JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA

DEPARMENT OF COMPUTER SCIENCE & ENGINEERING

Course: Computer Organization & Architecture Lab

Course Code: 18B17CI474

B. Tech. (CSE VI Sem.)

Experiment # 4

Aim: Design of combinational logic circuits.

A combinational logic circuit is the digital system whose **outputs depend only on its current combination of input values** and remains independent of previous output values. All logic gates, adders/subtractors, comparators, multiplexers, de-multiplexers, encoders, decoders, ALUs etc. are the examples of the combinational circuits. Other important features of these circuits are as following:-

- No feedback connections in the circuits.
- No clock signal i.e. time independent
- Memoryless
- Helps in reducing design complexity

In this experiment, only the following four combinational circuits to be designed:-

- **Multiplexer:** A multiplexer, also known as data selector, is a digital switch which allows digital information from several sources to be routed onto a single output line. A set of ($\boldsymbol{n}$) **selection/control lines** control the selection of a particular input line. Therefore, a multiplexer ($size$: $\boldsymbol{2^n \times 1}$) is a **multiple-input** ($\boldsymbol{m = 2^n}$) and **single-output switch**.

- **De-multiplexer:** A circuit that receives information on a single input line and transmits the information on any of the $\boldsymbol{m = 2^n}$ possible output lines is called as a de-multiplexer. Therefore, a de-multiplexer ($size$: $\boldsymbol{1 \times 2^n}$) is called a **single input multiple-output switch**. The values of $\boldsymbol{n}$ selection lines control the selection of specific output line.

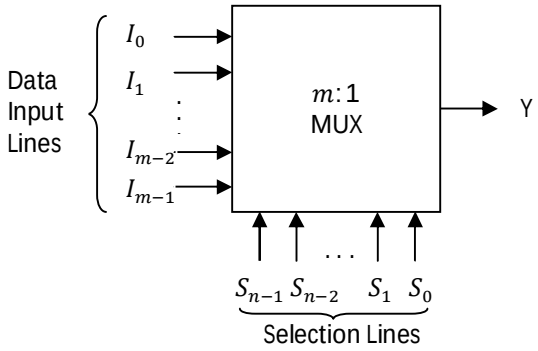
- **Decoder:** A decoder is a **multiple input** and **multiple output** logic circuit. A decoder converts coded inputs into a differently coded outputs. Often, the input code has fewer bits than the output code. Each input combination produces only one active output. A decoder circuit of size ($\boldsymbol{n \times m}$) has $\boldsymbol{n}$ inputs and produces $\boldsymbol{m = 2^n}$ possible outputs varies from **0** to $\boldsymbol{2^{n-1}}$. Usually, a decoder is provided with **enabled inputs** to activate decoded output based on data inputs. It can be implemented using **AND/NAND gates.**

- **Encoder:** A digital circuit that performs the **inverse operation** of a decoder is called as an encoder. An encoder of size ($\boldsymbol{m \times n}$) has $\boldsymbol{m = 2^n}$ (or less) input lines, $\boldsymbol{n}$ output lines and there may be an **enable line**. In an encoder, the output lines generate the binary code corresponding to the input value. It can be implemented using **OR gates** whose inputs can be determined directly from the truth table.

Note: Above mentioned circuits can be implemented in larger size using two or more smaller size of similar circuits suitably. For example, a 3 to 8 decoder can be obtained using two 2 to 4 decoders.

Multiplexer

Block Diagram



Boolean Expression

$$Y = \sum_{m=0}^{2^n-1} (S_0 S_1 \dots S_{n-2} S_{n-1}) I_m$$

Select lines will hold the binary values equivalent to the decimal value ($\boldsymbol{m}$) of data lines. For Example, the Boolean expression for **4 to 1** MUX is as:-

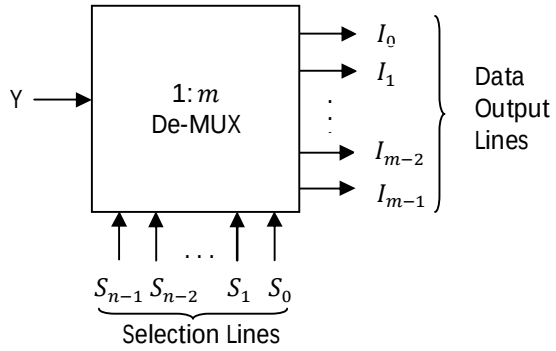
$$Y = \overline{S_1}\,\overline{S_0}\, I_0 + \overline{S_1}\, S_0\, I_1 + S_1\,\overline{S_0}\, I_2 + S_0 S_1\, I_3$$

Truth Table (4-to-1 MUX)

Select Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

De-Multiplexer

Block Diagram



Boolean Expression

$$I_m = (S_0 S_1 \dots S_{n-2} S_{n-1}) Y$$

Only one data line of decimal value $\boldsymbol{m}$ equivalent to the binary values of select lines will be active at a time. For Example, the Boolean expression for **1 to 4** De-MUX is as:-

$$I_m = \overline{S_1}\,\overline{S_0} Y + \overline{S_1}\, S_0 Y + S_1\,\overline{S_0} Y + S_0 S_1 Y$$

Truth Table (1-to-4 De-MUX)

Input	Select Lines		Output
Y	S_1	S_0	I_m
Y	0	0	$I_0 = Y$
Y	0	1	$I_1 = Y$
Y	1	0	$I_2 = Y$
Y	1	1	$I_3 = Y$

Exercise#1: Design a half adder using two 4:1 multiplexers (shown in Fig.1) in **structural style of architecture.**

Exercise#2: Design 8:1 multiplexer using two 4:1 and one 2:1 multiplexers (shown in Fig.2) in **structural style of architecture**. Use inputs and selection lines in the form of bus.

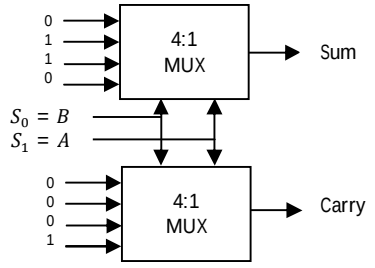


Figure 1: Half Adder using Multiplexers

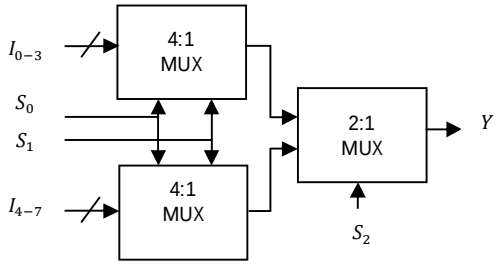
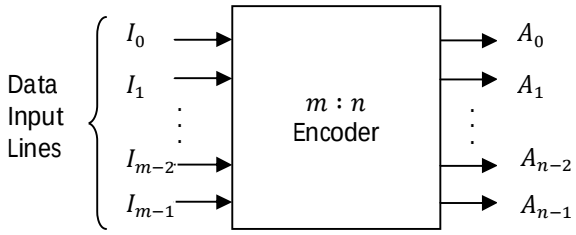


Figure 2:8-to-1 Multiplexer

Encoder

Block Diagram



Boolean Expressions (8-to-3 Encoder)

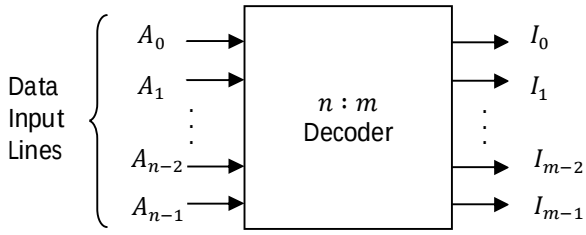
$$LSB: A_0 = I_1 + I_3 + I_5 + I_7$$
$$A_1 = I_2 + I_3 + I_6 + I_7$$
$$MSB: A_2 = I_4 + I_5 + I_6 + I_7$$

Truth Table (8-to-3 Encoder)

Inputs								Outputs		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	A_2	A_1	A_0
0	0	0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	**1**	0	0	0	1
0	0	0	0	0	**1**	0	0	0	1	0
0	0	0	0	**1**	0	0	0	0	1	1
0	0	0	**1**	0	0	0	0	1	0	0
0	0	**1**	0	0	0	0	0	1	0	1
0	**1**	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Decoder

Block Diagram



Boolean Expression (3-to-8 Decoder)

$$I_m = \overline{A_2}\,\overline{A_1}\,\overline{A_0} + \overline{A_2}\,\overline{A_1}\,A_0 + \overline{A_2}\,A_1\,\overline{A_0} + \overline{A_2}\,A_1A_0 + A_2\overline{A_1}\,\overline{A_0} + A_2\overline{A_1}A_0 + A_2A_1\overline{A_0} + A_2A_1A_0$$

Only one output line of decimal value $\boldsymbol{m}$ equivalent to the binary values of data lines will be active at a time.

Truth Table (3-to-8 Decoder)

Inputs			Outputs							
A_2	A_1	A_0	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
0	0	0	0	0	0	0	0	0	0	**1**
0	0	1	0	0	0	0	0	0	**1**	0
0	1	0	0	0	0	0	0	**1**	0	0
0	1	1	0	0	0	0	**1**	0	0	0
1	0	0	0	0	0	**1**	0	0	0	0
1	0	1	0	0	**1**	0	0	0	0	0
1	1	0	0	**1**	0	0	0	0	0	0
1	1	1	**1**	0	0	0	0	0	0	0

Exercise#3: Design 3-to-8 decoder using two 2-to-4 decoders with enable line shown in Fig. 3 in **structural style of architecture**. Use inputs and outputs in the form of bus.

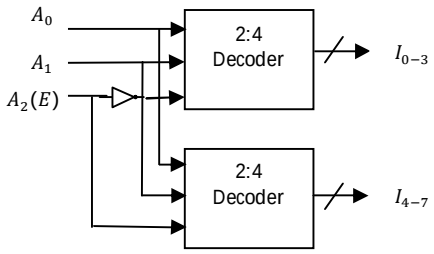


Fig. 3: 3-to-8 Decoder