9 Combinational Circuit Design Using MSI (Medium Scale IC)

Introduction

Combinational Circuit Design Using MSI (Medium Scale IC)

e.g.

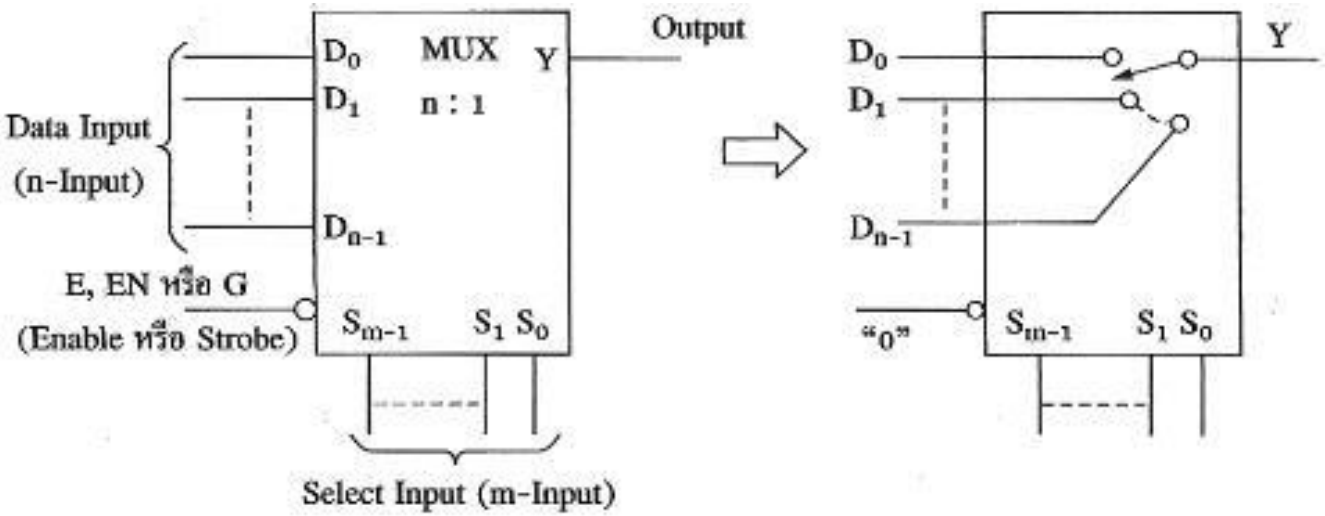
- **Multiplex (MUX),**
- **Decoder (DX),**
- **Read only Memory (ROM)**
- **Programmable Logic Devices (PLD)** Next Week

MULTIPLEXER

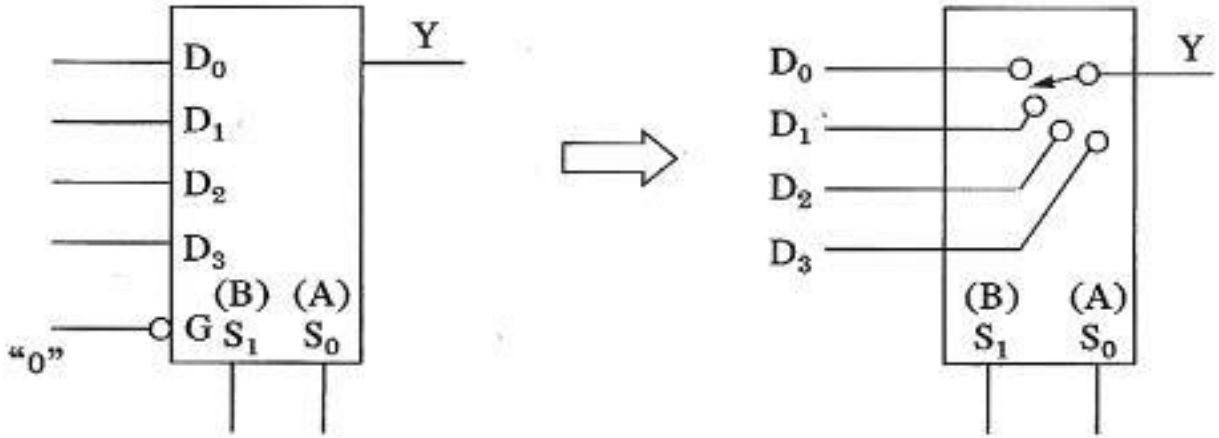
- A multiplexer (MUX) is a device that allows digital information from several sources to be routed onto a single line for transmission over that line to a common destination;
- The basic multiplexer has many data-input lines and a single output line.
- It also has data-selection inputs, which indicates the input channel for data transmission to the output.

MULTIPLEXER

Circuit symbols of n-to-1 Data Selector/Multiplexer (*n*:1 MUX)

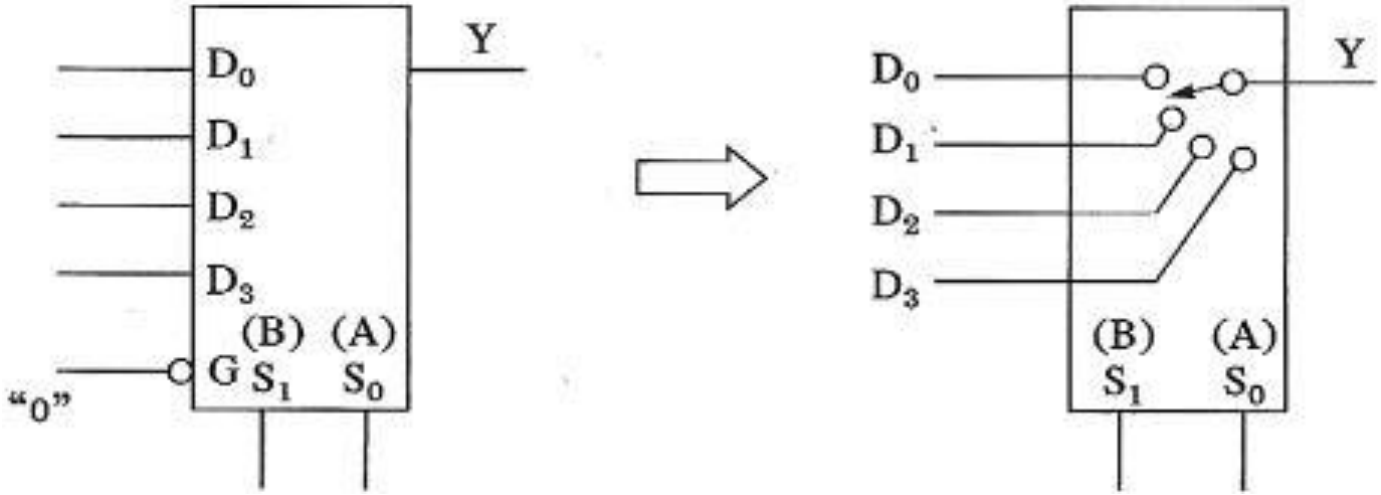


Circuit symbols of 4-to-1 Data Selector/Multiplexer (4:1 MUX)



MULTIPLEXER

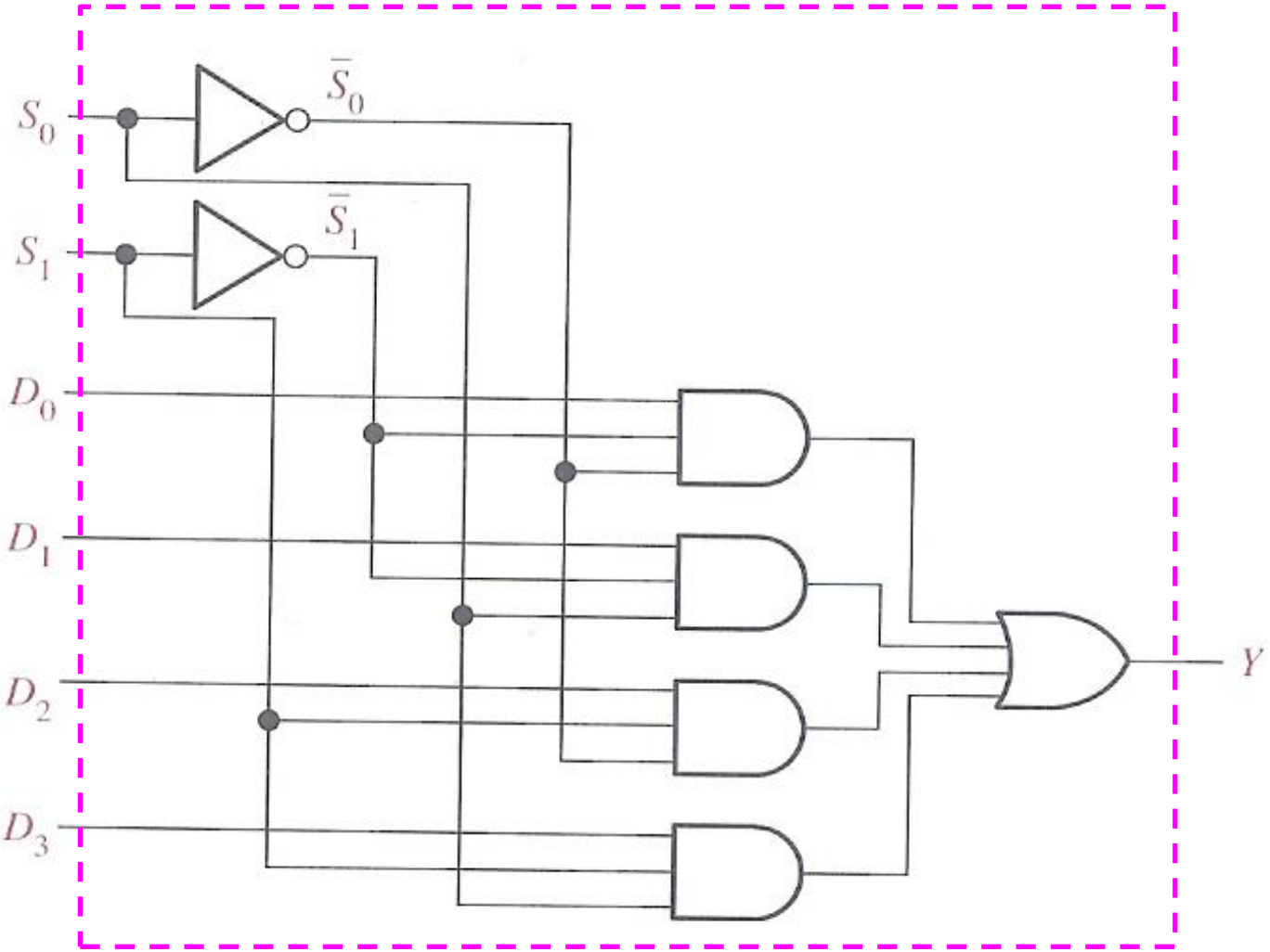
Data Selection of 4-to-1 Multiplexer



DATA-SELECT INPUTS S_1	S_0	INPUT SELECTED
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

MULTIPLEXER

Schematic circuit diagram of the Multiplexer with 4 Inputs



Combinational Circuit Design Using MULTIPLEXER

Combinational Circuit Design Using MULTIPLEXER

- Reduce the complication of the variables in the truth table.
- Reduce the number of logical gates by using MUX IC instead

As in the following example

Combinational Circuit Design Using MULTIPLEXER: Example of 3 Variables

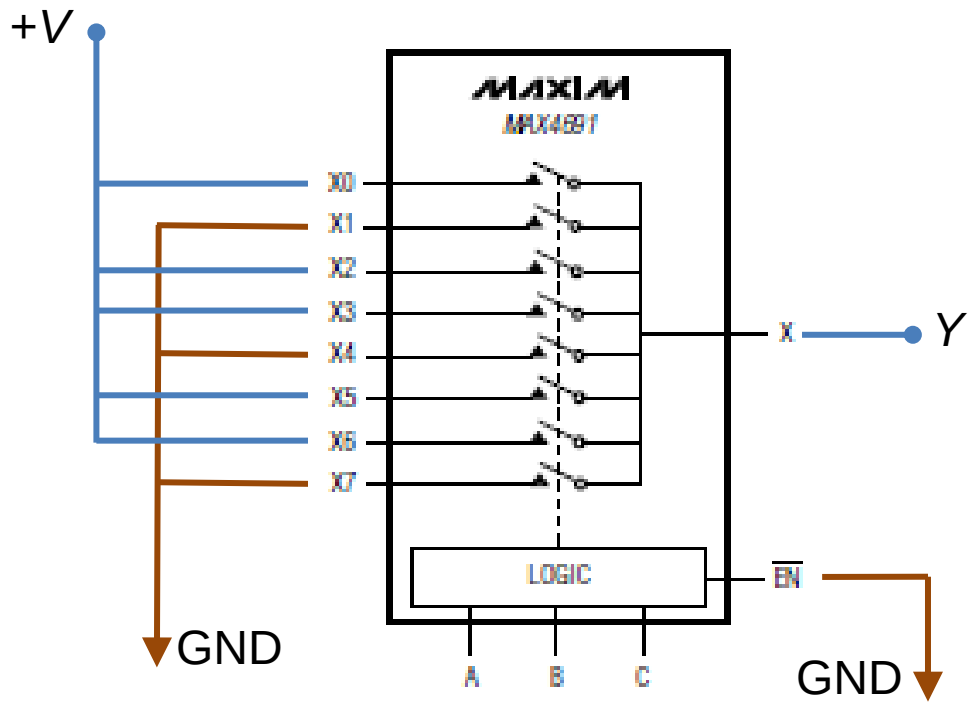
Design a logical logic circuit that provides the outputs as following truth table

ค่าจำนวนฐานสิบ	A	B	C	Output (Y)
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Let's see a MUX IC: MAX4691-MAX4694

Combinational Circuit Design Using MULTIPLEXER: Example of 3 Variables

- Since it is a circuit that has 8 input signals, 8:1 MUX is employed.
- The output signals of 1 occurs at the input signal = 0, 2, 3, 5 and 6;
- The output signals of 0 occurs at the input signal = 1, 4 and 7.
- So, use MULTIPLEXER to replace this mixed logic circuit as follows



Combinational Circuit Design Using MULTIPLEXER: Example of 3 Variables

- In design using MUX
- We can create a larger MUX from a smaller MUX connected in parallel.
- We can replace a larger MUX and replace it with a smaller MUX in conjunction with VEM principles.
- From the previous example, If we set *C* to be the Entered Variable, we get:

$$Y = \sum m(0,2,3,5,6) = \overline{A}\overline{B}\overline{C} + \overline{A}B\left(\overline{C} + C\right) + A\overline{B}C + AB\overline{C} = f(A,B)$$

A \ B	0	1
0	C	1
1	C	C

Combinational Circuit Design Using MULTIPLEXER: Example of 3 Variables

For the requirement of the output:

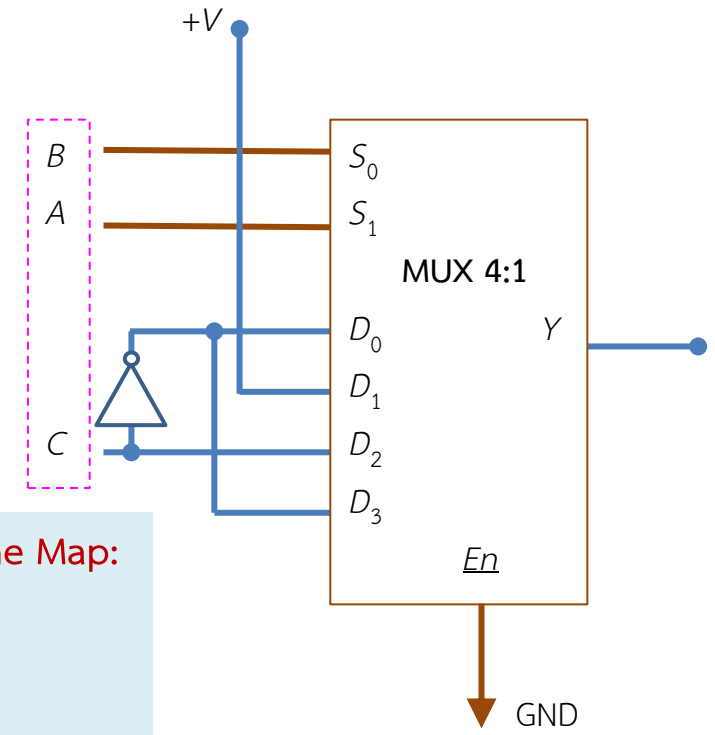
$$Y = \sum m(0,2,3,5,6) = \bar{A}\bar{B}\bar{C} + \bar{A}B\left(\bar{C} + C\right) + A\bar{B}C + AB\bar{C} = f(A,B)$$

We can write the VEM as following table:

B \ A	0	1
0	$\bar{C}$	C
1	$\bar{C}+C$	$\bar{C}$

➔

B \ A	0	1
0	$\bar{C}$	C
1	1	$\bar{C}$



- As the principle of VEM of *N* variables, and *n* entered in the Map:
- The Size of MULTIPLEXER used in the system will be:

  MULTIPLEXER of 2^{N-n} : 1
- Typically, the largest MUX is 16:1 or 2^4:1.
- For the system with the *N* variable, the no. of variables has to put in the map would be $n = N - 4$.

Combinational Circuit Design Using MULTIPLEXER:
Example of 4 Variables

To design a mixed logic circuit whose outputs signal is

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

If we use 8:1 MULTIPLEXER, it can handle only 3 variables.

So, we need to put one variable in to map by marking *D*:

$$Y = \bar{A}\bar{B}\bar{C}\left(\bar{D} + D\right) + A\bar{B}\bar{C}\left(D\right) + \bar{A}B\bar{C}\left(\bar{D}\right) + \bar{A}\bar{B}C\left(\bar{D}\right) = f\left(A, B, C\right)$$

C \ AB	00	01	11	10
0	0	$\bar{D}$	D	1
1	0	0	0	$\bar{D}$

Combinational Circuit Design Using MULTIPLEXER:
Example of 4 Variables

For outputs signal

$$Y = A\bar{B}\bar{C}\left(\bar{D} + D\right) + AB\bar{C}D + A\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} = f\left(A, B, C\right)$$

Then we can write the VEM as:

C \ AB	00	01	11	10
0	0	$\bar{D}$	D	1
1	0	0	0	$\bar{D}$

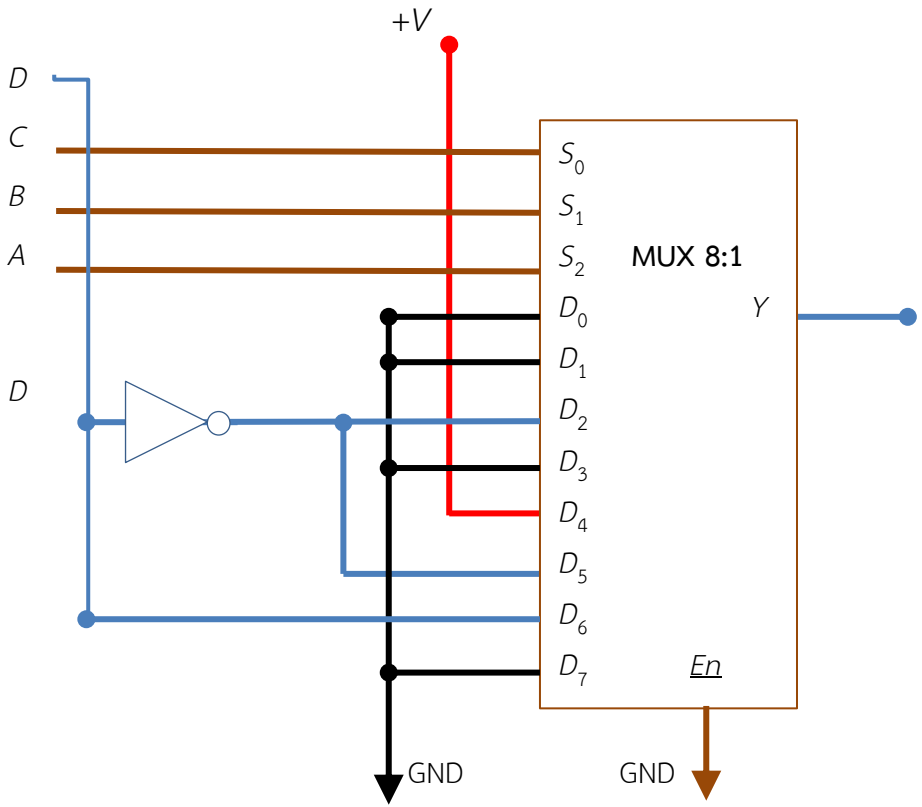
And can get the circuit ➔

0: .....

1: .....

D: .....

$\bar{D}$: .....



Combinational Circuit Design Using MULTIPLEXER: Example of 5 Variables

To design a mixed logic circuit whose output signal is

$$Y = \bar{A}\bar{B}C\bar{D}E + A\bar{B}C\bar{D}\bar{E} + \bar{A}BC\bar{D}\bar{E} + \bar{A}BCDE + A\bar{B}C\bar{D}E + \bar{A}\bar{B}CD\bar{E} + ABCDE$$

If we use 8:1 MULTIPLEXER, it can handle only 3 variables.

So, we need to put TWO variables in to map by marking DE:

$$Y = \bar{A}\bar{B}C\left(\bar{D}E + \bar{D}\bar{E}\right) + A\bar{B}C\left(\bar{D}\bar{E}\right) + \bar{A}BC\left(DE\right) + \bar{A}\bar{B}C\left(\bar{D}\bar{E} + D\bar{E}\right) + ABC\left(DE\right)$$

C \ AB	00	01	11	10
0	DE	$\bar{D}\bar{E}+\bar{D}E$	$\bar{D}\,\bar{E}$	0
1	0	0	$D\,E$	$\bar{D}\bar{E}+D\bar{E}$

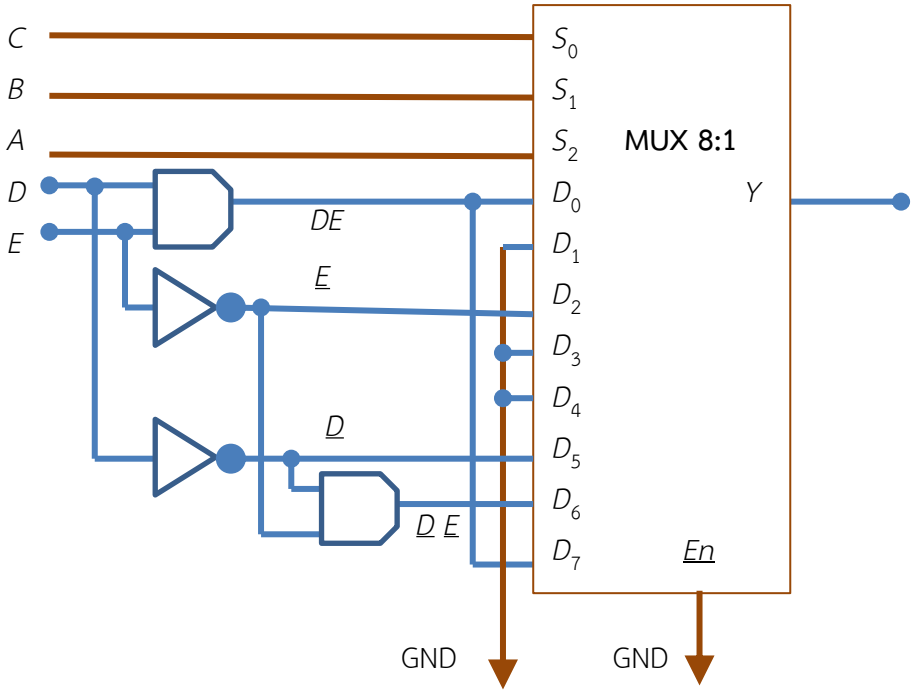
C \ AB	00	01	11	10
0	DE	$\bar{E}$	$\bar{D}E$	0
1	0	0	DE	$\bar{D}$

Combinational Circuit Design Using MULTIPLEXER:
Example of 5 Variables

Then we can write the VEM as:

C \ AB	00	01	11	10
0	DE	$\overline{DE}+DE$	$D\overline{E}$	0
1	0	0	DE	$\overline{DE}+\overline{DE}$

C \ AB	00	01	11	10
0	DE	$\overline{E}$	$\overline{DE}$	0
1	0	0	DE	$\overline{D}$



And can get the circuit

The greater number of variables entered in the map,

The more complication in design of the control circuit.

The Expansion the SIZE of MULTIPLEXER: 16*n* to 1 MULTIPLEXER.

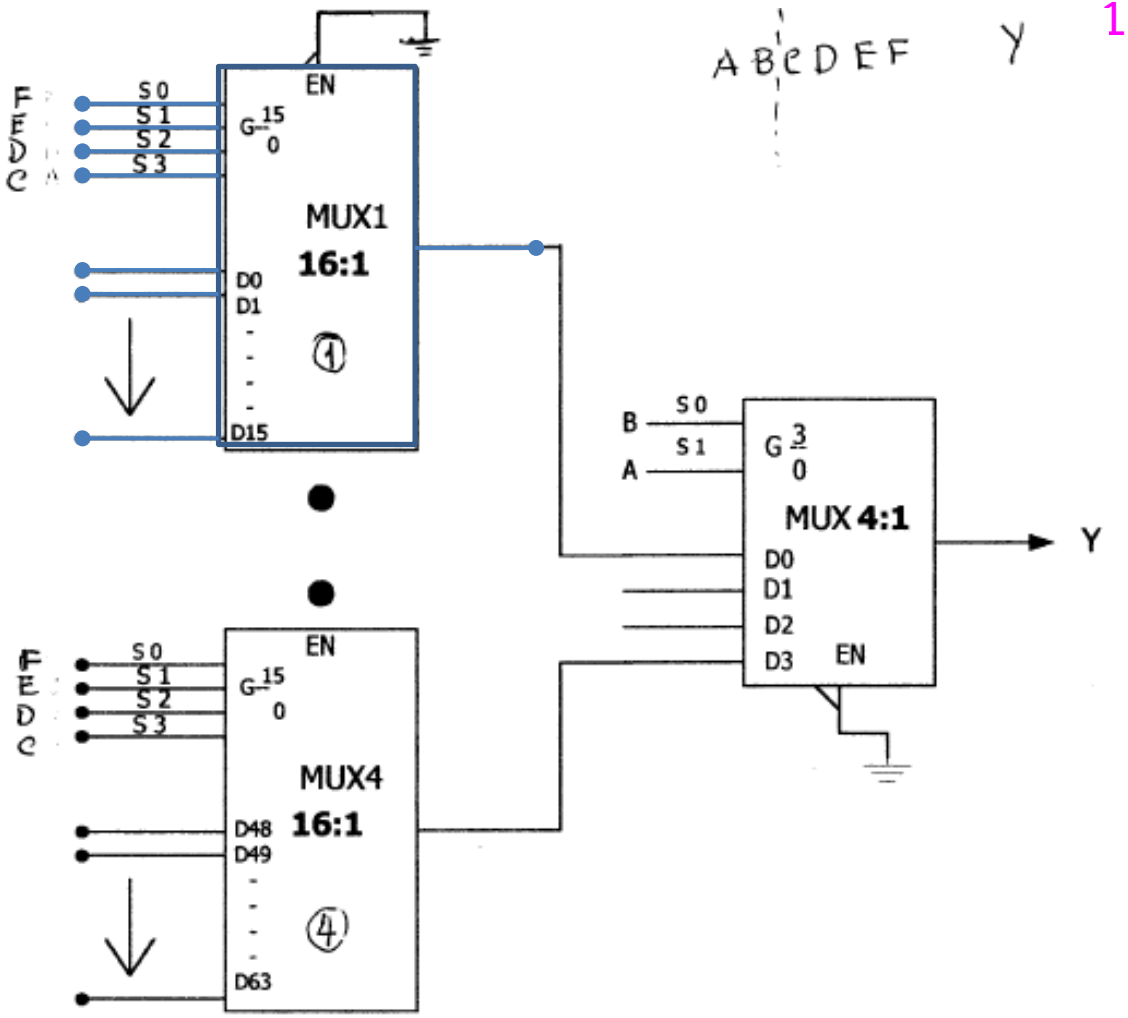
- For more input channels,
- we want to expand the MULTIPLEXER to be a larger size.
- We can do this by gather a number of MULTIPLEXERs together.
- For example, if a system of 64 channels required,
- we have to use four MUX 16: 1 for 64 channel inputs,
- and one MUX 4: 1 to gather them together for ONE output.
- As follows

The Expansion the SIZE of MULTIPLEXER: 16*n* to 1 MULTIPLEXER.

Example of a 64 channels switching system

$35 = 10\ 0011_2$

$12 = 00\ 1100_2$



Combinational Circuit Design Using MULTIPLEXER: In the case of Don't Care.

- Designing a combinational circuit using MULTIPLEXER in the case of Don't Care, it will be more complicated.
- Because we have to consider the Don't care either LOGIC (0/1) or VARIABLE that it would become to.
- However, this would make the design easier since the Don't care could be ANY appropriate signals those make more choices for circuit connection.
- For example, Design a circuit that outputs signals is: $F(A,B,C) = \sum m(0,3,6) + d(5,7)$

$$F(A,B,C) = G(A,B) = \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}\overline{C} + dA\overline{B}C + dABC$$

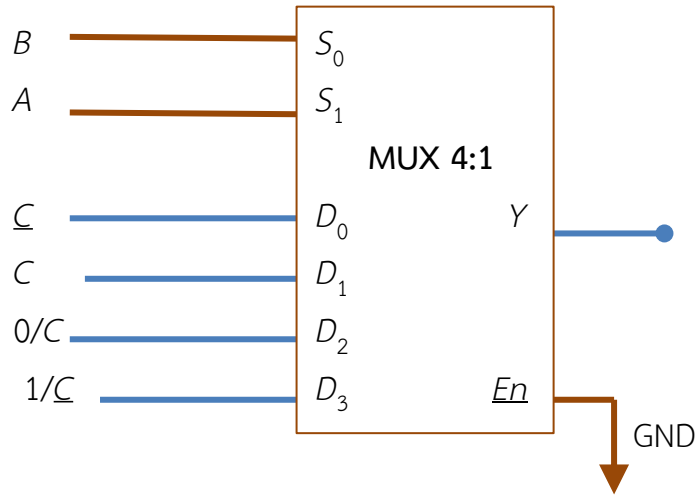
VEM Method: Give $$F(A,B,C) = G(A,B) = \overline{A}\overline{B}\left(\overline{C}\right) + \overline{A}B\left(C\right) + AB\left(\overline{C} + dC\right) + A\overline{B}\left(Cd\right)$$

And can be written as VEM as follows:

B \ A	0	1
0	$\overline{C}$	Cd
1	C	$\overline{C} + Cd$

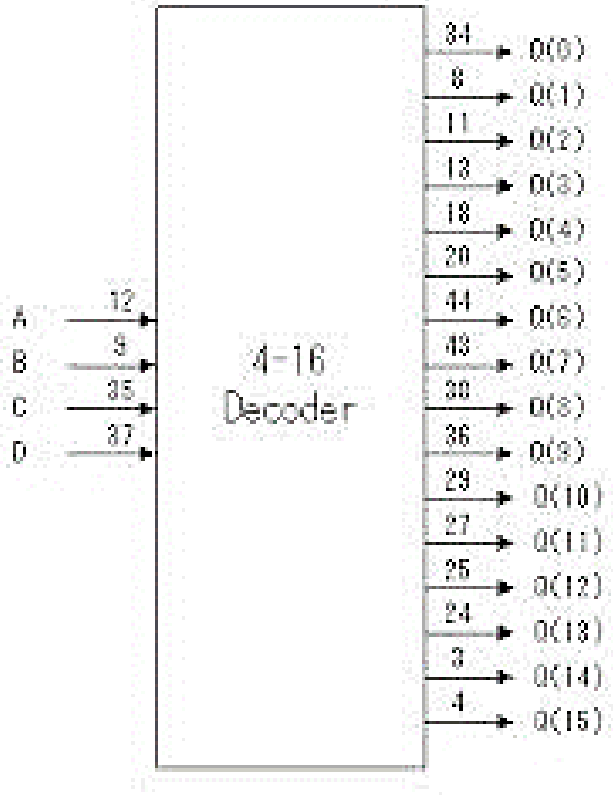
➔

B \ A	0	1
0	$\overline{C}$	$0/C$
1	C	$1/\overline{C}$



Decoders

- The decoder is the circuit that gives the output *n* to be *H*.
- Where *n* is a number corresponding to the input code.
- For example, input = 1101 will allow *H* to come out from output 13.
- Called DX *n*:2^n
- The largest size in the IC format is DX 4:16 (24-pin IC).
- Applications:
- 24 Device (ON/OFF) selection.



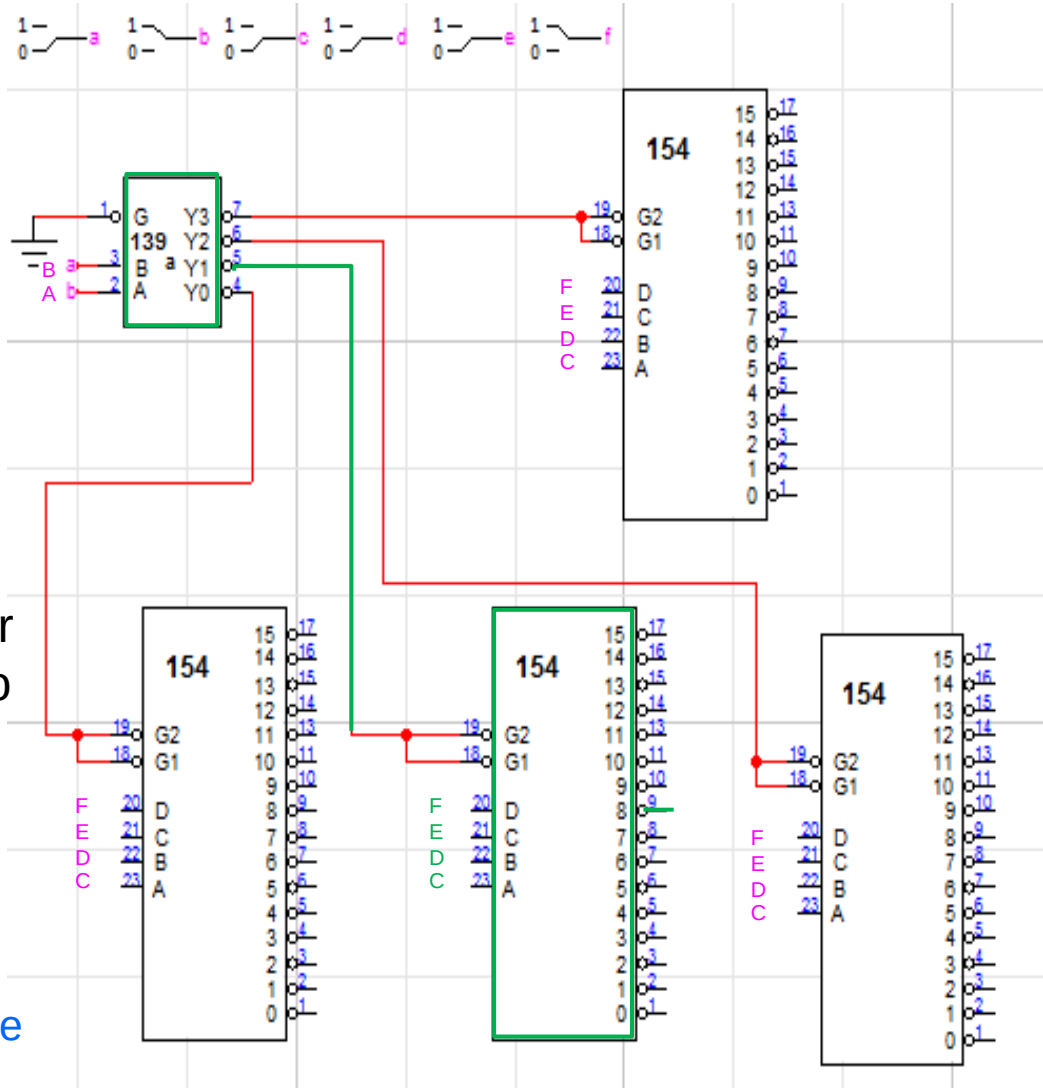
Decoder Expansion

- Decoder Expansion can be done by using *n* 4:16 decoders, which provides16*n* outputs.
- However, these individual decoders are required to selected by another *m*:*n* decoder, where $2m \geq n$.
- For example: A 6:64 decoder
- is made by using FOUR DX 4:16 decoders (4-bit code);
- In conjunction with A DX 2:4 Decoder (2-bit code) to select one from four to activated.
- Example: To decode 25 = 01 1001

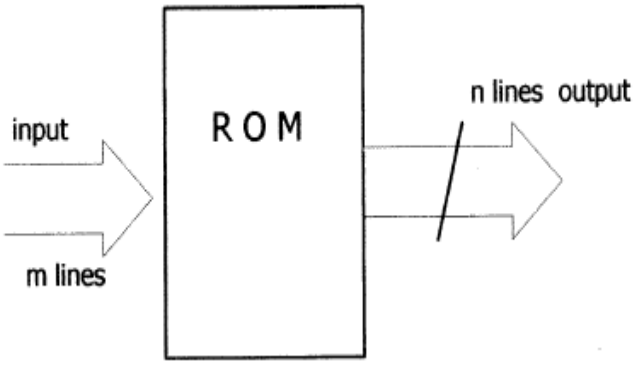
6bit code = AB CDEF

Connect to IC139 IC154

Then the output 9 of the second IC154 will be activated to L.



Read Only Memory: ROM

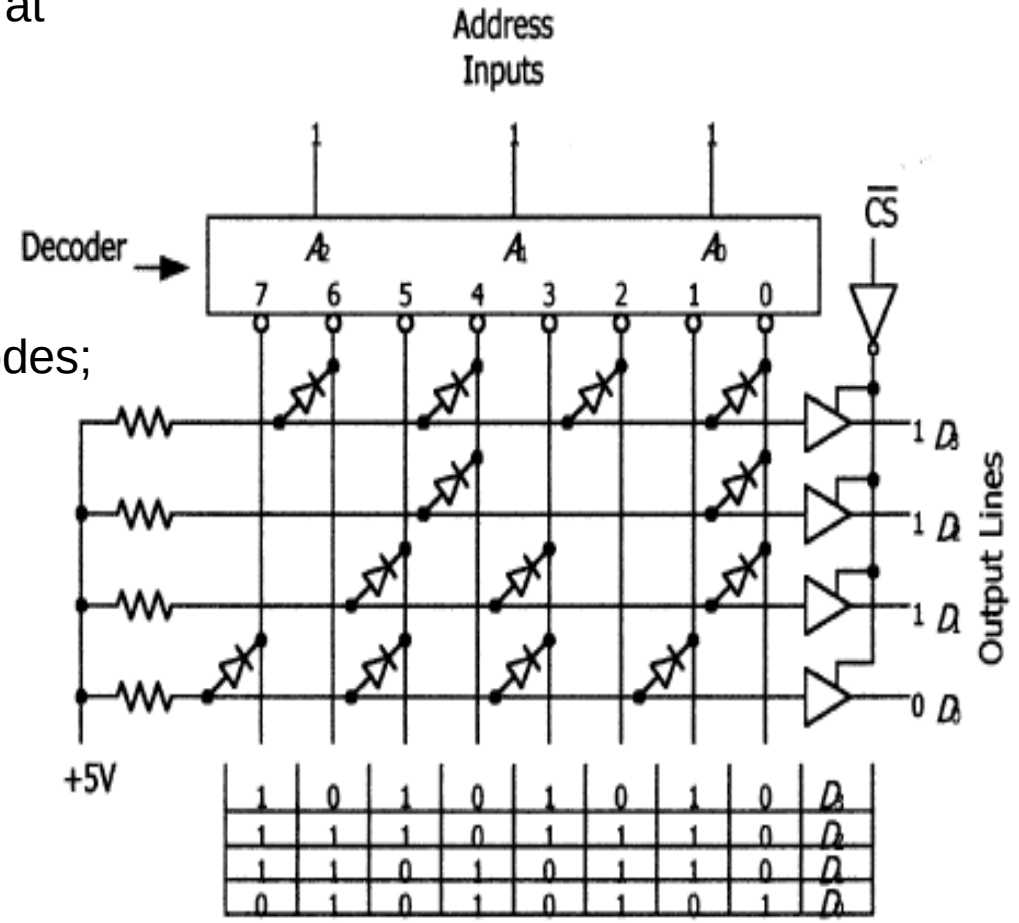


ROM *m*:*n*

- ROM *m*:*n*
- The *m*-line input will point the address 0-2^m - 1;
- The individual address has a *n*-bit data.
- Which is permanent programmed values.
- The easy structure of ROM is in the next page.

The Structure of ROM Size $2^3 \times 4$

- ROM has the structure of the diode array
- Input the address (3bit);
- The activated address will produce 0 at its output (active LOW):
- The data will be read out by:
- With Diode: D = 0;
- Without Diode: D = 1;
- Initially, the ROM would have ALL diodes;
- The ROM can be programmed by exploding the diode(s) in the array.
- Example: Input = 101_2 = 5
- Address 5 = 0;
- Data out = 1100_2
- These data be used to
- Turn ON/OFF 4 devices in the same time;
- Drive the 7-Segment LED;
- Etc.



ROM Implication

- To design a ROM that provides the following signal outputs:

$$Y_1 = F(A,B,C) = \sum m(1,3,4,6) \qquad Y_2 = F(A,B,C) = \sum m(2,4,5,7)$$
$$Y_3 = F(A,B,C) = \sum m(0,1,5,7) \qquad Y_4 = F(A,B,C) = \sum m(1,2,3,4)$$

- There are three variable inputs, which produces a code of values 0 - 7
- Use the corresponding value code to generate the outputs Y_1 - Y_4.