LECTURE 8: DIGITAL LOGIC CIRCUIT DESIGN

Decoders

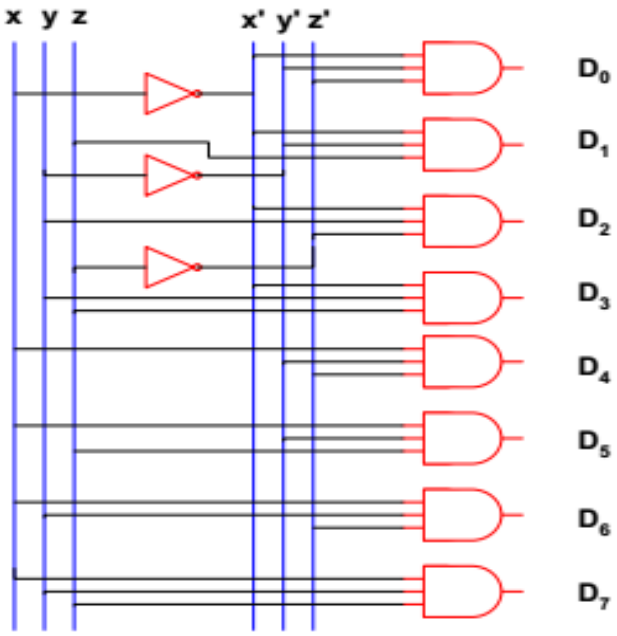
A binary code of n bits is capable of representing up to 2^n distinct elements of coded information. A decoder is a combinational circuit that converts binary information from n input lines to up to 2^n output lines. These decoders are called n-to-m line decoders such that:

$$m \leq 2^n$$

3-to-8 Line Decoder

A 3-to-8 line decoder has three elements and eight outputs. The decoder decodes the input binary code represented by the three bits and generates all eight minterms of the inputs. Only one output is one while the other seven are zeros.

This decoder can be implemented using three inverters and eight AND gates as shown.



3-to-8 Line Decoder

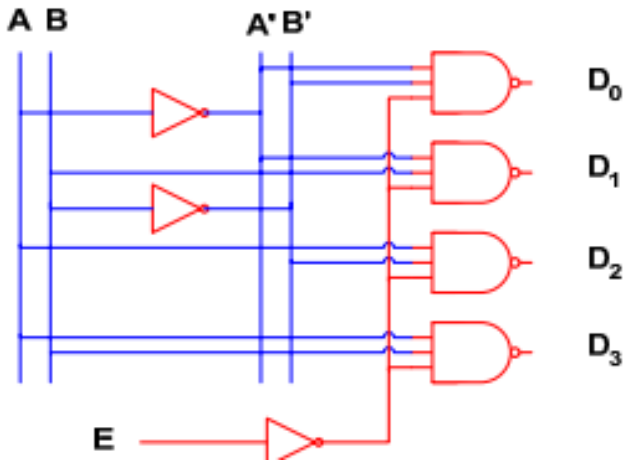
LECTURE 8: DIGITAL LOGIC CIRCUIT DESIGN

The following truth table is for the decoder.

Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

2-to-4 Line Decoder With Enable Input Using NAND gates

If we use NAND gates to construct the decoder then the outputs are inverted. Decoders are also constructed with one or more enable inputs. An example is shown for the 2-to4 line decoder.



E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

Demultiplexers

A decoder with enable input can function as a demultiplexer. A demultiplexer is a combinational circuit that has one input and up to 2^n outputs and it directs the input to an output depending on the values of n selection lines.

A 4 X 16 decoder can be constructed from two 3 X 8 decoders with enable inputs.

Combinational Logic Implementation

Decoders can be used to implement logic functions. Since all the minterms of the function are available at the output then there is no need for simplification. All what is needed is an OR gate for each function to sum the required minterms.

Example:

Implement a full adder circuit using an appropriate decoder and OR gates.

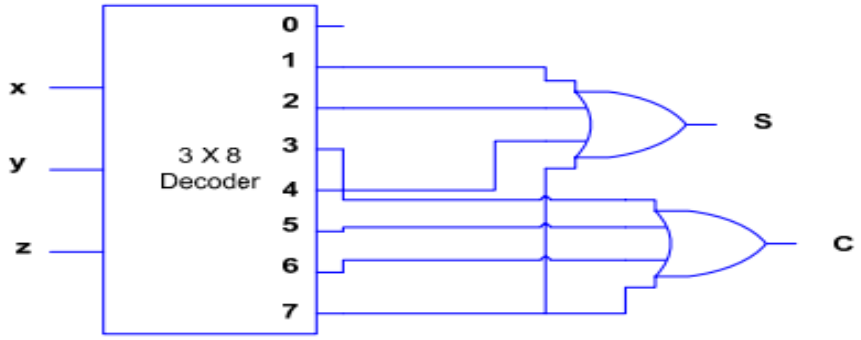
The required outputs are:

Sum → $S(x,y,z) = \Sigma(1,2,4,7)$

Carry → $C(x,y,z) = \Sigma(3,5,6,7)$

LECTURE 8: DIGITAL LOGIC CIRCUIT DESIGN

The implementation will be as follows:



Encoders

An encoder is a digital circuit that performs the inverse operation of the decoder. It has 2^n inputs and n outputs that represents the code of the order of the input that is set to one. The truth table of an octal to binary encoder is shown below.

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

The encoder is implements by OR gates. As given in the truth table, the outputs are given by:

$$X = D_4 + D_5 + D_6 + D_7$$

$$Y = D_2 + D_3 + D_6 + D_7$$

$$Z = D_1 + D_3 + D_5 + D_7$$

This encoder has two main drawbacks:

1. **When more than one input is 1 at the same time, then the output could indicate a wrong code. E.g. D5 and D6 are one at the same time, then x,y, and z are ones indicating D7 is one.**

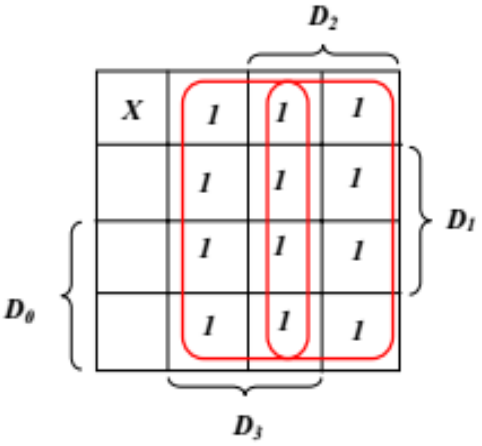
2. If no input is one, which is not valid code, then the outputs are all zeros, which indicates a code for D0.

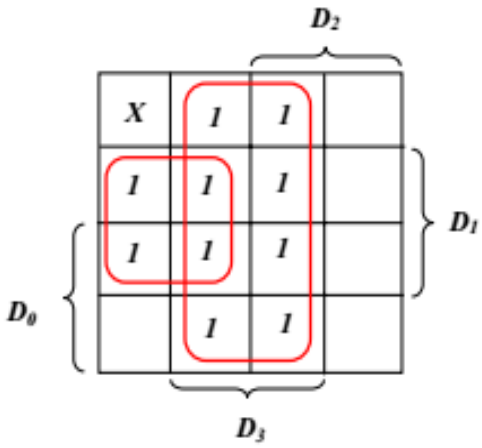
To overcome these problems, we may use a priority encoder.

The truth table of a 4 to 2 priority encoder is given below:

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Using Karnaugh maps to simplify the output functions:





$X = D_2 + D_3$

$Y = D_3 + D_1 D_2'$

$V = D_0 + D_1 + D_2 + D_3$

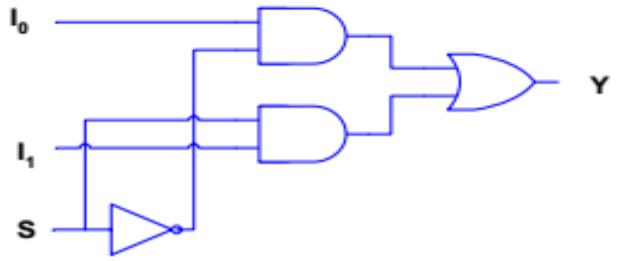
LECTURE 8: DIGITAL LOGIC CIRCUIT DESIGN

Multiplexers

A multiplexer is a combinational circuit that selects one of many input lines (normally 2^n lines) and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines (normally n selection lines).

2-to-1 Line Multiplexer

A 2-to-1 line multiplexer has two inputs, one selection line and one output. This is shown in the following logic circuit.

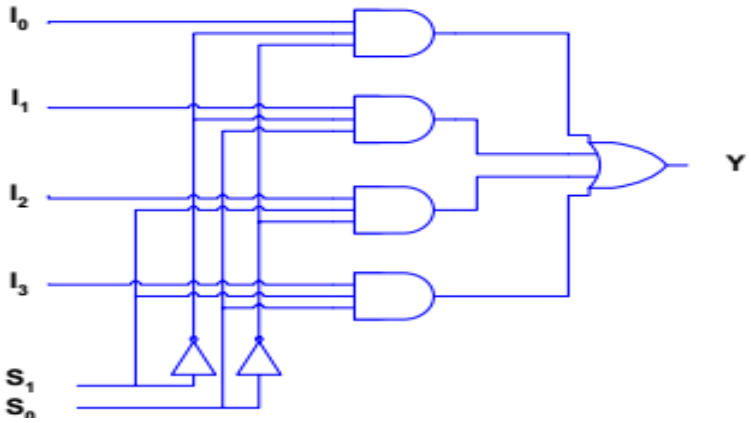


4-to-1 Line Multiplexer

A 4-to-1 line multiplexer consists of four AND gates. Each input is connected to one AND gate. Selection lines S1 and S0 are decoded to select a particular AND gate. The outputs of the AND gates are applied to a single OR gate that provides the output of the multiplexer Y.

The output of the multiplexer is then given by:

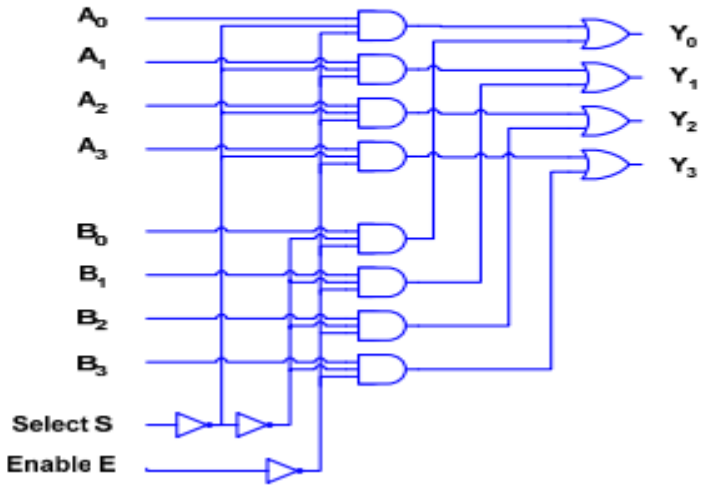
$$Y = S_1' S_0' I_0 + S_1' S_0 I_1 + S_1 S_0' I_2 + S_1 S_0 I_3$$



The function table of the multiplexer is shown next.

S_1	S_0	Y
0	O	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Multiplexers may have an enable input, similar to decoders, to control the operation of the unit. A quadruple 2-to-1 multiplexer with enable input is shown next.



The function table of the quadruple 2-to-1 multiplexer with the enable input will be as follows:

E	S	Y
1	X	All 0's
0	0	Select A
0	1	Select B

Boolean Function Implementation

A multiplexer is a decoder and an OR gate that provides the output. The multiplexer can be used to implement Boolean

functions of n variables. This can be achieved using either 2^n-to-1 multiplexer or $2^{(n-1)}$-to-1 multiplexer.

1. Using 2^n-to-1 multiplexer

The n variables are connected to the n selection lines. Each input of the multiplexer is set to 0 or 1, depending on which minterm of the function is present.

Example: Implement $F(x,y,z) = \Sigma(1,2,6,7)$ using 8-to-1 multiplexer.

Solution: Connect the variables x, y, z to the selection inputs S_2, S_1, and S_0. Then set $I_0 = I_3 = I_4 = I_5 = 0$ and $I_1 = I_2 = I_6 = I_7 = 1$.

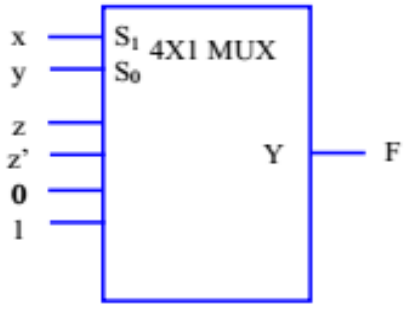
2. Using $2^{(n-1)}$-to-1 multiplexer

We connect (n-1) variables to the selection lines. The multiplexer inputs are going to be either 0 or 1 or the remaining variable or the complement of the remaining variable.

Example: Implement $F(x,y,z) = \Sigma(1,2,6,7)$ using 4-to-1 multiplexer.

Solution: Connect the variables x and y to the selection inputs S_1, and S_0. The inputs of the multiplexers can be obtained from the truth table as shown below.

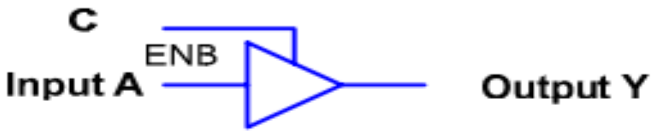
x	y	z	F	
0	0	0	0	F=z
0	0	1	1	
0	1	0	1	F=z'
0	1	1	0	
1	0	0	0	F=0
1	0	1	0	
1	1	0	1	F=1
1	1	1	1	



LECTURE 8: DIGITAL LOGIC CIRCUIT DESIGN

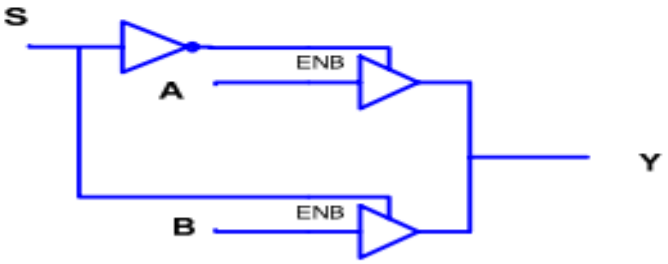
Three State Gates

A three state gate is a digital circuit that exhibits three states. Two states are equivalent to logic 0 and 1. The third state is a high impedance state which is controlled by a control input C. The most commonly used 3-state gate is the buffer.



The output Y = A when C =1 and is high impedance state when C = 0.

It is possible to implement multiplexers using 3-state buffers as shown.



A 4-to-1 multiplexer may be constructed using four 3-state buffers and a 2-to-4 decoder.