Digital Logic Design
(EL-1005)
LABORATORY MANUAL
Spring-2025

LAB 08
Binary Decoder

_____ _____ ___

STUDENT NAME ROLL NO SEC

_____

INSTRUCTOR SIGNATURE& DATE

MARKS AWARDED: /10

Lab Session 08: Binary Decoder

OBJECTIVES:

- Define basics of decoding mechanism
- Explain the working principle of 2 – 4 Decoder, 3-8-line Decoder & BCD to Seven Segment Decoder.
- Understand the usage of Seven Segment Display
- Familiarize with some important terminologies like
  - Common Anode & Common Cathode Display
  - Active High enable & Active Low enable devices

APPARATUS: Logic trainer, Logic probe

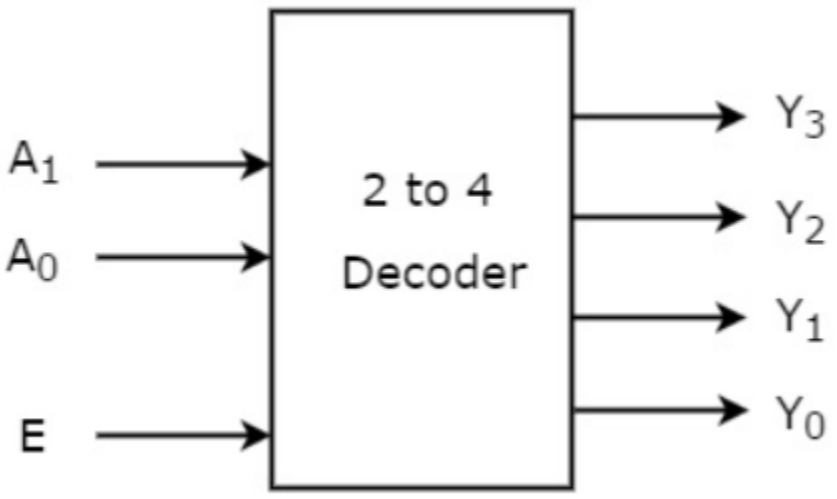
COMPONENTS: **74LS138 (3-8-line decoder), 74LS47 (BCD to Seven Segment Decoder)**

THEORY:

Decoder is a multiple-input, multiple-output logic circuit that converts coded inputs into coded output coded outputs. The basic function of decoder is to detect the presence of a specified combination of bits (code) at its input and indicate the presence of that code by a specified output. Various kinds of decoding include n-to-2^n decoding & binary-coded decimal decoding. Decoder has Enable inputs which must be on for the decoder to function.

a. 2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A_1 & A_0 and four outputs Y_3, Y_2, Y_1 & Y_0. The **block diagram** of 2 to 4 decoder is shown in the following figure.



One of these four outputs will be '1' for each combination of inputs when enable, E is '1'. The **Truth table** of 2 to 4 decoder is shown below.

Enable	Inputs		Outputs			
E	**A_1**	**A_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

From Truth table, we can write the **Boolean functions** for each output as

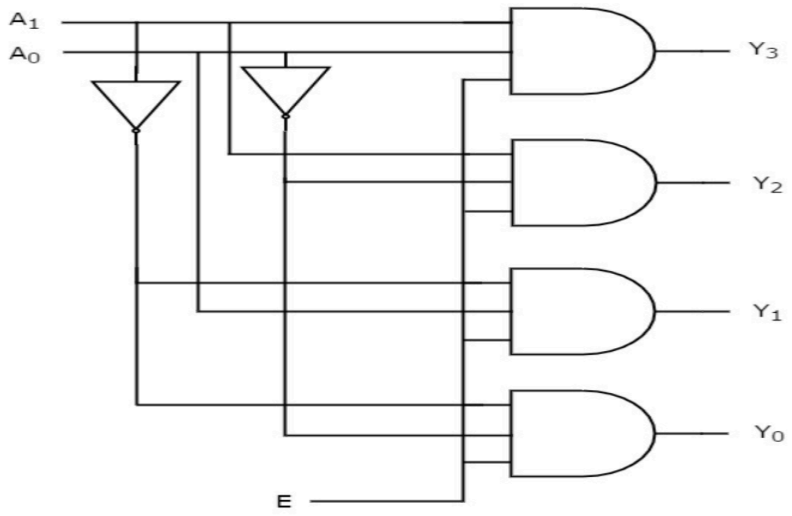
$$Y_3 = E.A_1.A_0$$

$$Y_2 = E.A_1.A_0{}'$$

$$Y_1 = E.A_1{}'.A_0$$

$$Y_0 = E.A_1{}'.A_0{}'$$

Each output is having one product term. So, there are four product terms in total. We can implement these four product terms by using four AND gates having three inputs each & two inverters. The **circuit diagram** of 2 to 4 decoder is shown in the following figure.



a. Implementation of 3-8-line Decoder using IC 74LS138

This decoder decodes one of eight lines depending on the three binary select inputs and three enable pins. It is an octal decoder capable of decoding eight possible three-bit combinations to eight separate active-Low outputs.

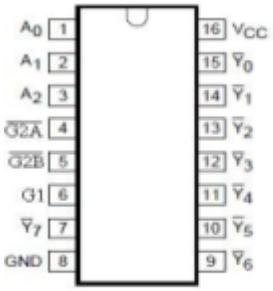


Fig 1. Pin Configuration

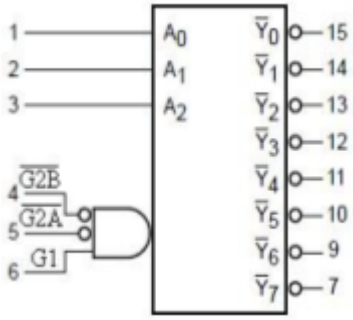
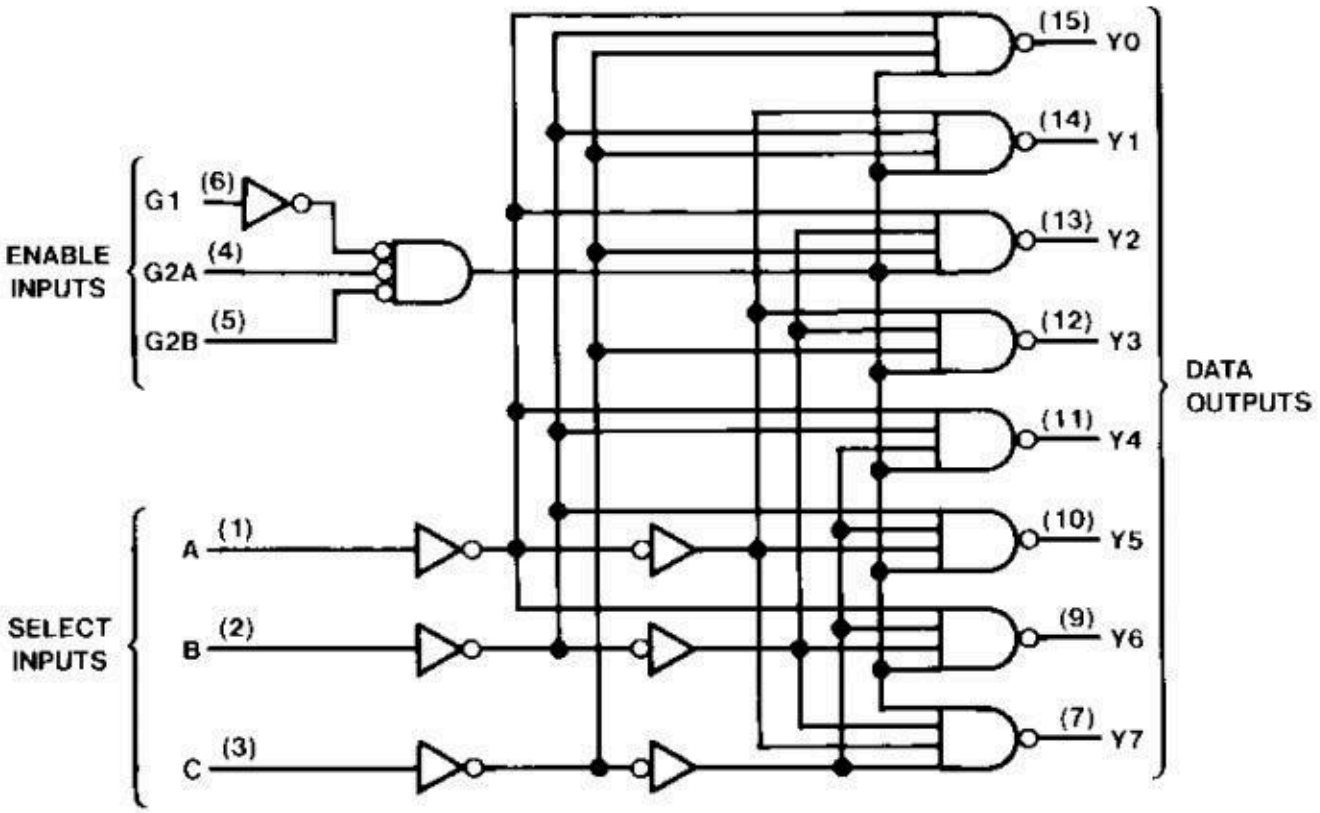


Fig 2. Logic Symbol

$A_0 - A_2$	Input Bits
G_{2A}', G_{2B}'	Enable (Active LOW) Inputs
G_1	Enable (Active HIGH) Inputs
Y_0' - Y_7'	Active LOW Outputs



b. Implementation of BCD to Seven Segment Decoder using IC 74LS47

The BCD to Seven Segment Decoder accepts BCD code on its input and provides outputs to derive Seven-segment display devices to produce a decimal read-out by turning on the appropriate LED segments.

If the Seven Segment display is a common-anode display that is an active Low (Low- enable) device because it takes 0 to turn on a segment, the decoder IC to be used must also have active Low outputs. IC 74LS47 has active low outputs therefore it will require a common-anode display device for compatibility.

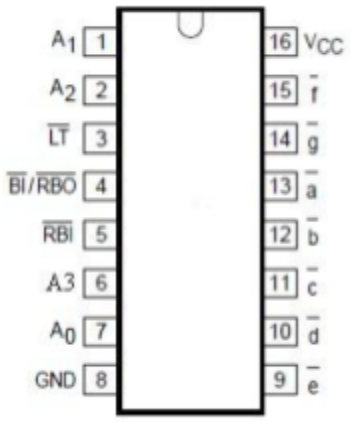


Fig 3. Pin Configuration

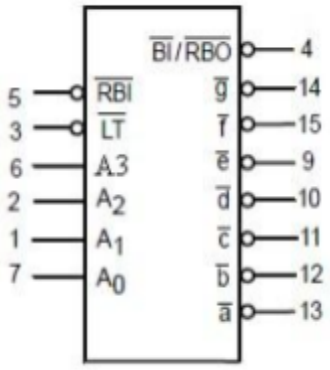
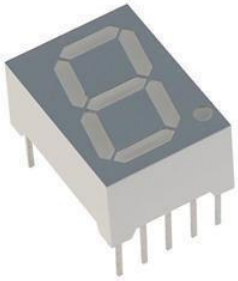
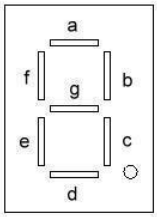


Fig 4. Logic Symbol

A0 – A3	BCD Inputs
RBI’	Ripple Blanking Input (Active LOW)
LT’	Lamp Test Input (Active LOW)
BI’/RBO’	Blanking Input or Ripple Blanking Output (Active LOW)
a' – g'	Active LOW Outputs

Seven –Segment Display





Common Anode Display Active low inputs

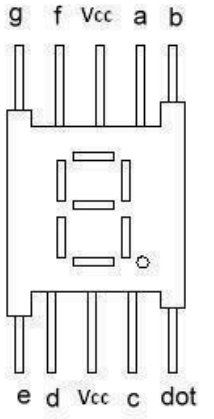


Fig 6. Pin Configuration

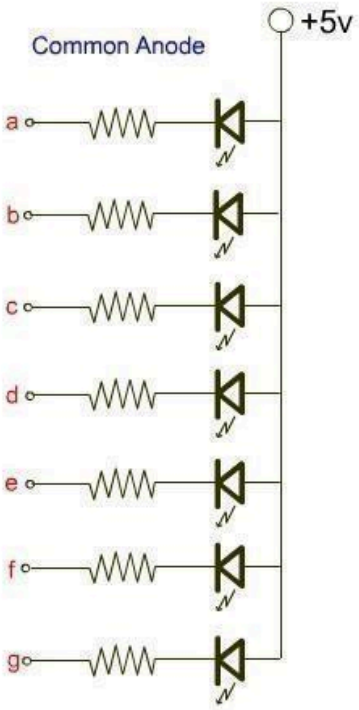


Fig 7. Internal circuit

LAB TASK

Exercise # 01

Implementation of 3-8-line Decoder using IC 74LS138

a. 3-8-line Decoder

INPUTS						OUTPUTS							
Enable			Select										
G_1	G_{2A}'	G_{2B}'	A_2	A_1	A_0	Y_0'	Y_1'	Y_2'	Y_3'	Y_4'	Y_5'	Y_6'	Y_7'

Exercise # 02

Implementation of BCD to Seven Segment Decoder using IC 74LS47

b. BCD to Seven Segment Decoder

INPUTS				OUTPUTS						
A_3	A_2	A_1	A_0	a'	b'	c'	d'	e'	f'	g'

Exercise #03:

Implement a 4-16 bit decoder using the 2-4 decoder IC (Hint: for enable use the last inputs)