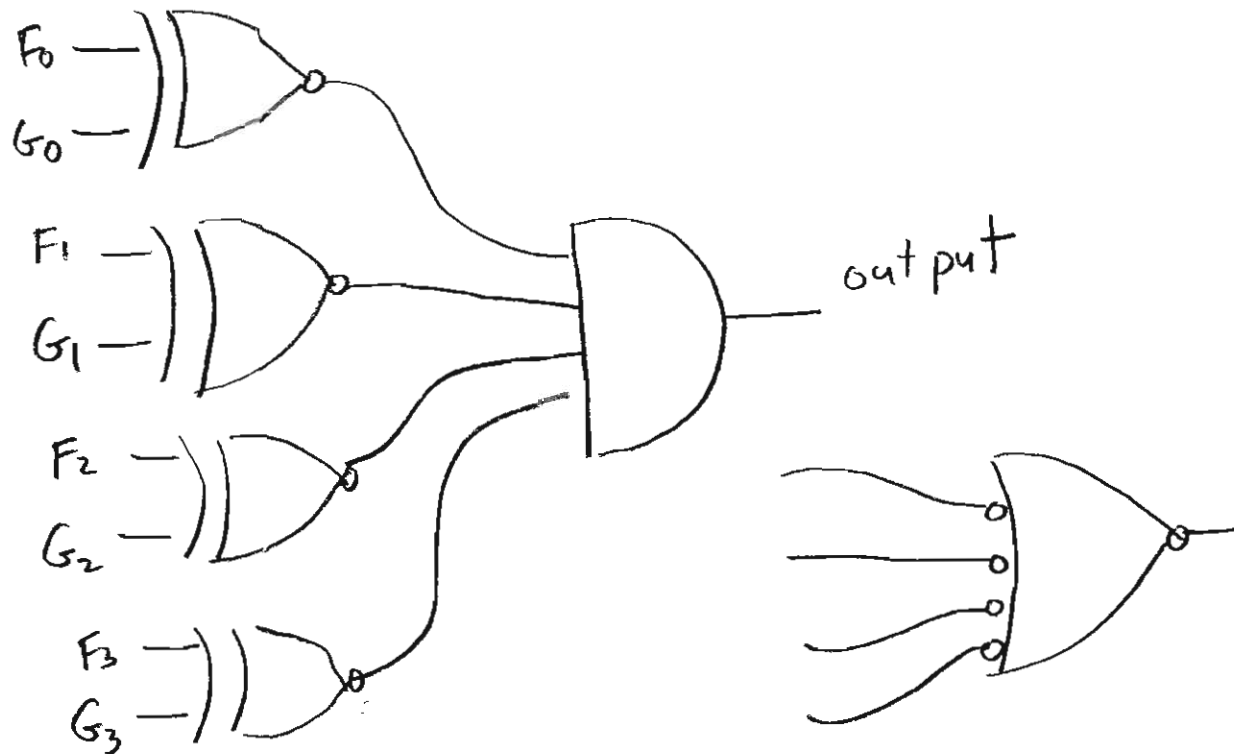
$$A = A_1A_0 \text{ vs } B_1B_0 = B$$

Basic comparator



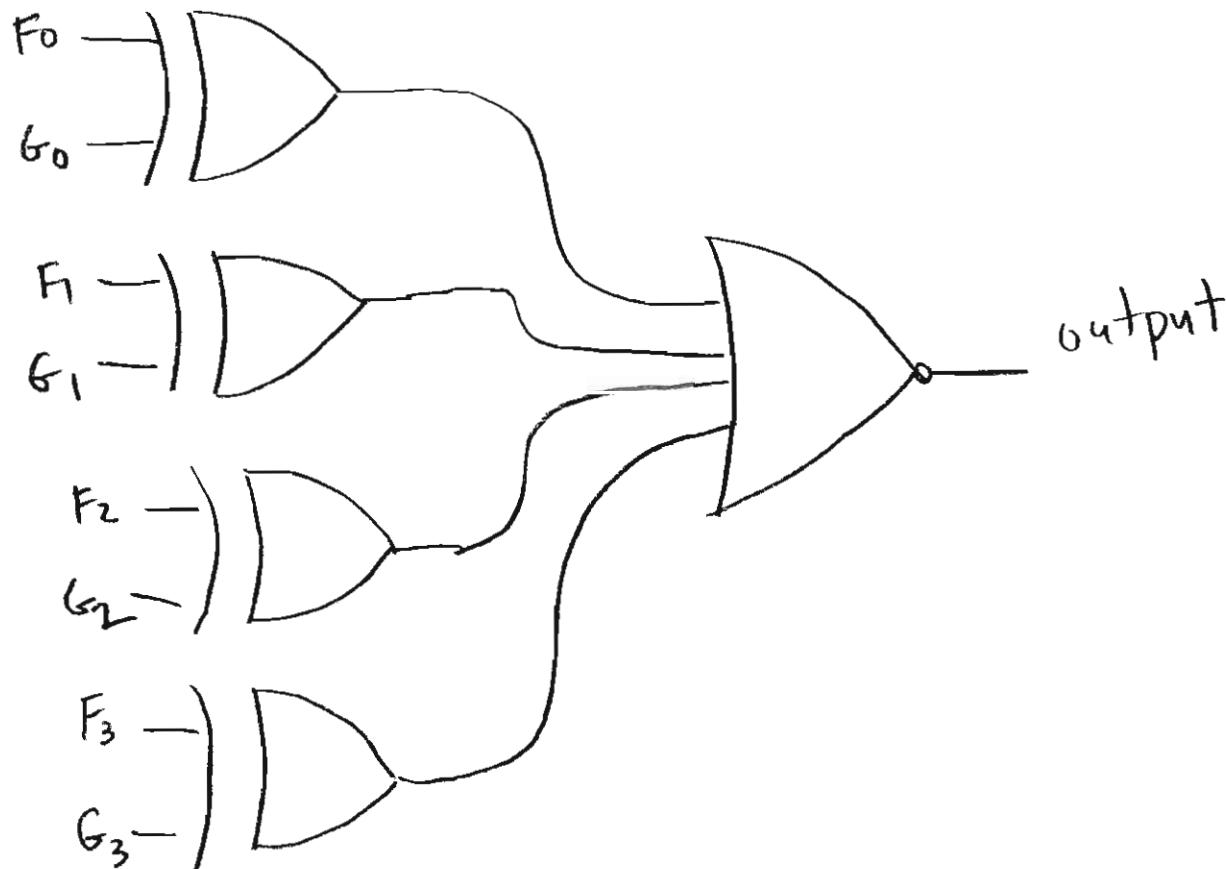
Example 5: Design a basic comparator for 4-bit strings (F and G) designed with X-Nor and And Gates.

$$F = F_3F_2F_1F_0$$
$$G = G_3G_2G_1G_0$$



Example 6: Design a basic comparator for 4-bit strings designed with X-Or and Nor Gates.

$$F = F_3F_2F_1F_0$$
$$G = G_3G_2G_1G_0$$

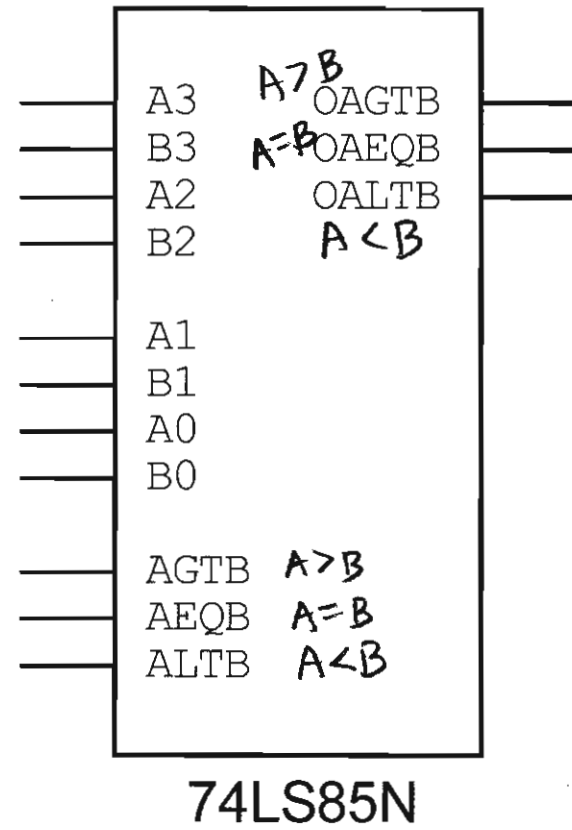
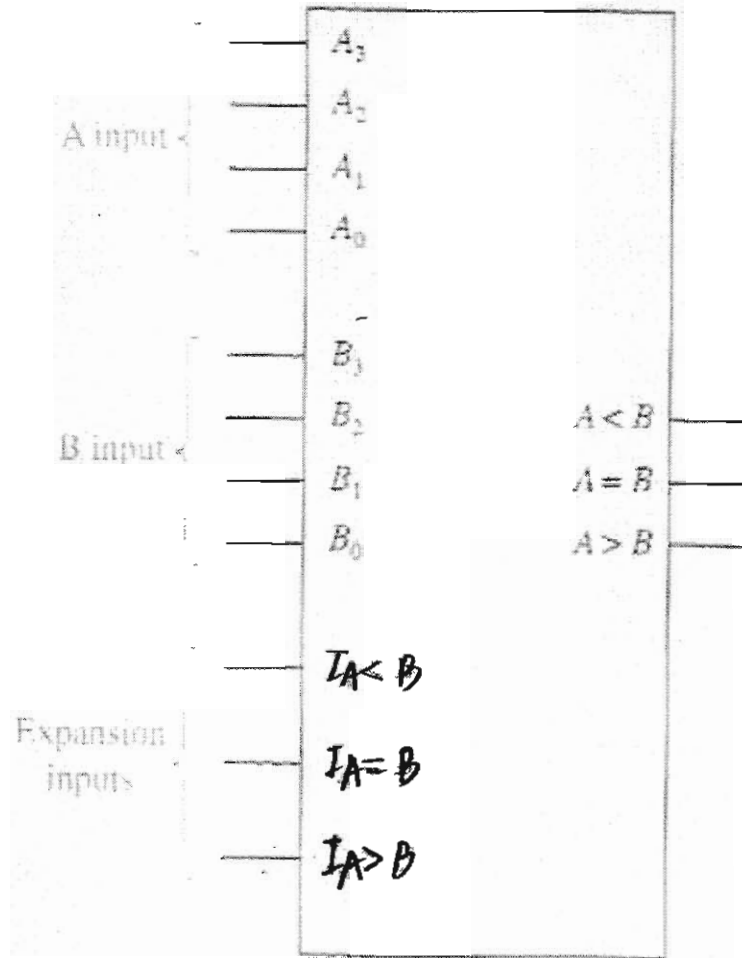


Integrated-circuit Magnitude Comparator

Compare two binary strings (A and B) with the following outputs:

- $A = B$
- $A > B$
- $A < B$

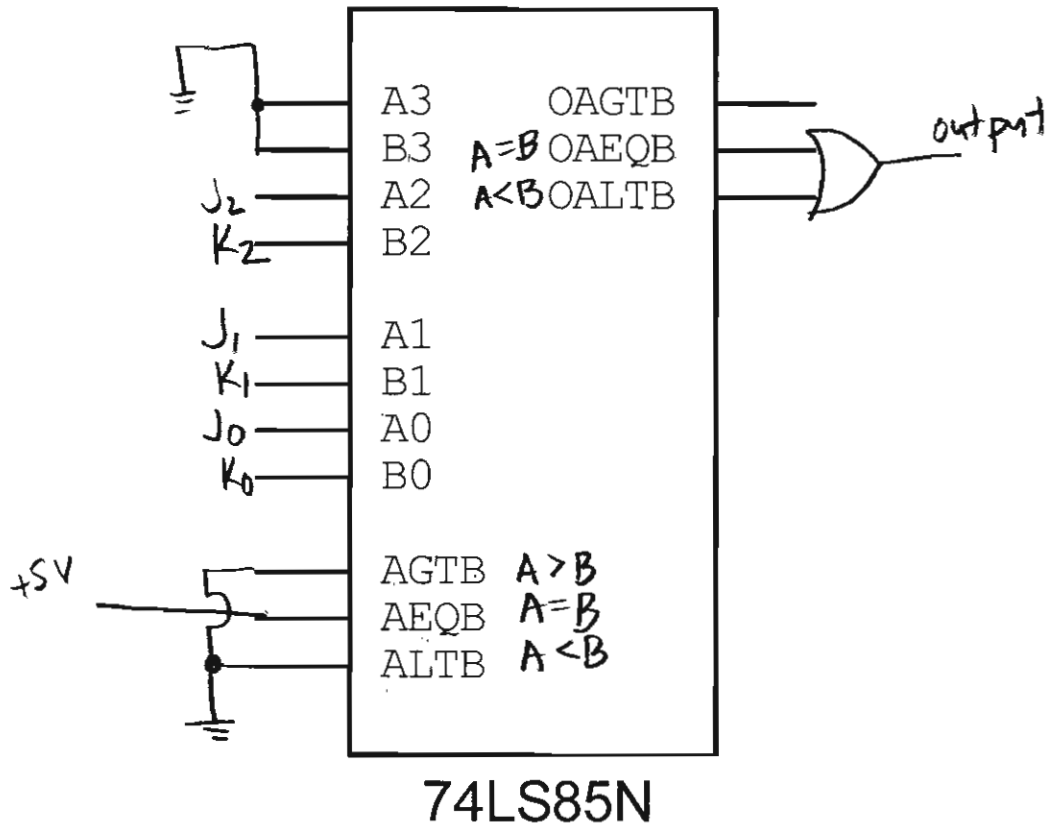
Logic Diagram of 7485 Magnitude Comparator



Example 7: Use the 7485 chip to design a 3-bit Comparator for strings J and K.

$J = J_2 J_1 J_0$
 $K = K_2 K_1 K_0$

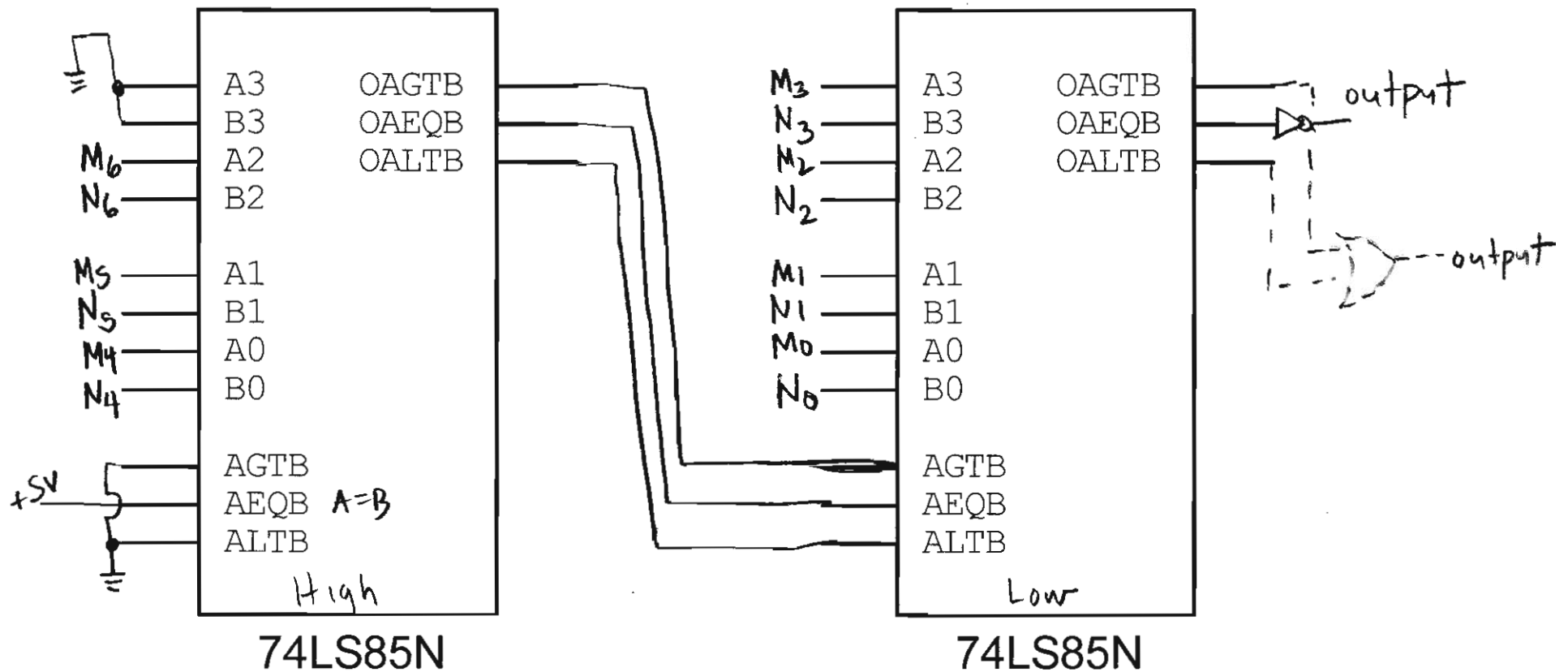
$J \leq K$
 $A \leq B$



Example 8: Use the 7485 chip to design a 7-bit Comparator for strings M and N. $M \neq N$

$$M = M_6 M_5 M_4 M_3 M_2 M_1 M_0$$

$$N = N_6 N_5 N_4 N_3 N_2 N_1 N_0$$



Active High Level:

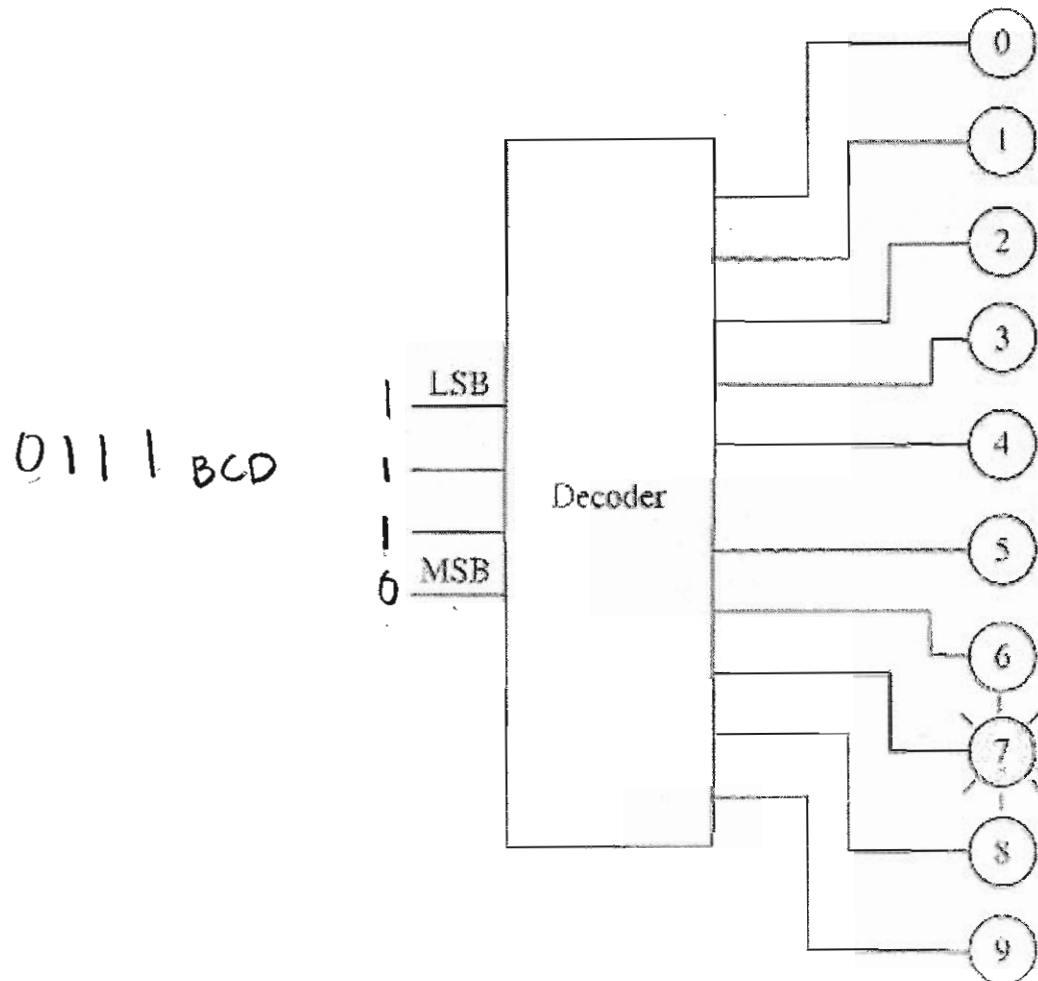
Function is active when it is receiving/sending a high voltage.

Active Low Level:

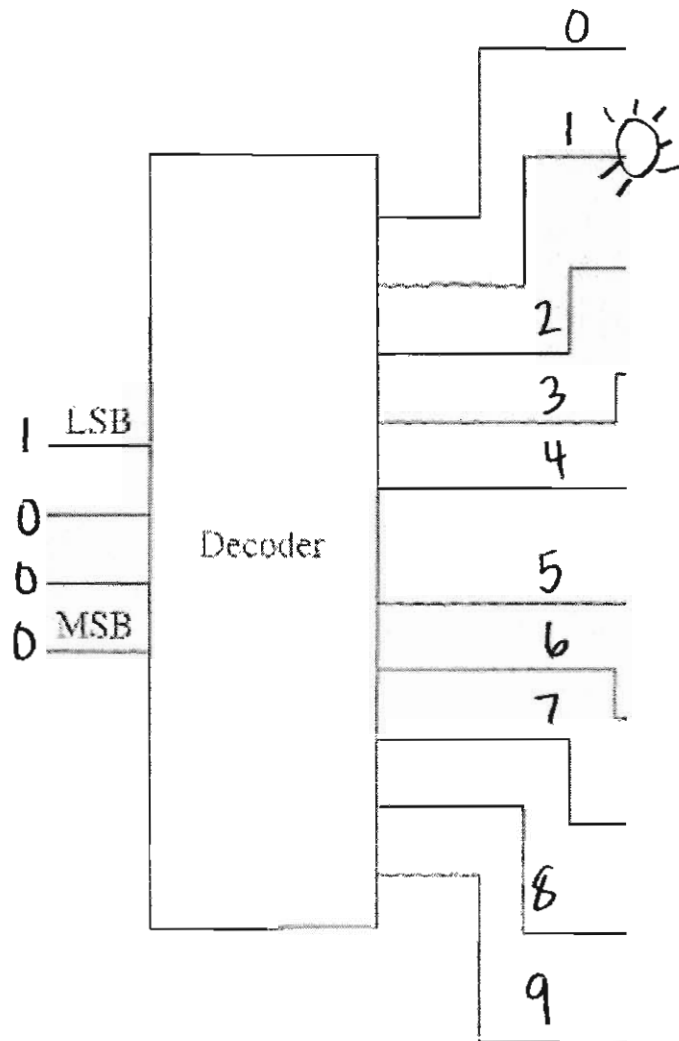
Function is active when it is receiving/sending a low voltage.

Decoder:

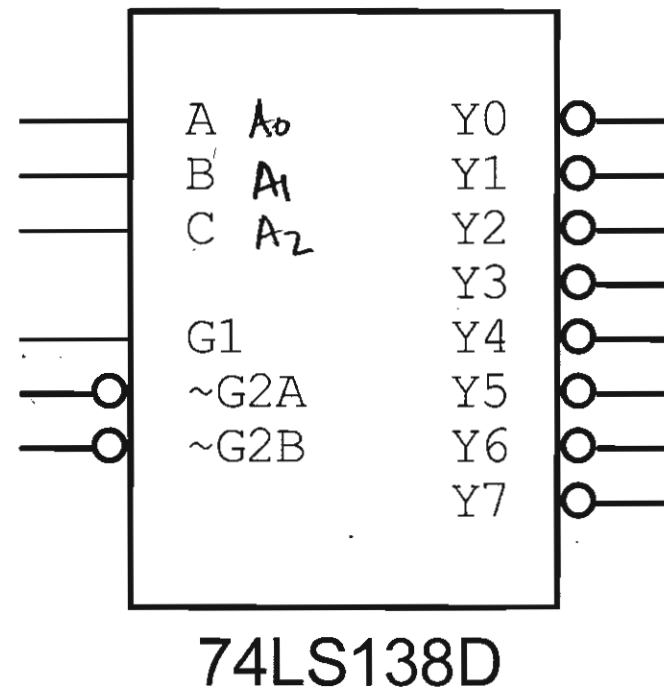
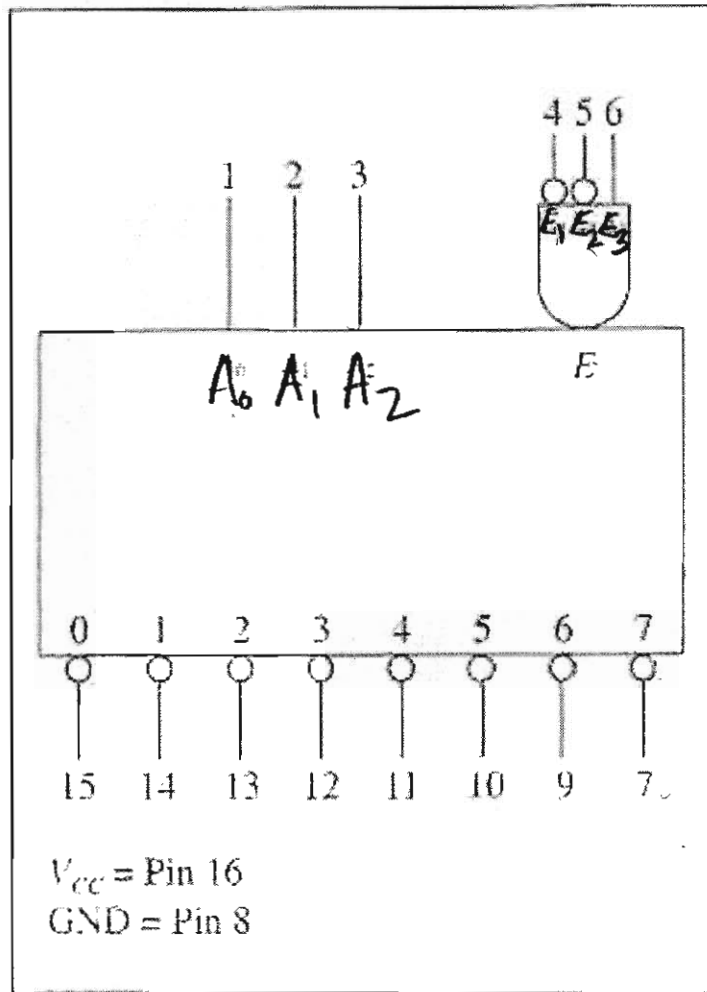
- Decodes a string of binary inputs to a single output representing the inputs.
- Binary inputs represent octal, hex, or BCD values.
- values.



Example 9: What binary string is needed to light Lamp-1 connected to an octal-decoder.

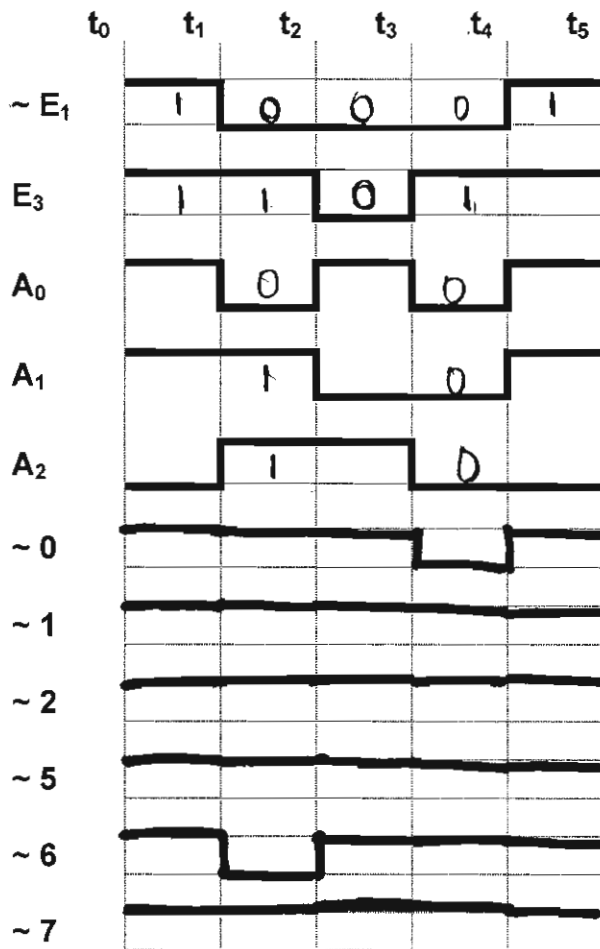


Logic Diagram of 74138 Octal Decoder



Example 10: Sketch the output waveforms given the inputs shown for a 74138 Octal Decoder

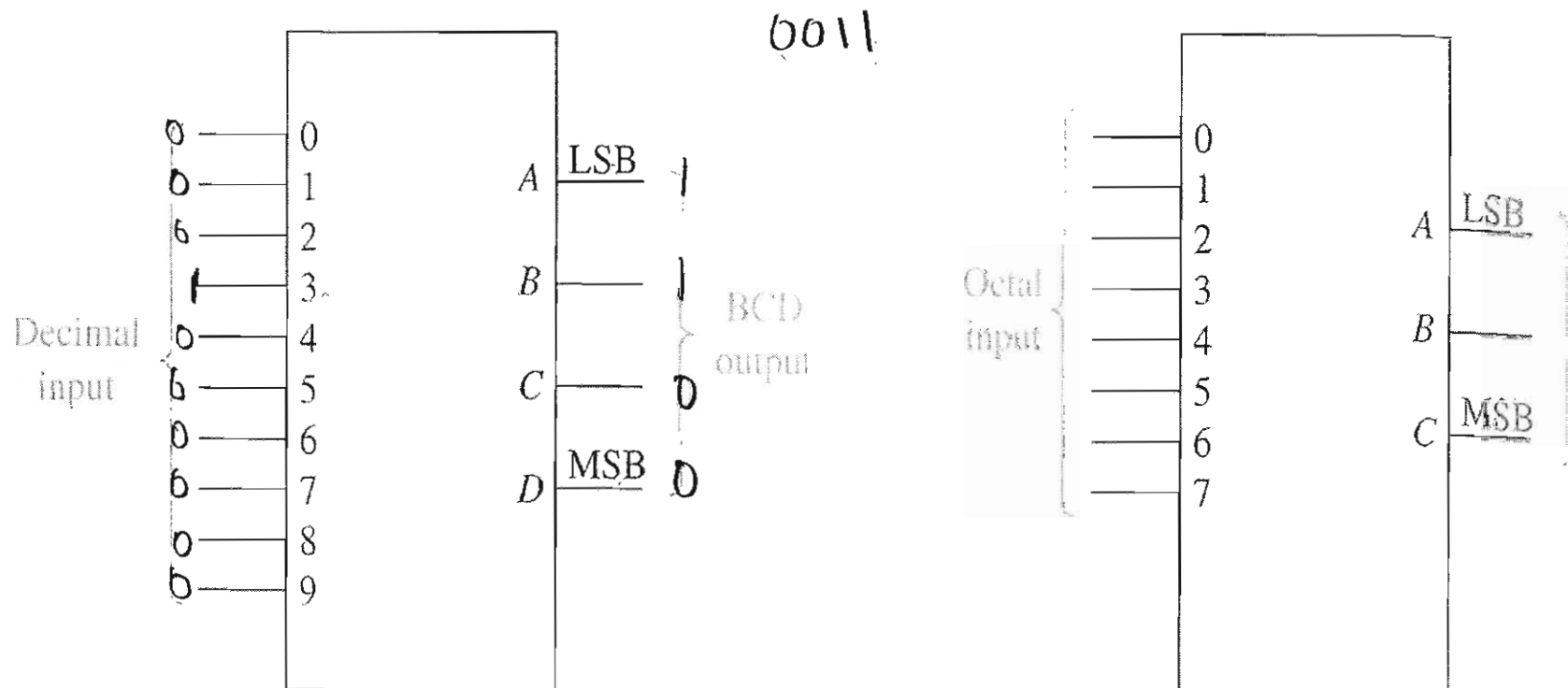
$\sim E_2 = 0$



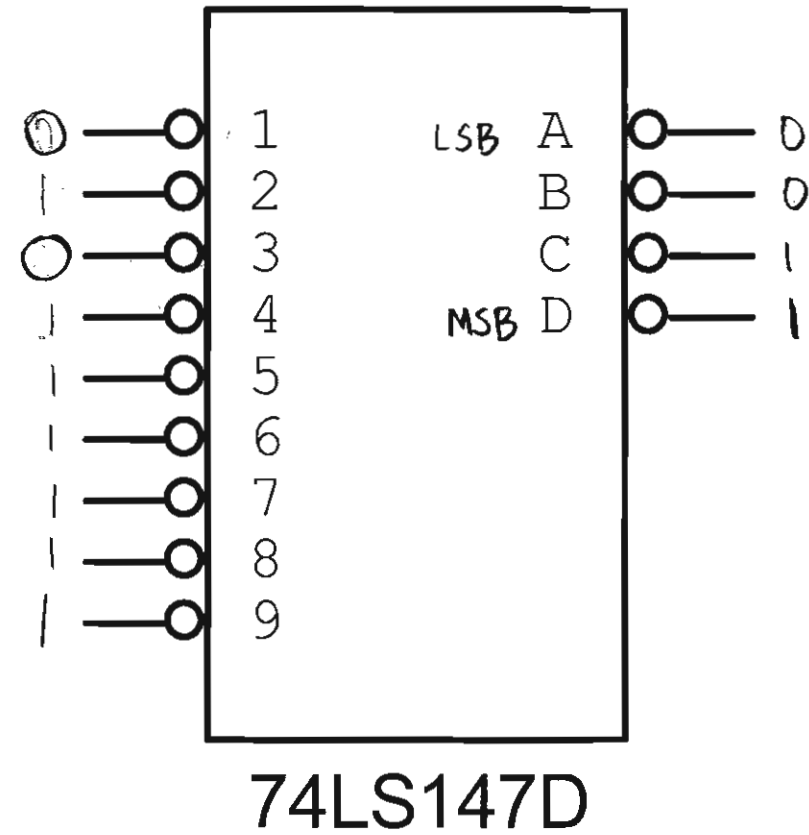
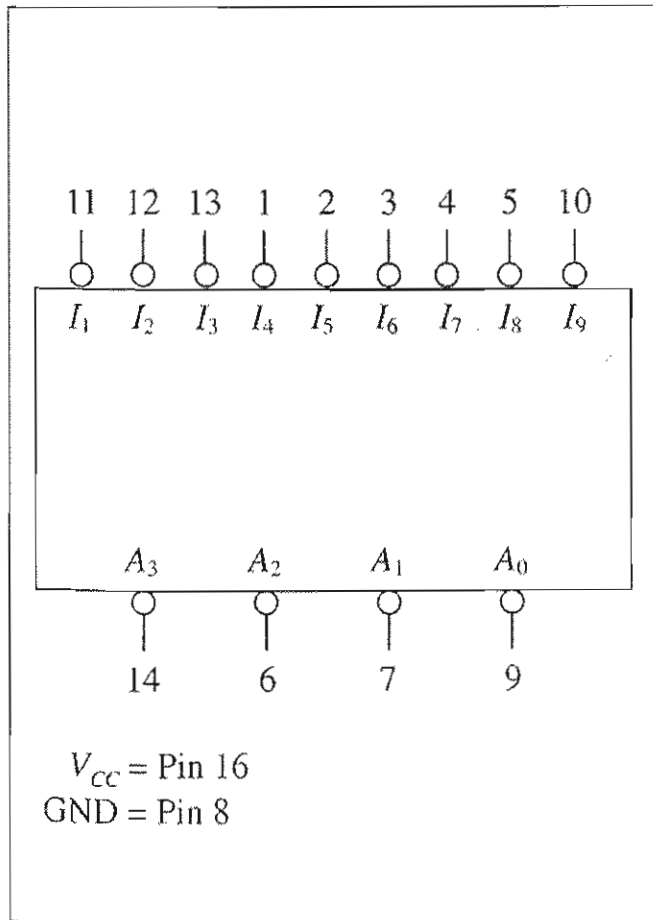
Inputs						Outputs							
\bar{E}_1	\bar{E}_2	E_3	A_0	A_1	A_2	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

Encoder:

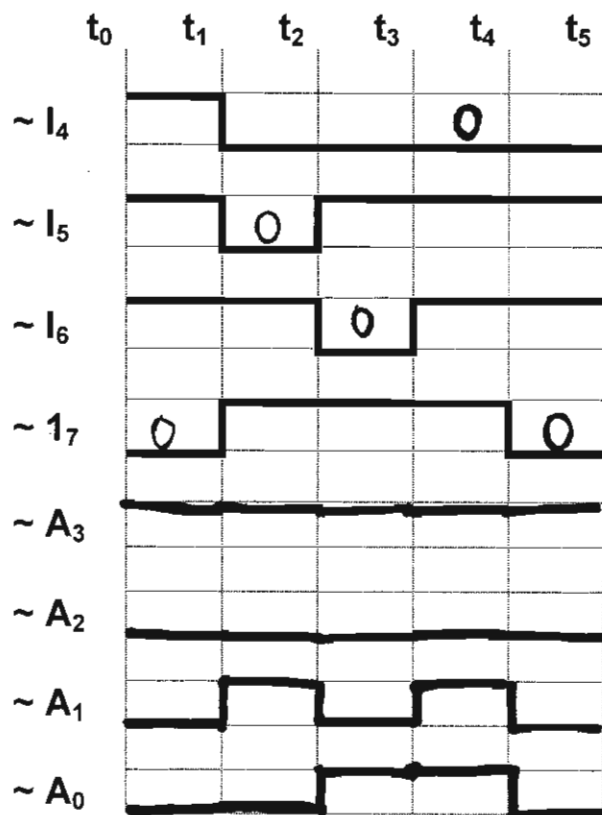
- Opposite a decoder
- Takes a single input representing a distinct value and converts it to a binary representation of the distinct value.
- Binary representation can be BCD, Hex, or Octal value.



Logic Diagram of 74147 decimal-to-BCD priority Encoder.

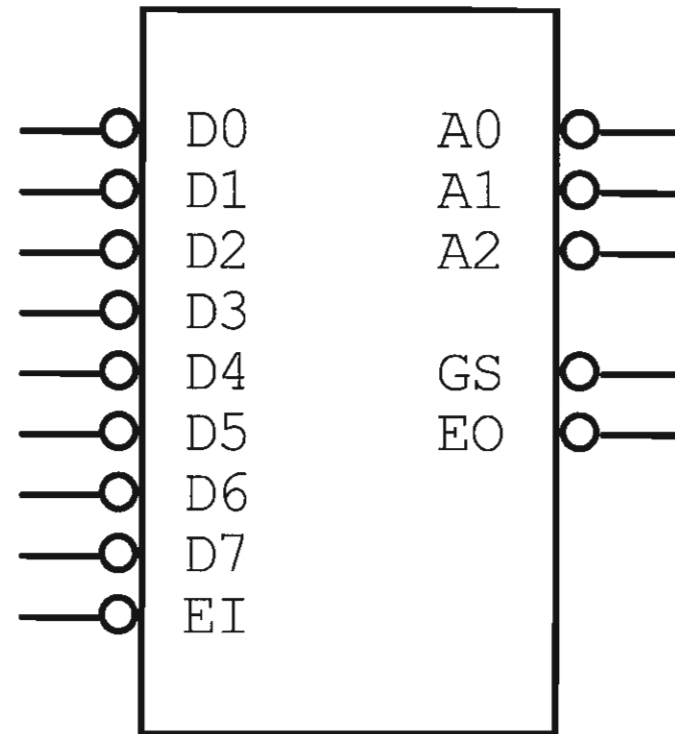
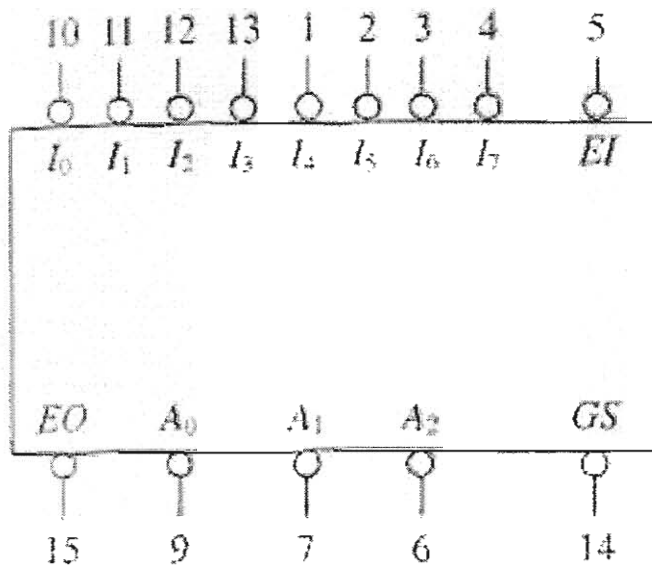


Example 11: Sketch the outputs ($\overline{A_0}, \overline{A_1}, \overline{A_2}, \overline{A_3}$) as the following inputs ($\overline{I_4}, \overline{I_5}, \overline{I_6}, \overline{I_7}$) are switching. The remaining inputs are set to high for the 74147 decimal-to-BCD priority Encoder.



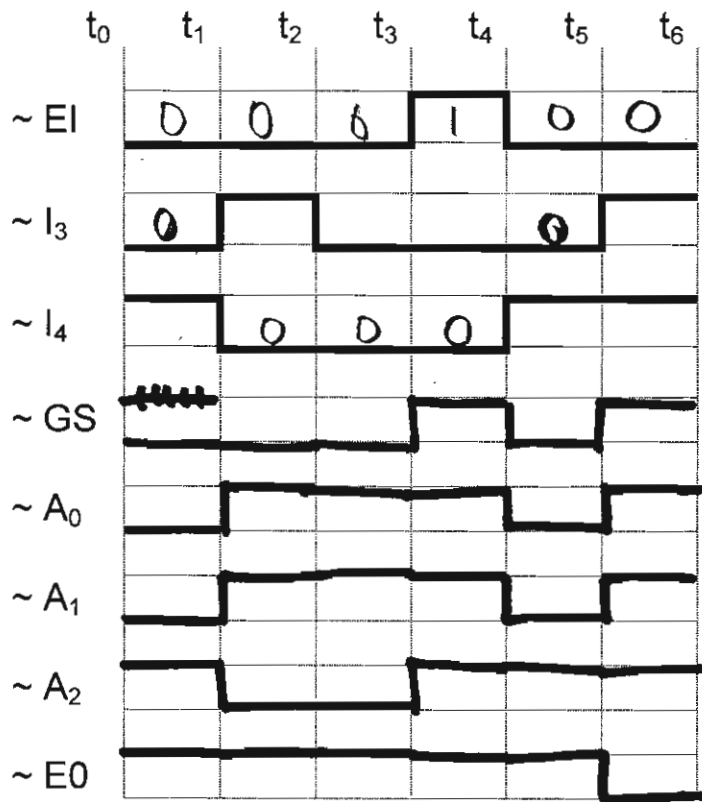
Input									Output			
$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	$\overline{I_8}$	$\overline{I_9}$	$\overline{A_3}$	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	L	H	H	H	H	H	L	L	H
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

Logic Diagram of 74148 octal-to-binary priority Encoder.



74LS148N

Problem 8-15 pg 348: Sketch the output waveforms ($\overline{A_0}, \overline{A_1}, \overline{A_2}, \overline{EO}, \overline{GS}$) given that the inputs ($\overline{I_0}, \overline{I_1}, \overline{I_2}, \overline{I_5}, \overline{I_6}, \overline{I_7}$) are set high for 74148 chip.



Inputs									Outputs				
\overline{EI}	$\overline{I_0}$	$\overline{I_1}$	$\overline{I_2}$	$\overline{I_3}$	$\overline{I_4}$	$\overline{I_5}$	$\overline{I_6}$	$\overline{I_7}$	\overline{GS}	$\overline{A_0}$	$\overline{A_1}$	$\overline{A_2}$	\overline{EO}
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	L	H	L	H	H
L	X	L	H	H	H	H	H	H	L	L	H	H	H
L	L	H	H	H	H	H	H	H	L	H	H	H	H