

COMBINATIONAL LOGIC CIRCUITS-II

Objective:

To familiarize the design of combinational logic circuits such as decoders, encoders, multiplexers, de-multiplexers and programmable logic devices.

Syllabus:

Design of decoders, encoders, priority encoder, multiplexers, de-multiplexers, higher order de-multiplexers and multiplexers, realization of Boolean functions using decoders, multiplexers, PROM, PAL and PLA.

Outcomes:

At the end of the unit, Students will be able to

- design decoder, encoder and priority encoder, multiplexer and de-multiplexer.
- construct higher order decoders, de-multiplexers and multiplexers.
- implement Boolean functions using decoders, multiplexers and programmable logic devices.

Learning Material

Decoder:

A decoder is a logic circuit that converts an n bit binary input code into M ($=2^n$) output lines such that each output line will be activated for only one of the possible combinations of inputs.

(OR)

A decoder is a Combinational circuit that converts binary information from ' n ' input lines to a maximum of 2^n unique output lines.

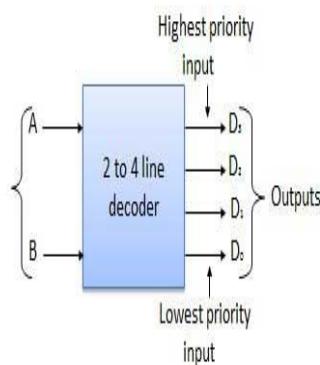
- The decoder is called n -to- m -line decoder, where $m \leq 2^n$.
- A decoder selects one of 2^n outputs by decoding the binary value on the n inputs.
- The decoder generates all of the minterms of the n input variables.
- Exactly one output will be active for each combination of the inputs.
- Decoders are available in two different types of output forms:
 - (1) Active high output type decoders and
 - (2) Active low output type decoders.
- Active high output type of decoders and construct with AND gates and

- Active low output type of decoders and construct with NAND gates.

2 to 4 Line Decoder with active high outputs:

- The block diagram of 2 to 4 line decoder is shown below.
- A and B are the two inputs where D_0 through D_3 are the four outputs.
- Truth table explains the operations of a decoder. It shows that each output is 1 for only a specific combination of inputs.

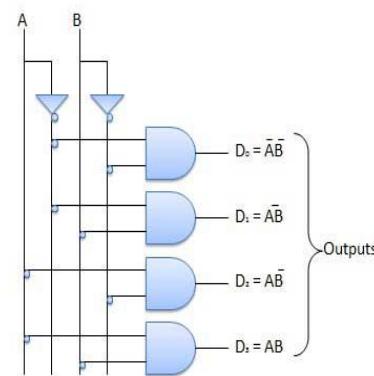
Block diagram



Truth Table

Inputs		Output			
A	B	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

Logic Diagram



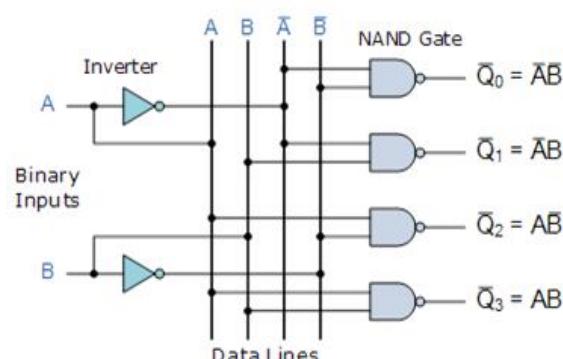
2-to-4 Line Binary Decoder with active low outputs:

- Active low output type of decoders will give the output low for given input combination and all other outputs are high.

Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Logic Diagram



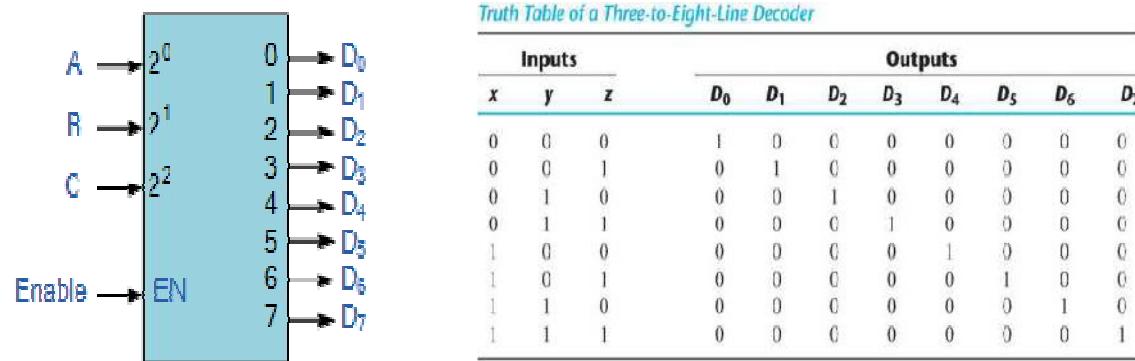
- Some decoders are constructed with NAND gates, it becomes more economical to generate the decoder minterms in their complemented form.
- As a NAND gate produces the AND operation with an inverted output, the NAND decoder looks like this with its inverted truth table.

- As indicated by the truth table, only one output can be equal to 0 at any given time, all other outputs are equal to 1.

3-to-8 line decoder:

- In 3 to 8 line decoder, for each possible input combination, there are seven outputs that are equal to 0 and only one that is equal to 1.
- 3 to 8 line decoder is also called Binary-to-Octal decoder or converter.
- It is also called 1of 8 decoder, because only one of the 8 outputs is active at a time.
- Decoders are widely used in the memory system of computer, where they respond to the address code input from the CPU to activate the memory storage location specified by the address code.
- Decoders are also used to convert binary data to a form suitable for displaying on decimal read outs.
- Decoders can be used to implement combinational circuits, Boolean functions etc.

Block Diagram



Logic Diagram

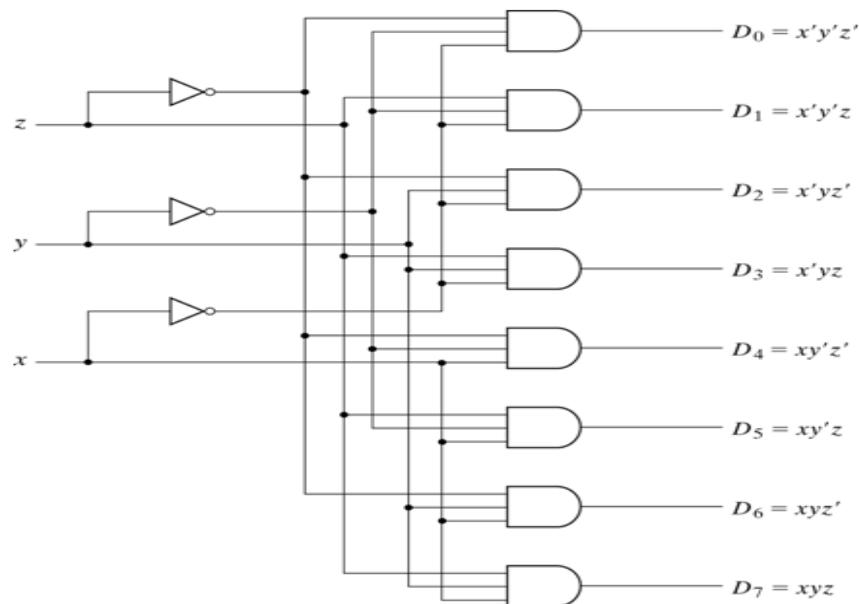
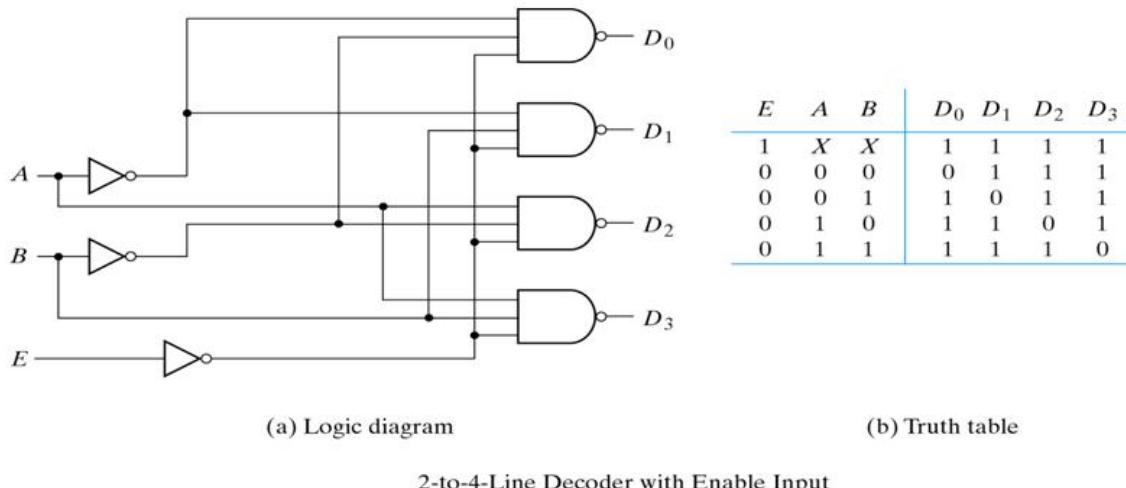


Fig. 3-to-8-Line Decoder

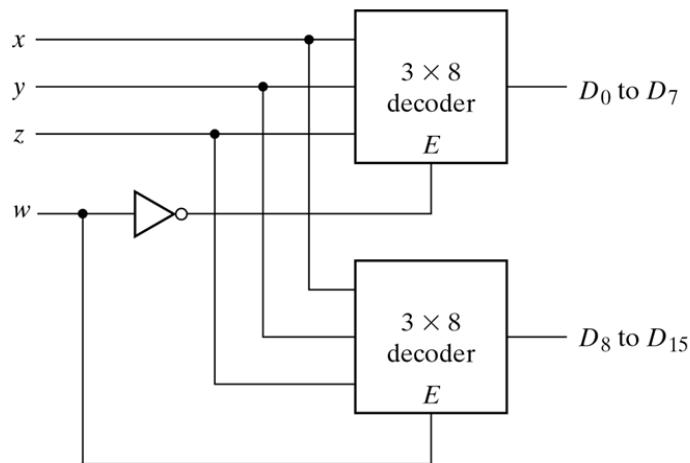
2 to 4 line Decoder with active low enable input

- An enable input can be added to control the operation
 - E=1: disabled
 - None of the outputs are equal to 0
- ❖ Decoders with enable inputs are useful for constructing the higher order decoders.
- ❖ The below circuit operates with complemented outputs and a complemented enable input.
- ❖ The decoder is enabled when E=0 (i.e active low enable)
- ❖ As indicated by the truth table, only one output can be equal to 0 and all other outputs are equal to 1 at any given time.
- ❖ The output whose value is equal to 0 represent the minterm selected by the inputs A and B.
- ❖ The circuit is disabled when E is equal to 1 regardless of the other two inputs.
- ❖ When the circuit is disabled, none of the outputs are equal to 0 and none of the minterms are selected.



Designing a 4-to-16 Decoder using 3 to 8 line decoders

- ❖ Decoders with enable inputs can be connected together to form a larger decoder.
 - ❖ Two 3 to 8 lone decoders with enable inputs connected to form a 4 to 16 line decoder.
 - ❖ When $w=0$, the top decoder is enabled and the other is disabled.
 - ❖ The bottom decoder outputs are all 0's and the top eight outputs generate minterms 0000 to 0111.
 - ❖ When $w=1$, the enable conditions are reversed. The bottom decoder outputs generate minterms 1000 to 1111, while the outputs of the top decoder are all 0's.
 - ❖ In general, enable inputs are convenient feature for standard components to expand their number of inputs and outputs.



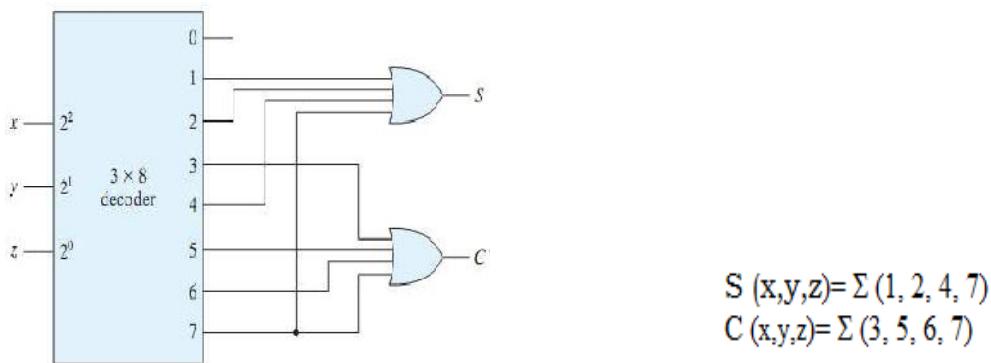
4 × 16 Decoder Constructed with Two 3 × 8 Decoders

Combinational Logic Implementation

- A decoder provides the 2^n minterms of n input variables.
 - They can be used to form any combinational circuits with extra OR gates (sum of minterms)
- A function having a list of k minterms can be expressed in its complemented form F' with $2^n - k$ minterms
 - If $k > 2^n / 2$, F' will have fewer minterms (fewer OR gates)
 - NOR gates are used instead for implementing F'

Example:

Implement full adder circuit whose outputs are given below with a suitable decoder and external gates.

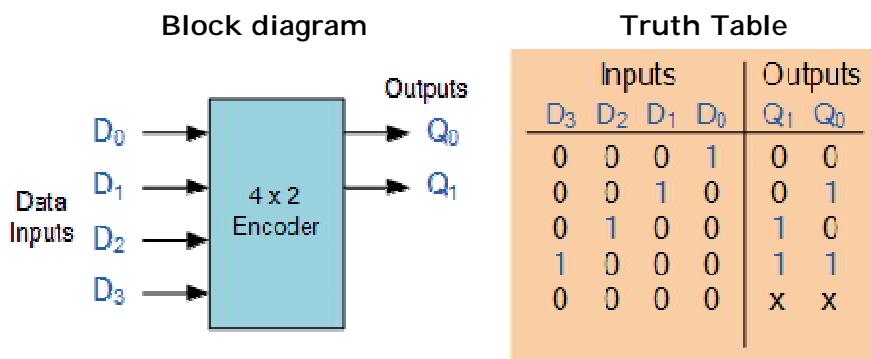


Encoder:

- Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder.
- An encoder has 2^n number of input lines and n number of output lines.

- The output lines generate the binary code of the input positions.
- Only one input can be active at any given time.
- An extra output may be required to distinguish the cases that $D_0 = 1$ and all inputs are 0.

4 to 2 line encoder



Octal to binary encoder:

- Octal-to-Binary take 8 inputs and provides 3 outputs, thus doing the opposite of what the 3-to-8 decoder does.
- At any time, only one input line has a value of 1.
- When all inputs are 0's the outputs will be 0's. The Zero output can also be generated when $I_0 = 1$.
- The figure below shows the truth table of an Octal-to-binary encoder.

Truth Table										Logic Diagram											
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Y_2	Y_1	Y_0
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	0	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	1	—	—	—	—	—	—	—	—	—	—	—
0	0	0	0	0	0	0	1	1	1	1	—	—	—	—	—	—	—	—	—	—	—

The logic diagram shows three AND gates. The first AND gate has inputs I_0 , I_1 , and I_2 . The second AND gate has inputs I_3 , I_4 , and I_5 . The third AND gate has inputs I_6 , I_7 , and I_0 . The outputs of these three AND gates are labeled Y_0 , Y_1 , and Y_2 respectively.

- For an 8-to-3 binary encoder with inputs I_0-I_7 the logic expressions of the outputs Y_0-Y_2 are:

$$Y_0 = I_1 + I_3 + I_5 + I_7$$

$$Y_1 = I_2 + I_3 + I_6 + I_7$$

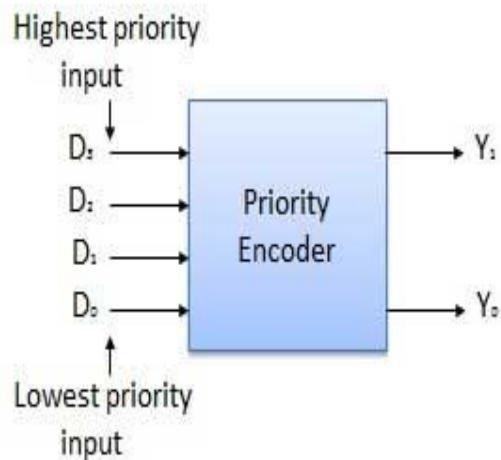
$$Y_2 = I_4 + I_5 + I_6 + I_7$$

- Issues of Encoder
 - What if more than one input is active?
 - What if no inputs are active?

Priority Encoder:

- This is a special type of encoder. Priority is given to the input lines.
- If two or more input lines are 1 at the same time, then the input line with highest priority will be considered.
- There are four inputs D_0, D_1, D_2, D_3 and two outputs Y_0, Y_1 . Out of the four inputs D_3 has the highest priority and D_0 has the lowest priority.
- That means if $D_3 = 1$ then $Y_1 Y_0 = 11$ irrespective of the other inputs. Similarly if $D_3 = 0$ and $D_2 = 1$ then $Y_1 Y_0 = 10$ irrespective of the other inputs.

Block diagram

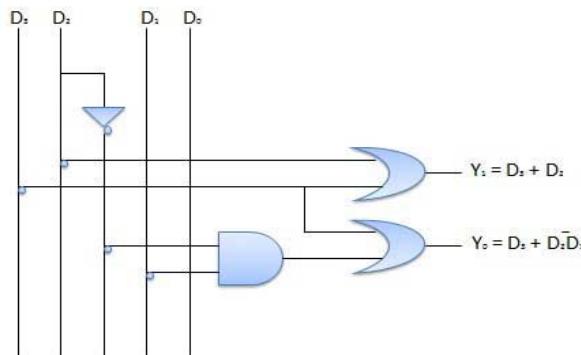
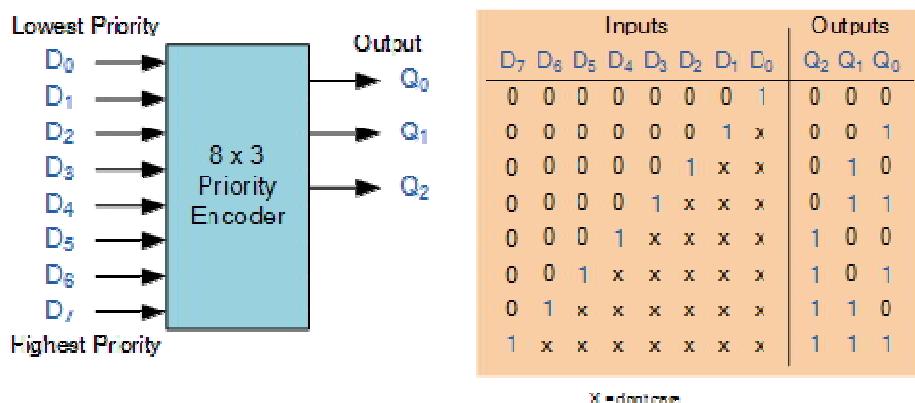


Truth Table

Highest	Inputs			Lowest	Outputs	
D ₃	D ₂	D ₁	D ₀		Y ₀	Y ₁
0	0	0	0	0	x	x
0	0	0	1	1	0	0
0	0	1	x	x	0	1
0	1	x	x	x	1	0
1	x	x	x	x	1	1

Logic Circuit

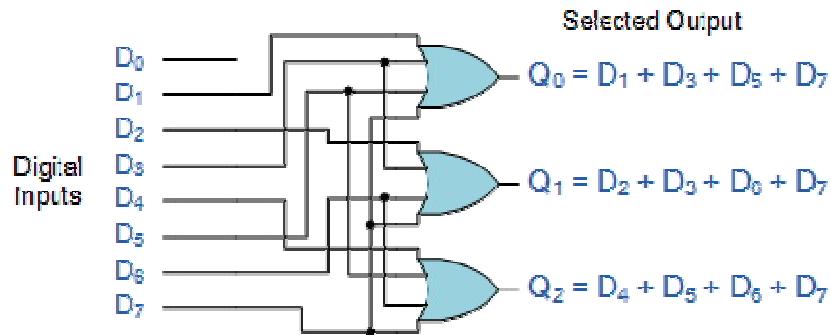
- We have obtained the expression of Y₀ and Y₁ by K- Map minimization

8-to-3 Bit Priority Encoder:

The truth table for a 8-to-3 bit priority encoder is given as:

Where X equals "dont care", that is logic "0" or a logic "1".

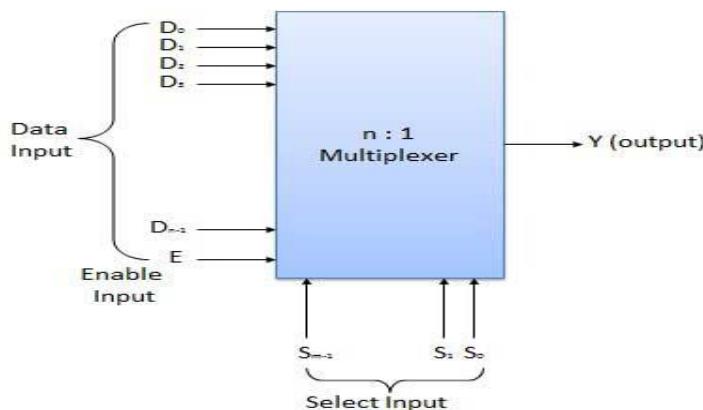
Digital Encoder using Logic Gates



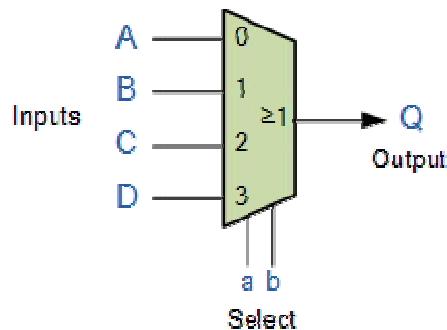
Multiplexers

- Multiplexer is a special type of combinational circuit.
- There are n -data inputs, one output and m select inputs with $2^m = n$.
- It is a digital circuit which selects one of the n data inputs and routes it to the output.
- The selection of one of the n inputs is done by the selected inputs. Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y .
- E is called the strobe or enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low.

Block diagram



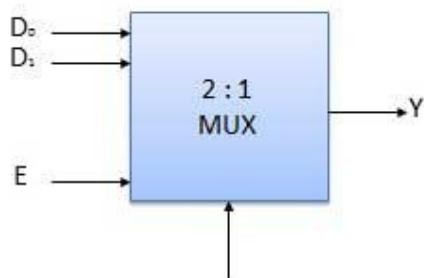
Multiplexer Symbol



Multiplexers come in multiple variations

- 2 : 1 multiplexer
- 4 : 1 multiplexer
- 16 : 1 multiplexer
- 32 : 1 multiplexer

Block Diagram

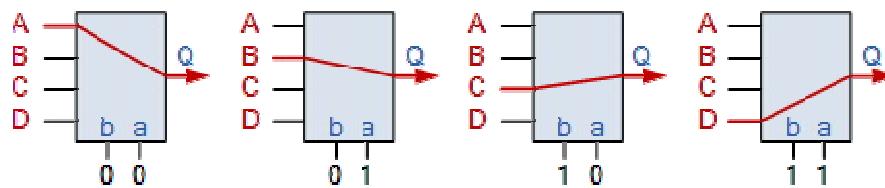


Truth Table

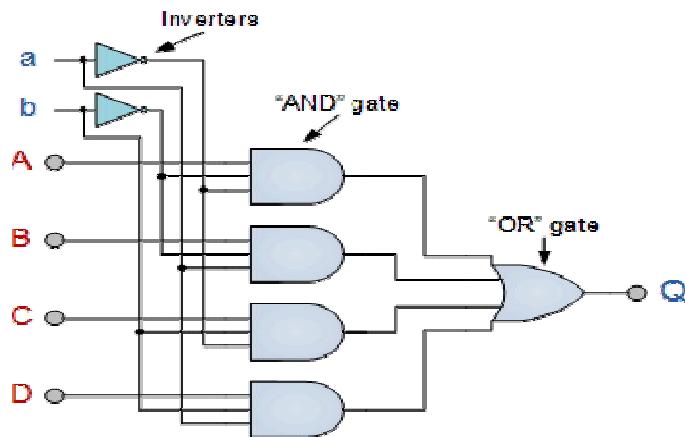
Enable	Select	Output
E	S	Y
0	x	0
1	0	D_0
1	1	D_1

x = Don't care

Multiplexer Input Line Selection



4 Channel Multiplexer using Logic Gates



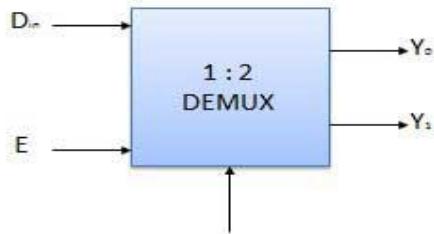
De-multiplexers

- De-multiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs.
- It has only one input, n outputs, m selection inputs.
- At a time only one output line is selected by the selection lines and the input is transmitted to the selected output line.
- A de-multiplexer is equivalent to a single pole multiple way switch.

Demultiplexers comes in multiple variations.

- 1 : 2 Demultiplexer
- 1 : 4 Demultiplexer
- 1 : 16 Demultiplexer
- 1 : 32 Demultiplexer

Block diagram

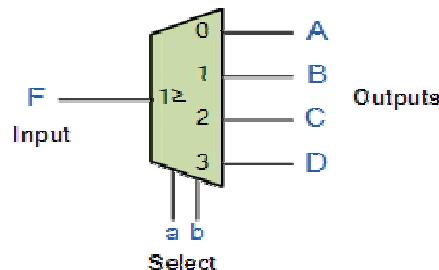


Truth Table

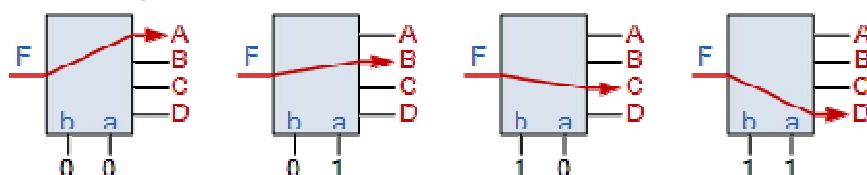
Enable	Select	Output
E	S	Y ₀ Y ₁
0	x	0 0
1	0	0 D _{in}
1	1	D _{in} 0

x = Don't care

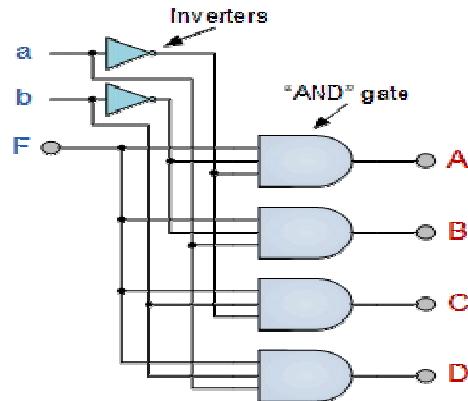
The Demultiplexer Symbol



Demultiplexer Output Line Selection



4 Channel De-multiplexer using Logic Gates

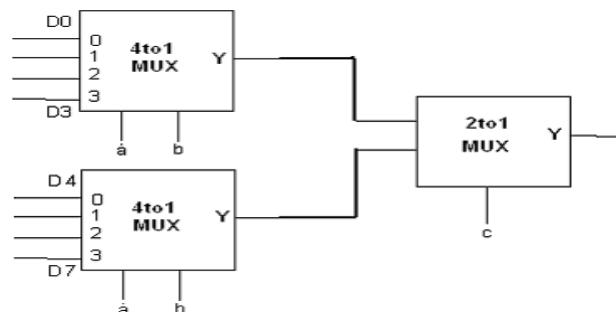


HIGHER MUX from LOWER MUX

Q- Implement (a) 8 to 1 MUX (b) 16 to 1 MUX using 4 to 1 MUX.

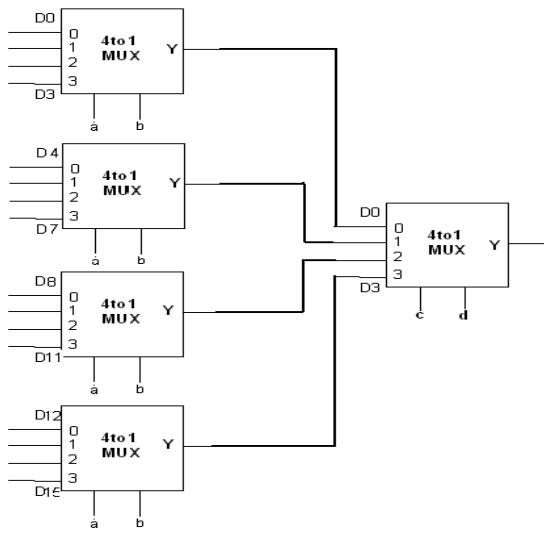
(a) Selection lines are abc_2

Following is the 8 to 1 multiplexer from 4 to 1 multiplexers.



(b) Selection lines are $abcd_2$

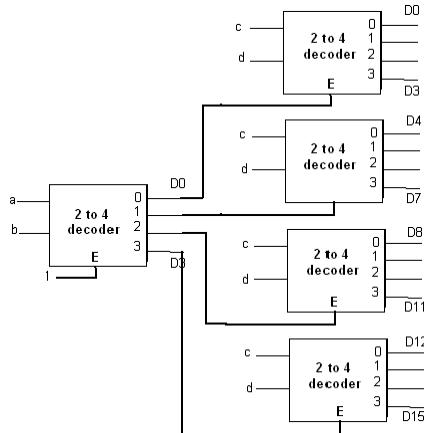
Following is the circuit for **16 to 1 MUX**



Higher Decoder from Lower Decoders

- **Obtain a 4 to 16 decoder using 2 to 4 decoder :**

we take $abcd_2$ as the input to the decoder. Following is the diagram to design 4 to 16 decoder using 2 to 4 decoders.

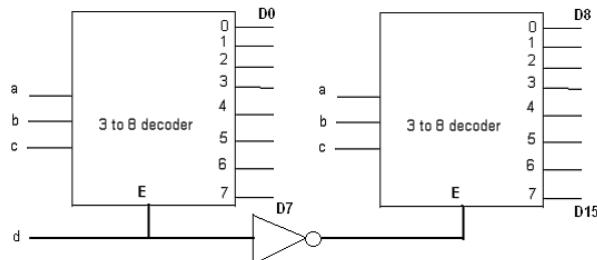


- When we have $a=0$ $b=0$ then top most decoder is enabled and 1 is placed on the output line out of 0 to 3 based on the value of cd_2
- When we have $a=0$ $b=1$ then 2nd decoder from top is enabled and 1 is placed on the output line out of 4 to 7 based on the value of cd_2
- When we have $a=1$ $b=0$ then 3rd decoder is enabled and 1 is placed on the output line out of 8 to 11 based on the value of cd_2
- When we have $a=1$ $b=1$ then bottom most decoder is enabled and 1 is placed on the output line out of 12 to 15 based on the value of cd_2

- Hence top 4 outputs generate min terms 0000 to 0011, next 4 generates min terms 0100 to 0111, next generates 1000 to 1011 and the last 4 outputs generate min terms 1100 to 1111.

Obtain 4 to 16 decoder using 3 to 8 decoders

Here first 8 outputs generate min terms 0000 to 0111 while next 8 generate 1000 to 1111.



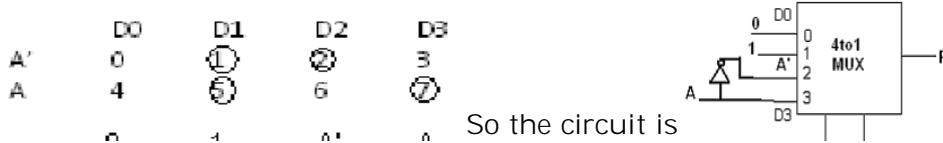
BOOLEAN FUNCTION IMPLEMENTATION USING MUX

- Take one variable for input lines and rest of the term for selection lines.
- Then list the min terms with the variable selected in complimented form in 1st row.
- List the min terms with variable selected in un-complimented form in 2nd row.
- Then encircle the min terms which are present in the function.
 - If we have no circled variable in the column, then we put 0 on the corresponding line.
 - If we have both circled variables, then we put 1 on the line.
 - If bottom variable is circled and top is not circled, apply A to input line.
 - If bottom variable is not circled and top is circled, apply A' to input line.

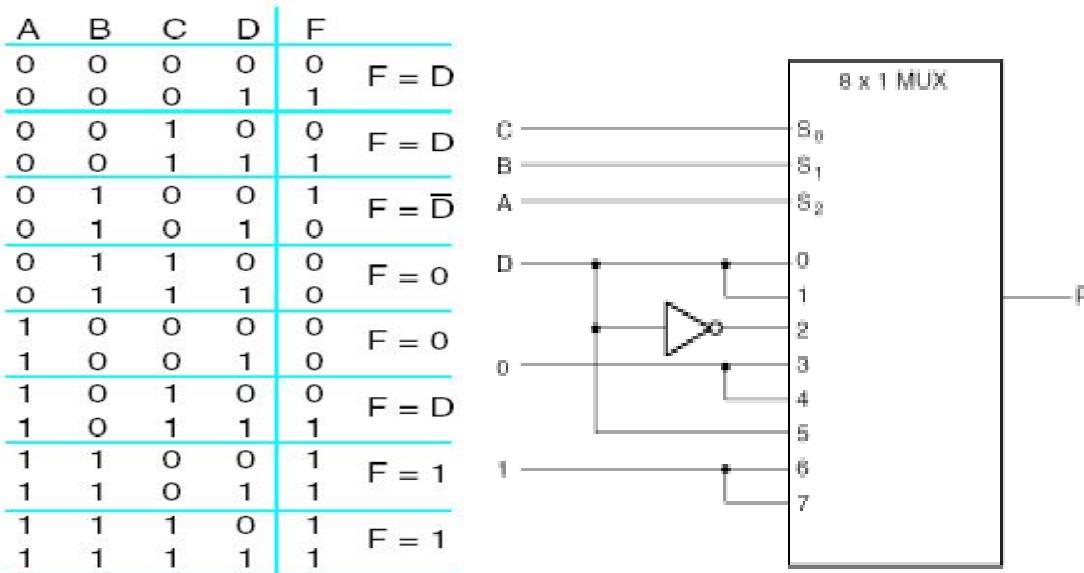
Eg. To implement the function $F(A, B, C) = \Sigma (1, 2, 5, 7)$ using MUX.

Let's take the variable A for input lines and B & C for selection lines.

So we list the min terms as follow:



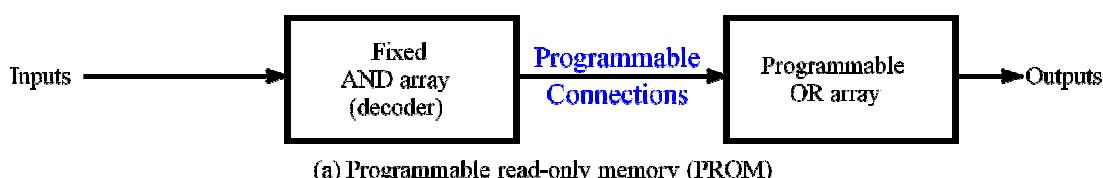
- Consider the function $F(A,B,C,D) = \sum(1,3,4,11,12,13,14,15)$ implement with an 8-to-1 line MUX.



Contents beyond Syllabus

PROM - Programmable Read Only Memory

The first PLD is PROM was introduced in 1970. PROM was introduced for computer memories in which program instructions and constant data values are stored. PROM have fixed AND plane and programmable OR plane. PROM can be used to program any combinational logics with limited number of inputs and outputs.

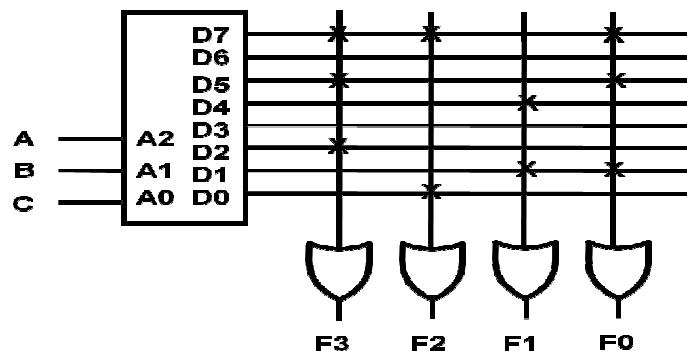


- Programmable Read Only Memories (PROM) have:

- N input lines,
- M output lines, and
- 2^N decoded minterms.
- Fixed AND array with 2^N outputs implementing all N-literal minterms.
- Programmable OR Array with M outputs lines to form upto M sum of minterm expressions.
- A program for a ROM or PROM is simply a multiple-output truth table.
 - If a 1, a connection is made to the corresponding minterm for the corresponding output.
 - If a 0, no connection is made.

Example:

- An 8 X 4 ROM (N = 3 input lines, M= 4 output lines)
- The fixed "AND" array is a "decoder" with 3 inputs and 8 outputs implementing minterms.
- The programmable "OR" array uses a single line to represent all inputs to an OR gate. An "X" in the array corresponds to attaching the minterm to the OR
- Read Example: For input $(A_2, A_1, A_0) = 011$, output is $(F_3, F_2, F_1, F_0) = 0011$.
- What are functions F_3, F_2, F_1 and F_0 in terms of (A_2, A_1, A_0) ?

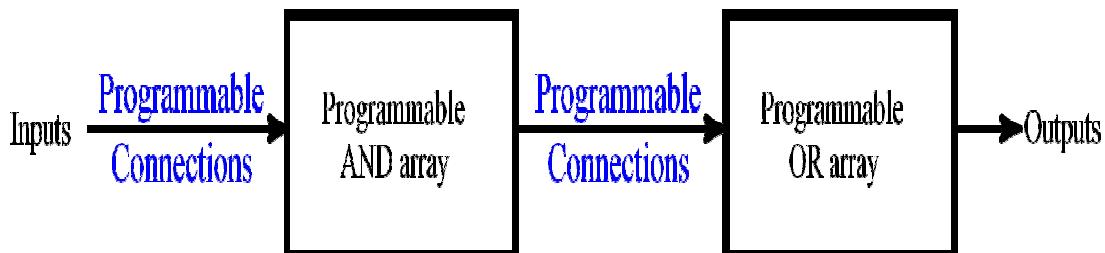


Programmable Logic Array (PLA)

Introduction

- One way to design a combinational logic circuit it to get gates and connect them with wires.
- One disadvantage with this way of designing circuits is its lack of portability.

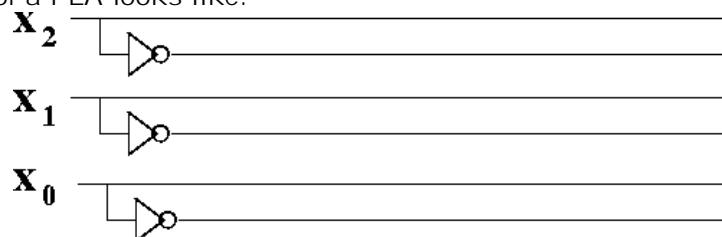
- You can now get chips called PLA (programmable logic arrays) and "program" them to implement Boolean functions. I'll explain what it means to program a PLA.
- Fortunately, a PLA is quite simple to learn, and produces nice neat circuits too.



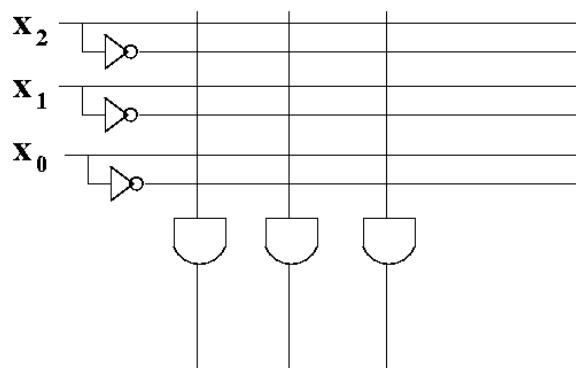
(c) Programmable logic array (PLA) device

Starting Out

The first part of a PLA looks like:



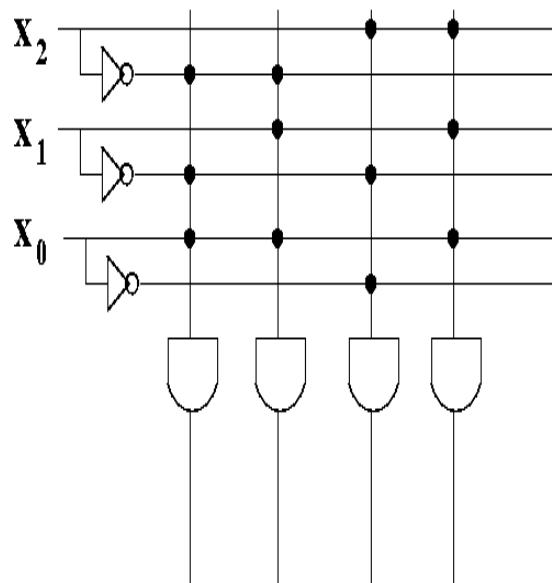
- Each variable is hooked to a wire, and to a wire with a NOT gate. So the top wire is x_2 and the one just below is its negation, \bar{x}_2 .
- Then there's x_1 and just below it, its negation, \bar{x}_1 .
- The next part is to draw a vertical wire with an AND gate. I've drawn 3 of them.



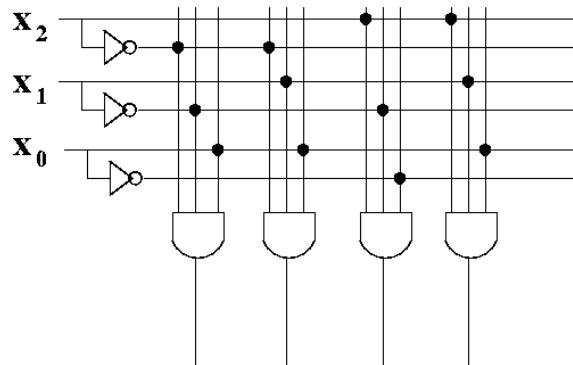
- Let's try to implement a truth table with a PLA.

x_2	x_1	x_0	z_1	z_0
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

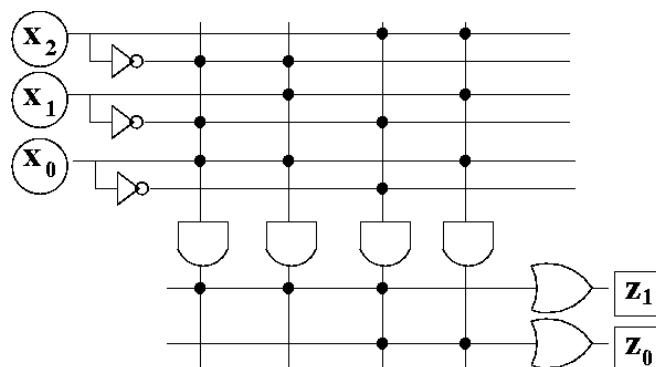
- Each of the vertical lines with an AND gate corresponds to a minterm. For example, the first AND gate (on the left) is the minterm: $x'_2x'_1x_0$.
- The second AND gate (from the left) is the minterm: $x'_2x_1x_0$.
- The third AND gate (from the left) is the minterm: $x'_2x_1x'_0$.
- I've added a fourth AND gate which is the minterm: $x_2x_1x_0$.
- The first three minterms are used to implement z_1 . The third and fourth minterm are used to implement z_0 .
- This is how the PLA looks after we have all four minterms.



- Now you might complain. How is it possible to have a one input AND gate? How can three inputs be hooked to the same wire to an AND gate? Isn't that invalid for combinational logic circuits?
- That's true, it is invalid. However, the diagram is merely a simplification. I've drawn the each of AND gate with three input wires, which is what it is in reality (there is as many input wires as variables).
- For each connection (shown with a black dot), there's really a separate wire. We draw one wire just to make it look neat.

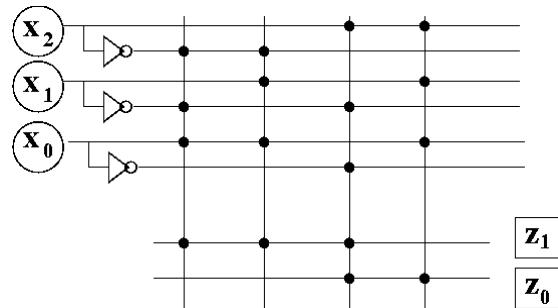


- The vertical wires are called the AND plane. We often leave out the AND gates to make it even easier to draw.
- We then add OR gates using horizontal wires, to connect the minterms together.

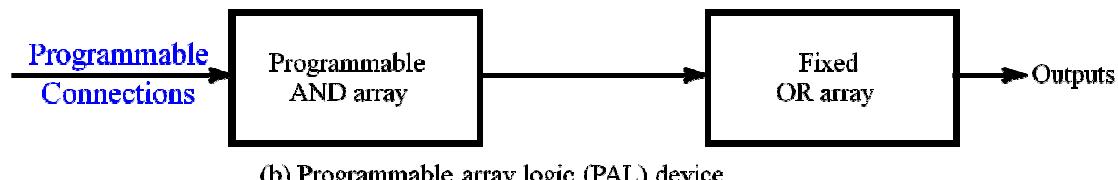


- Again, a single wire into the OR gate is really 4 wires. We use the same simplification to make it easier to read.
- The horizontal wires make up the OR plane.

- This is how the PLA looks when we leave out the AND gates and the OR gates. It's not that the AND gates and OR gates aren't there---they are, but they've been left out to make the PLA even easier to draw.



Programmable Array Logic (PAL)



(b) Programmable array logic (PAL) device

- The PAL is the opposite of the ROM, having a programmable set of ANDs combined with fixed ORs.
- Disadvantage
 - ROM guaranteed to implement any M functions of N inputs. PAL may have too few inputs to the OR gates.
- Advantages
 - For given internal complexity, a PAL can have larger N and M
 - Some PALs have outputs that can be complemented, adding POS functions
 - No multilevel circuit implementations in ROM (without external connections from output to input). PAL has outputs from OR terms as internal inputs to all AND terms, making implementation of multi-level circuits easier.

Example:

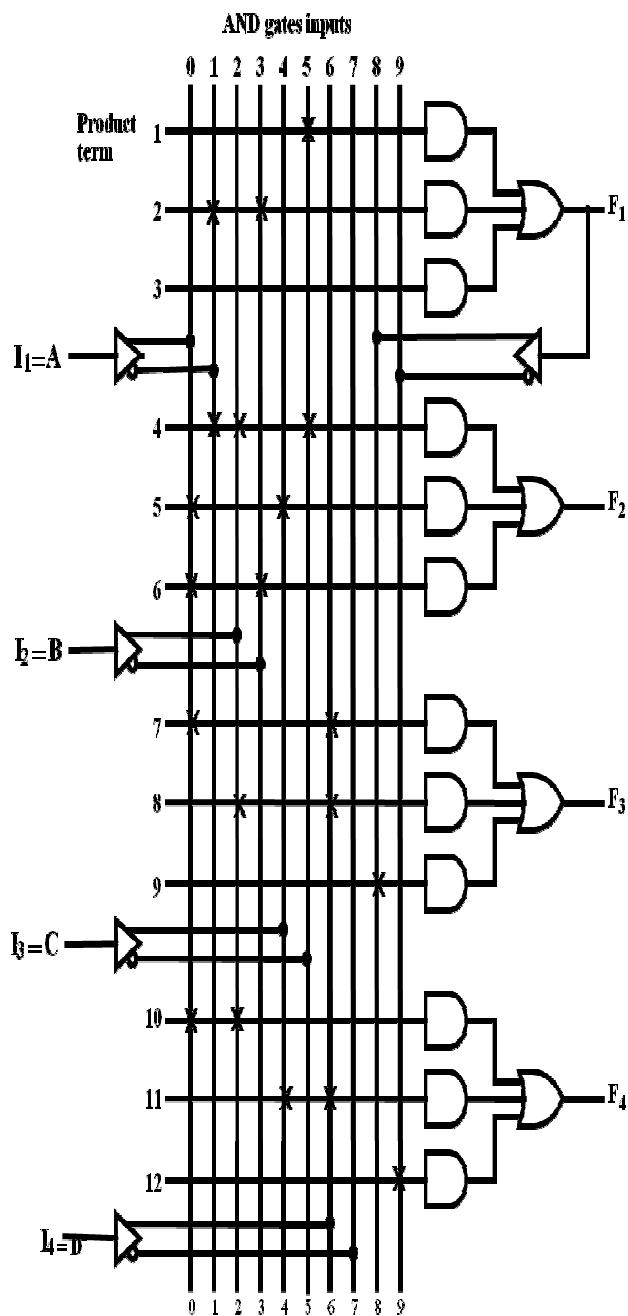
- 4-input, 3-output PAL with fixed, 3-input OR terms
- What are the equations for F1 through F4?

$$F1 = C' + A'B'$$

$$F2 = A'BC' + AC + AB'$$

$$F_3 = AD + BD + F_1'$$

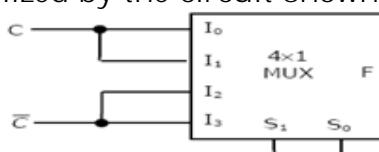
$$F_4 = AB + CD + F_1'$$



UNIT-IV
Assignment-Cum-Tutorial Questions
SECTION-A

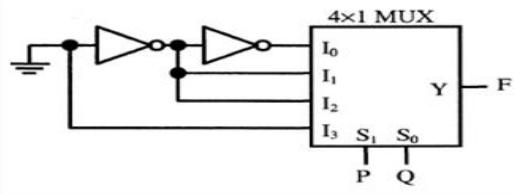
Objective Questions

1. How many 3-to-8-line decoders are required for a 1-of-32 decoder (5 to 32 line decoder)? []
 a) 1 b) 2 c) 4 d) 8
2. How many 1-of-16 decoders are required for decoding a 7-bit binary number?
 a) 5 b) 6 c) 7 d) 8 []
3. A decoder converts _____.
 a) non coded information into coded form.
 b) coded information into non coded form.
 c) HIGH to LOW
 d) LOW to HIGH
4. A combinational circuit which is used to change a BCD number into an equivalent decimal number is []
 a) Decoder b) Encoder c) Multiplexer d) De-multiplexer
5. A encoder with the priority function is called-----.
6. A combinational circuit which is used to change a decimal number into an equivalent BCD number is []
 a) Decoder b) Encoder c) Multiplexer d) De-multiplexer
7. How many data select lines are required for selecting eight inputs? []
 a) 1 b) 2 c) 3 d) 4
8. multiplexer is having _____ output lines.
9. A multiplexer is also known as []
 a) a data accumulator b) a data restorer
 c) a data selector d) a data distributor
10. A de-multiplexer is a device that converts some code into a recognizable number or character. [True/False] []
11. 1-of-16 decoder has 16 outputs and decodes an input of _____ bits. []
 A) two B) three C) four D) six
12. When data input I_6 of a octal-to-binary encoder is active, the data output is []
 A) $A = 0 B = 1 C = 0$ B) $A = 0 B = 0 C = 1$
 C) $A = 1 B = 1 C = 0$ D) $A = 1 B = 0 C = 0$
13. Which device is used in computer hardware to place ALU results into the correct register? []
 A)De-multiplexer B) Encoder C) Multiplexer D)
 Decoder
14. The logic realized by the circuit shown in figure is []



- (a) $F = A \cdot C$
- (b) $F = A + C$
- (c) $F = B \cdot C$
- (d) $F = B + C$

15. Which device is used in computer hardware to select which register is to be gated to the ALU operand inputs? []
 A) De-multiplexer B) Multiplexer C) Encoder D) Decoder
16. Without any additional circuitry, an 8:1 MUX can be used to obtain []
 a. Some but not all Boolean functions of 3 variables
 b. All functions of 3 variables but none of 4 variables
 c. All functions of 3 variables and some but not all of 4 variables
 d. All functions of 4 variables
17. The logic function implemented by the circuit below is (ground implies logic 0) []



- a. $F = \text{AND}(P, Q)$
 b. $F = \text{OR}(P, Q)$
 c. $F = \text{XNOR}(P, Q)$
 d. $F = \text{XOR}(P, Q)$

18. What are the minimum number of 2 to 1 MUX required to generate a 2 input AND gate and a 2 input EX-OR gate? []
 a. 1 and 2 b. 1 and 3 c. 1 and 1 d. 2 and 2

SECTION-B

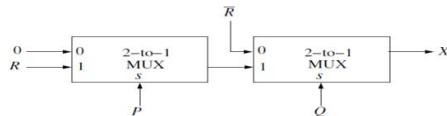
Descriptive Questions

- Differentiate encoder and decoder.
- Design 3-to-8 line Decoder.
- Design 4-to-2 line encoder. Discuss the problems encountered by ENCODER and the ways to rectify them.
- Design a 8-to-3 line Priority Encoder.
- Design 4X1 MUX.
- Design 1X8 De-multiplexer circuit.
- Design BCD to 7 Segment display decoder.
- Implement the following functions using a decoder constructed with AND gates
 $F_1(A, B, C) = \sum(1, 4, 6)$ $F_2(A, B, C) = \sum(3, 5)$ $F_3(A, B, C) = \sum(2, 4, 6, 7)$
- Implement the Boolean function $f(A, B, C, D) = \sum(2, 4, 9, 10)$ with multiplexer having active-HIGH Enable input.
- Implement the following Boolean function using 8:1 MUX.
 $F(w, x, y, z) = \sum_m(0, 3, 5, 8, 9, 10, 12, 14)$
- Design a 1X16 De-multiplexer using two 1X8 De-multiplexers having an active-LOW Enable input.

Section C:

Questions asked in GATE

1. Consider the two cascaded 2-to-1 multiplexers as shown in the figure.



The minimal sum of products form of the output X is

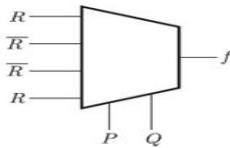
- (A) $P\bar{Q} + PQR$
- (B) $PQ + QR$
- (C) $PQ + \bar{P}\bar{Q}R$
- (D) $\bar{Q}R + PQR$

GATE-CS-2016

2. A RAM chip has a capacity of 1024 words of 8 bits each $1K \times 8$. The number of 2×4 decoders with enable line needed to construct a $16K \times 16$ RAM from $1K \times 8$ RAM is
 (A) 4 (B) 5 (C) 6 (D) 7

GATE-CS-2013

3. The Boolean expression for the output f of the multiplexer shown below is



- (A) $\overline{P \oplus Q \oplus R}$
- (B) $P \oplus Q \oplus R$
- (C) $P + Q + R$
- (D) $\overline{P + Q + R}$

GATE-CS-2010

4. How many 3-to-8 line decoders with an enable input are needed to construct a 6-to-64 line decoder without using any other logic gates?
 (A) 7 (B) 8 (C) 9 (D) 10

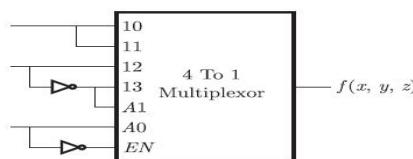
GATE-CS-2007

Suppose only one multiplexer and one inverter are allowed to be used to implement any Boolean function of n variables. What is the minimum size of the multiplexer needed?

- (A) $2n$ line to 1 line
- (B) $2n+1$ line to 1 line
- (C) $2n-1$ line to 1 line
- (D) $2n-2$ line to 1 line

GATE-CS-2007

5. Consider the following multiplexor where 10, 11, 12, 13 are four data input lines selected by two address line combinations $A_1A_0 = 00, 01, 10, 11$ respectively and f is the output of the multiplexor. EN is the Enable input.



The function $f(x, y, z)$ implemented by the above circuit is

- (A) xyz'
- (B) $xy + z$
- (C) $x + y$
- (D) None of the above

GATE-CS-2002