MULTIPLEXERS AND DEMULTIPLEXERS PART - 01

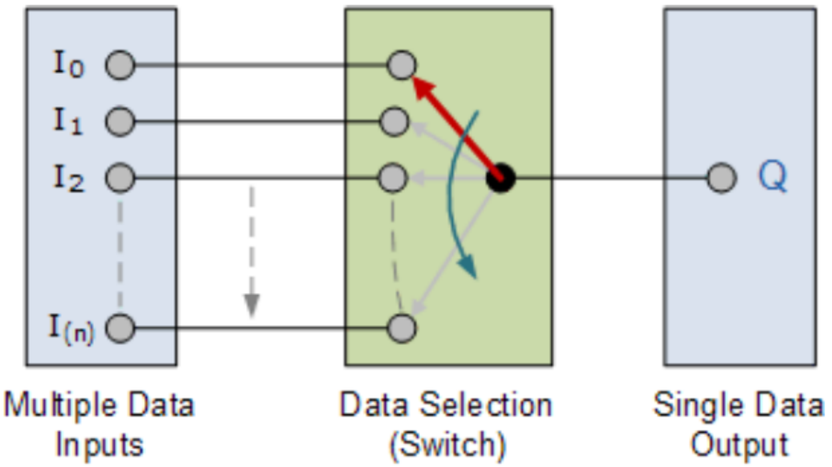
Multiplexer (MUX)

Switch

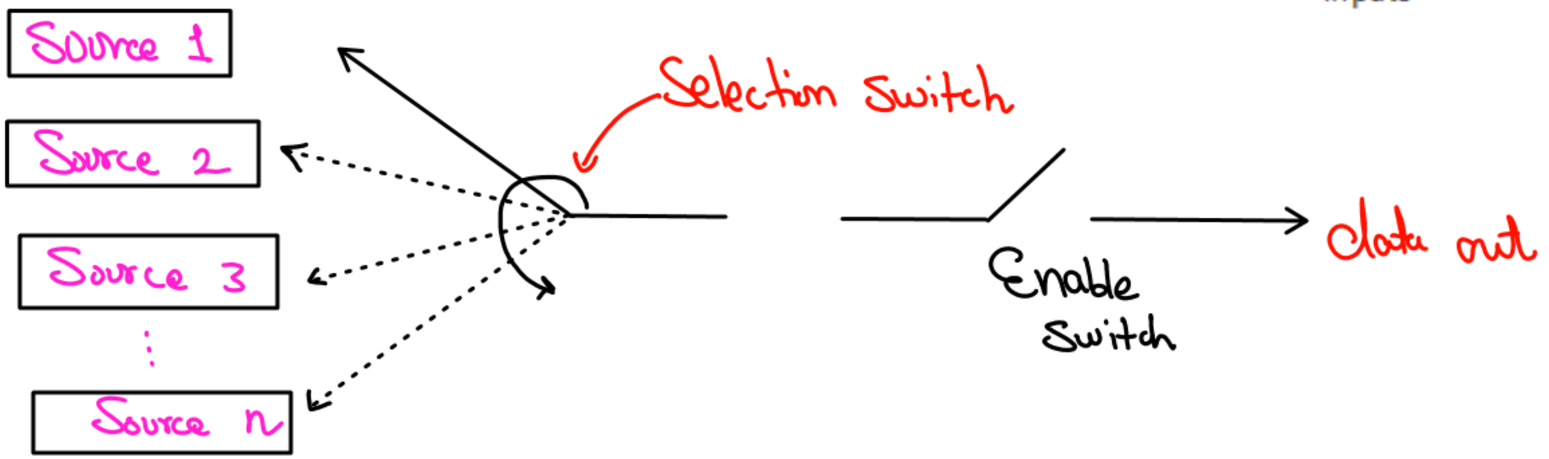
A multiplexer (often abbreviated as MUX) is a digital circuit that selects one of many input signals and forwards the selected input to a single output line.

A multiplexer is a combinational circuit with:

- 2^n input lines
- 1 output line
- n selection (or control) lines



* Selecting over multiple inputs & send to the o/p



Select Line

A select line (also called a control line) in a multiplexer determines which input is connected to the output.

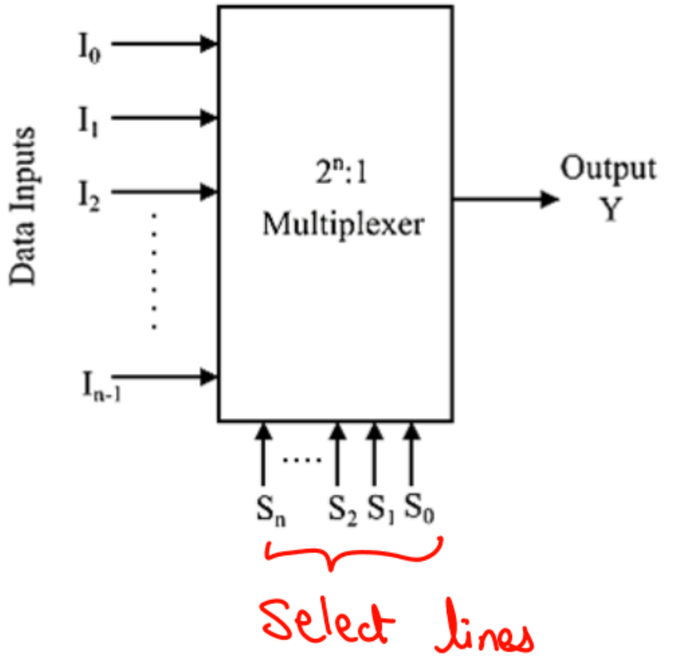
If there are m select lines and n input lines in a MUX then the number of possible input lines will be 2^m

Select Lines (m)	Input Lines ($n = 2^m$)
1	2
2	4
3	8
4	16
5	32
m	2^m

max

* Each Combination of Select line Represents the single input

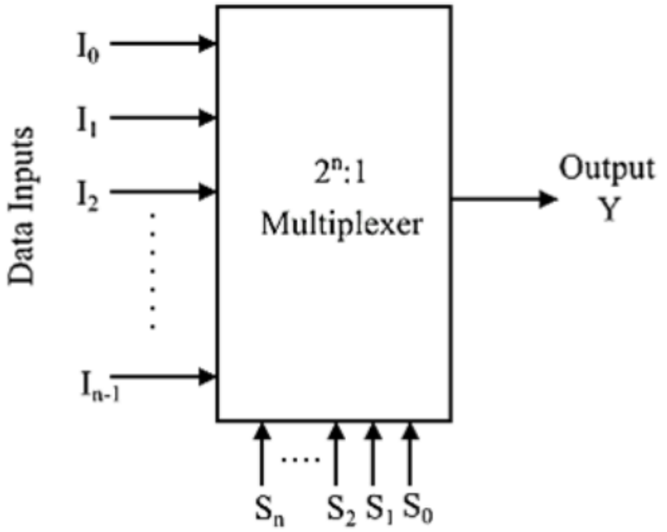
* thats why Mux is also called as a selector



If a multiplexer has N input lines, the number of select lines needed is: $\lceil \log_2 N \rceil$

Select Line

* for every Combination of select line there will be 1 input line associated with that Combination.



* Eg if there are total of 4 input lines then we need $\lceil \log_2(4) \rceil$ select lines

↓

2 select lines.

→ It is called as 4 : 1 Mux or 4 X 1 Mux

Input lines ↙ ↘ output lines

Input lines → Input line must be in power of 2.

→ We can represent a Mux as $2^m : 1$ mux or 2^m X 1 mux

m = select line

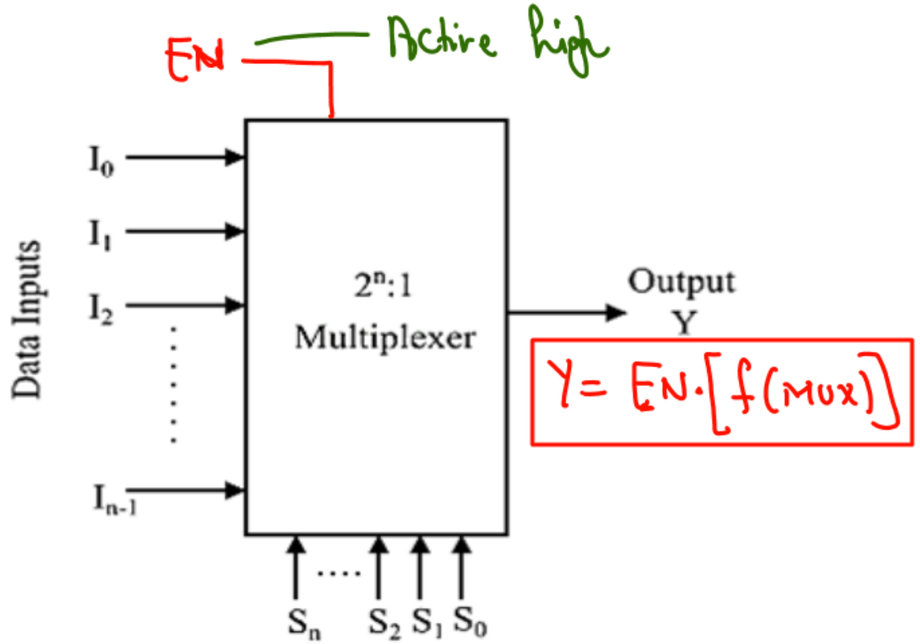
Input lines = $2^{\text{select lines}}$

Imp

*Mux are also called as 'Universal logic' & 'AND-OR logic'

Active high and Active low Multiplexer

Active High: The Enable switch of the MUX must be 1 to turn on the multiplexer

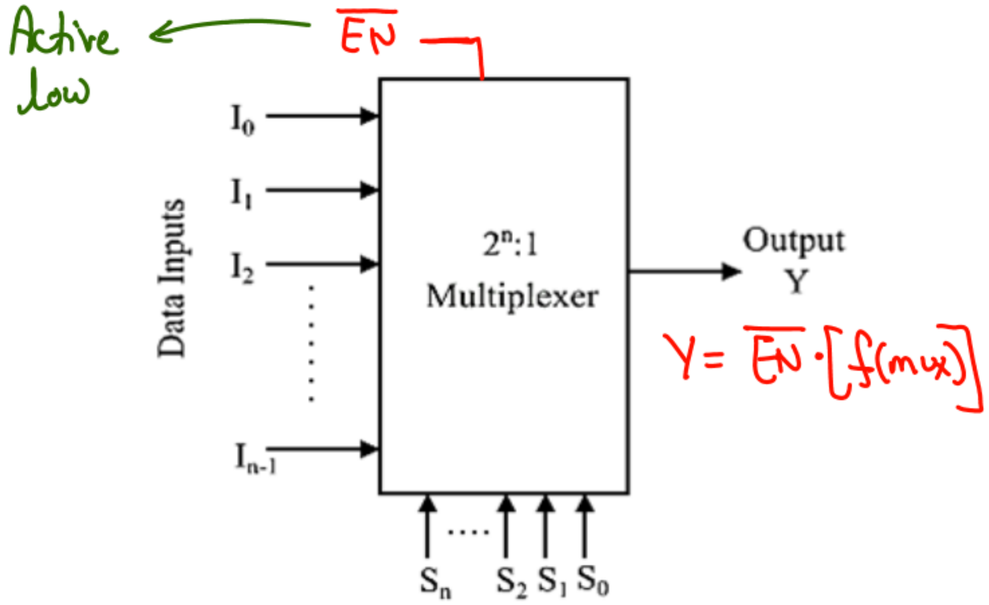


$Y = EN \cdot [f(MUX)]$

if EN = 1, then MUX works
EN = 0, then MUX will not work

Active low: If the enable switch is zero then mux work.

If EN = 0 then MUX turn ON
EN = 1 then MUX will Not work



$Y = \overline{EN} \cdot [f(mux)]$

2×1 Multiplexer

A 2×1 mux consists of two input lines and one select line (S) and one output (Y)

TRUTH TABLE			
S (Select)	D_0	D_1	Y (Output)
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

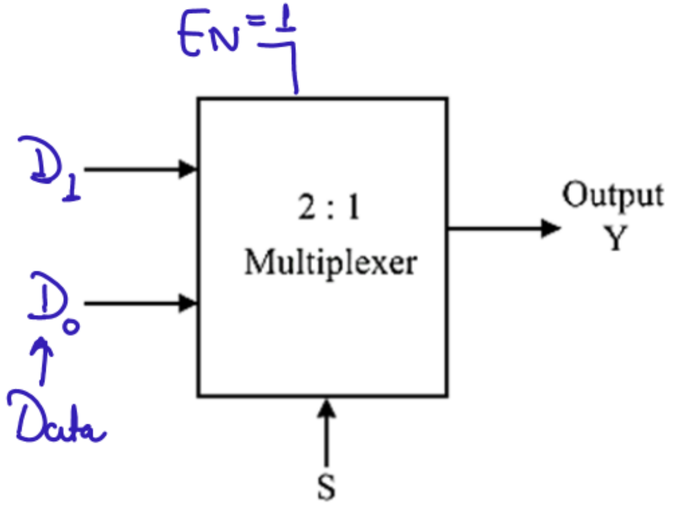
$S=0 \quad Y=D_0$

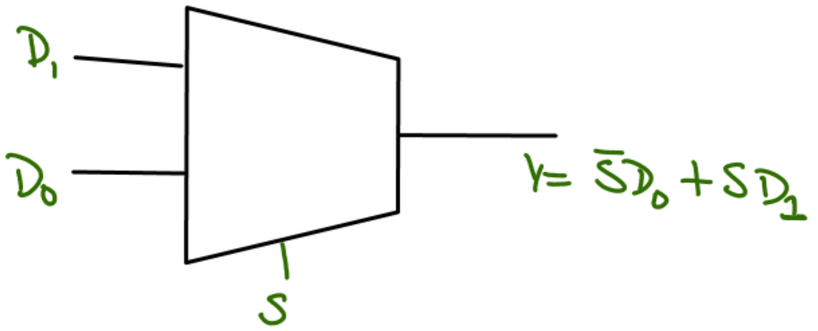
$S=1 \quad Y=D_1$

S	Y	
0	D_0	m_0
1	D_1	m_1

SOP

$$Y = \bar{S} D_0 + S D_1$$





A	B	C	Y
0	0	0	0
0	0	1	1 m_1
0	1	0	1 m_2
0	1	1	0
1	0	0	1 m_4
1	0	1	0
1	1	0	0
1	1	1	1 m_7

$$Y = m_1 + m_2 + m_4 + m_7$$

$$Y = \sum m(1, 2, 4, 7)$$

$$Y = m_1 Y_1 + m_2 Y_2 + m_4 Y_4 + m_7 Y_7$$

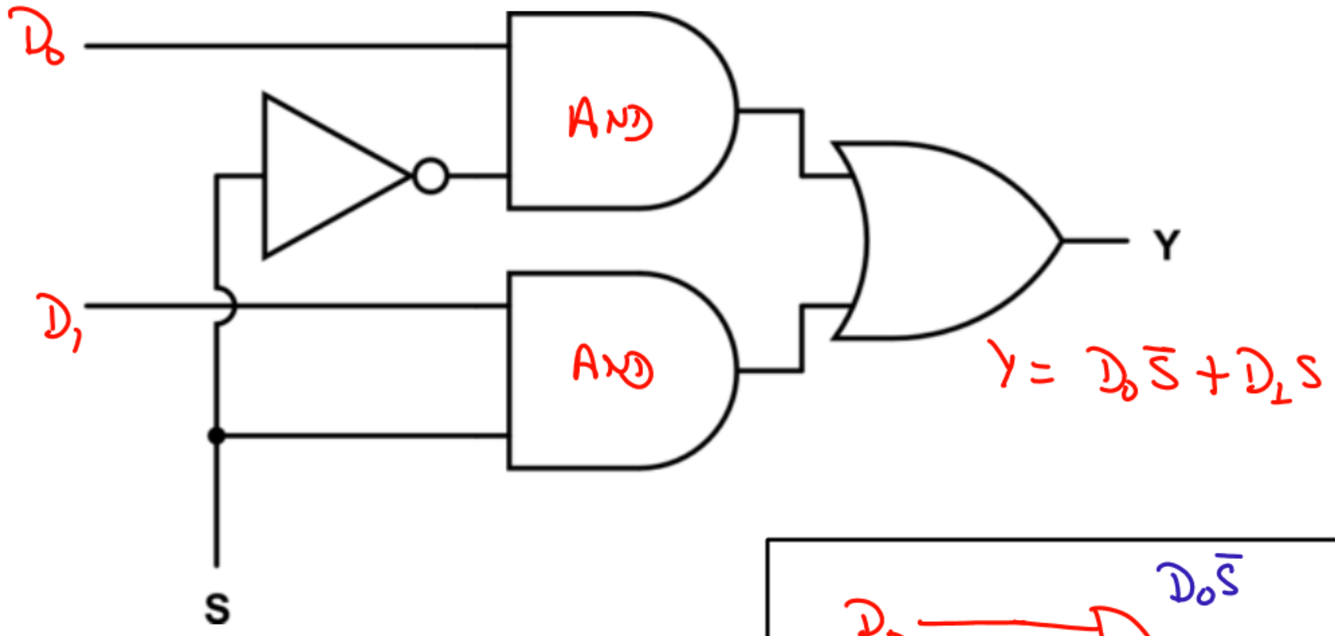
$Y_1 \downarrow 1$, $Y_2 \downarrow 1$, $Y_4 \downarrow 1$, $Y_7 \downarrow 1$

$$Y = Y_i (m_1 + m_2 + m_4 + m_7)$$

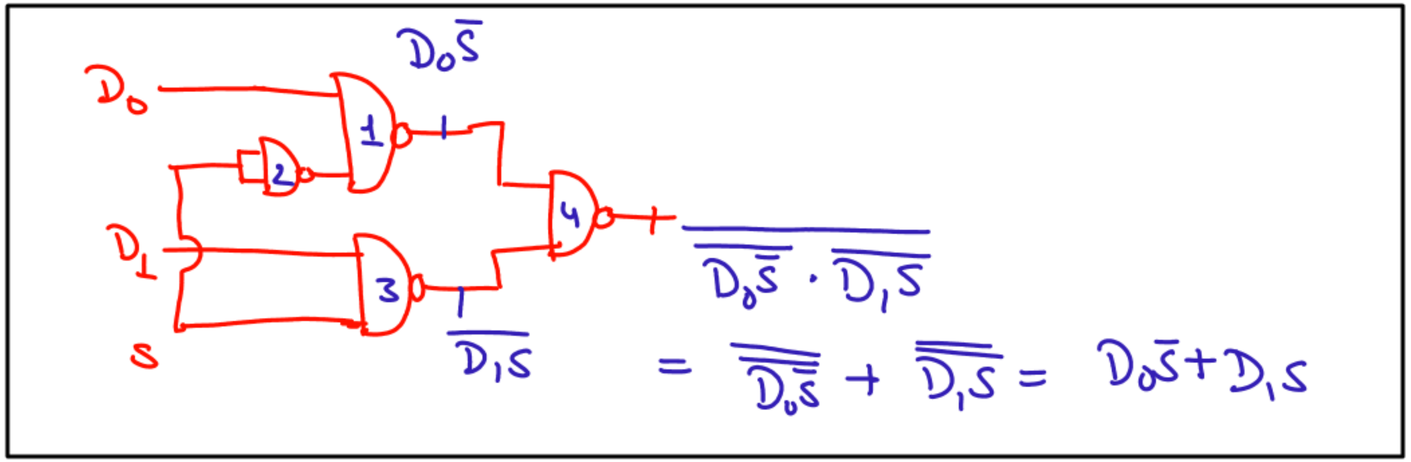
$Y_i \downarrow E$

2×1 Multiplexer

* 4 NAND Gates are required to Create 2x1 mux



$Y = D_0\bar{S} + D_1S$

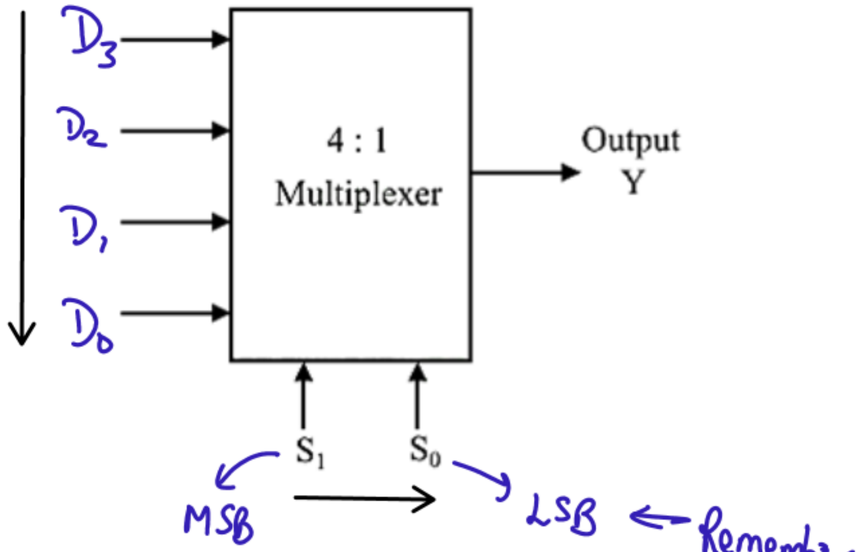


4×1 Multiplexer

* The sequence of Input lines & Select lines may be vary according to the sequence.

A 4:1 Multiplexer (4-to-1 MUX) is a combinational circuit that selects one of four data inputs and routes it to a single output, based on the values of two select lines.

Data Inputs: D0, D1, D2, D3 (4 inputs)
Select Inputs: S1, S0 (2 bits)
Enable (EN) – if present, enables/disables the MUX
Y – Output



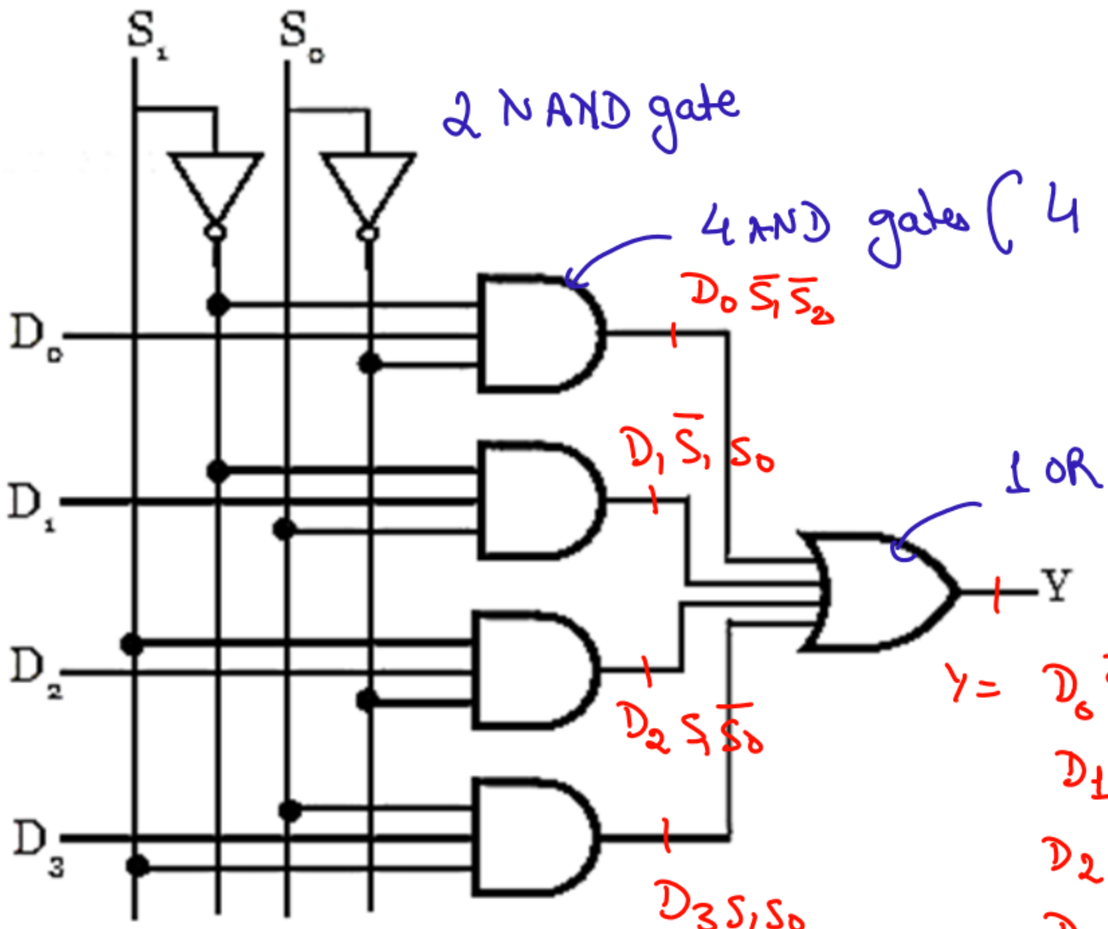
TRUTH TABLE		
S1	S0	Output (Y)
0	0	D0
0	1	D1
1	0	D2
1	1	D3

$\bar{S}_1\bar{S}_0D_0$
$\bar{S}_1S_0D_1$
$S_1\bar{S}_0D_2$
$S_1S_0D_3$

* No of Inverters required = no. of Select lines for a mux

$$Y = \bar{S}_1\bar{S}_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$$

4×1 Multiplexer



2 NAND gate

4 AND gates (4 Combinations of Select line)

$D_0 \bar{S}_1 \bar{S}_0$

$D_1 \bar{S}_1 S_0$

$D_2 S_1 \bar{S}_0$

$D_3 S_1 S_0$

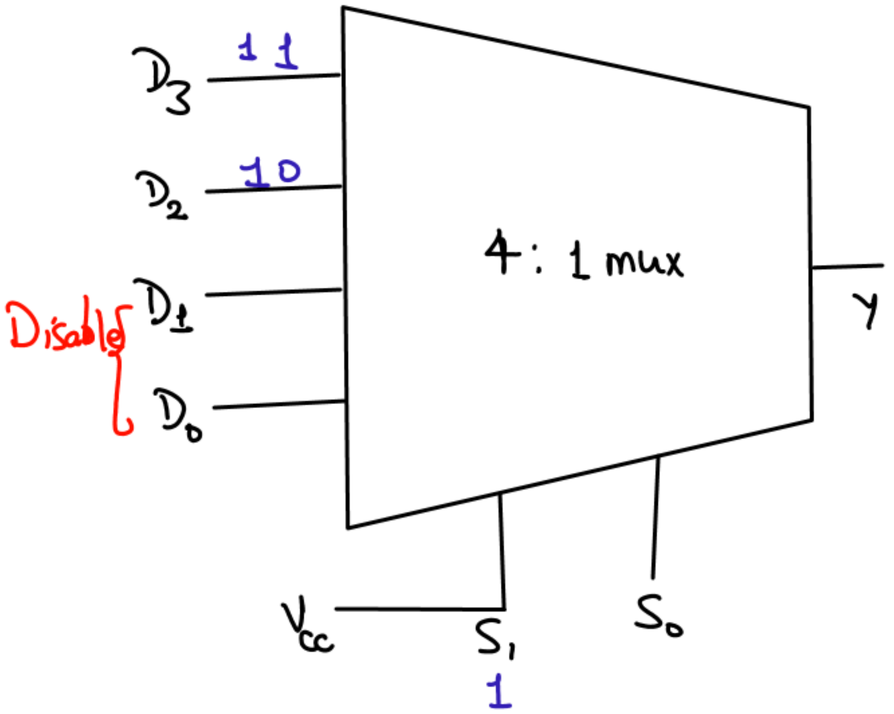
1 OR gate

$Y = D_0 \bar{S}_0 \bar{S}_1 + D_1 S_0 \bar{S}_1 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$

* 7 NAND gates are required to design 4×1 mux

Connecting Vcc to S1 pin in 4:1 Multiplexer

high



Possible Combinations

S_1	S_0	
1	0	D_2
1	1	D_3

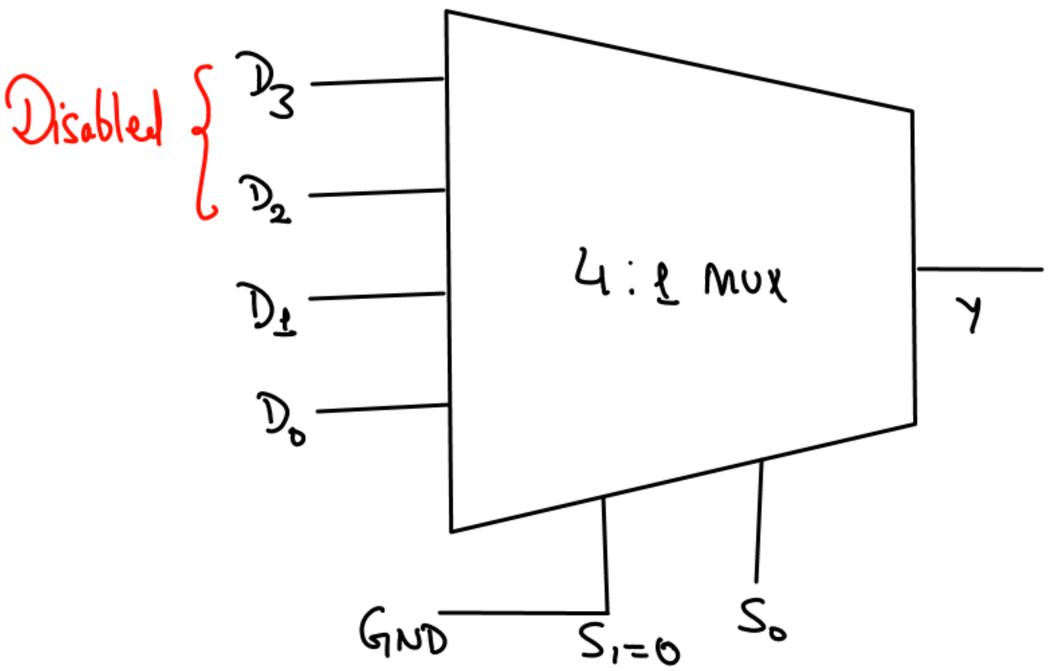
} output

* D_0 & D_1 can not get connected to the output if $S_1 = 1$.

∴ D_0 & D_1 are disabled input

Connecting GND to S1 pin in 4:1 Multiplexer

low



Possible Combinations

S_1	S_0	Y
0	0	D_0 } Reaches to the o/p
0	1	D_1 }

But,

D_3 & D_2 will not reach to the o/p

$\therefore$ D_3 & D_2 are disabled.

8×1 Multiplexer

An 8:1 multiplexer selects one of 8 data inputs (D_0–D_7) and sends it to the output Y, based on 3 select lines (S_2, S_1, S_0).

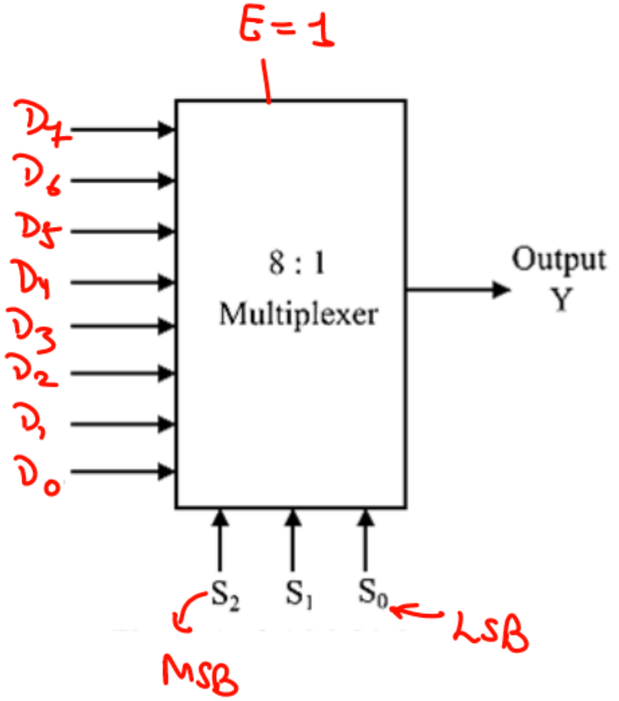
TRUTH TABLE			
S_2	S_1	S_0	Output (Y)
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

$\overline{S_2}\,\overline{S_1}\,\overline{S_0}\,D_0$

$\overline{S_2}\,\overline{S_1}\,S_0\,D_1$

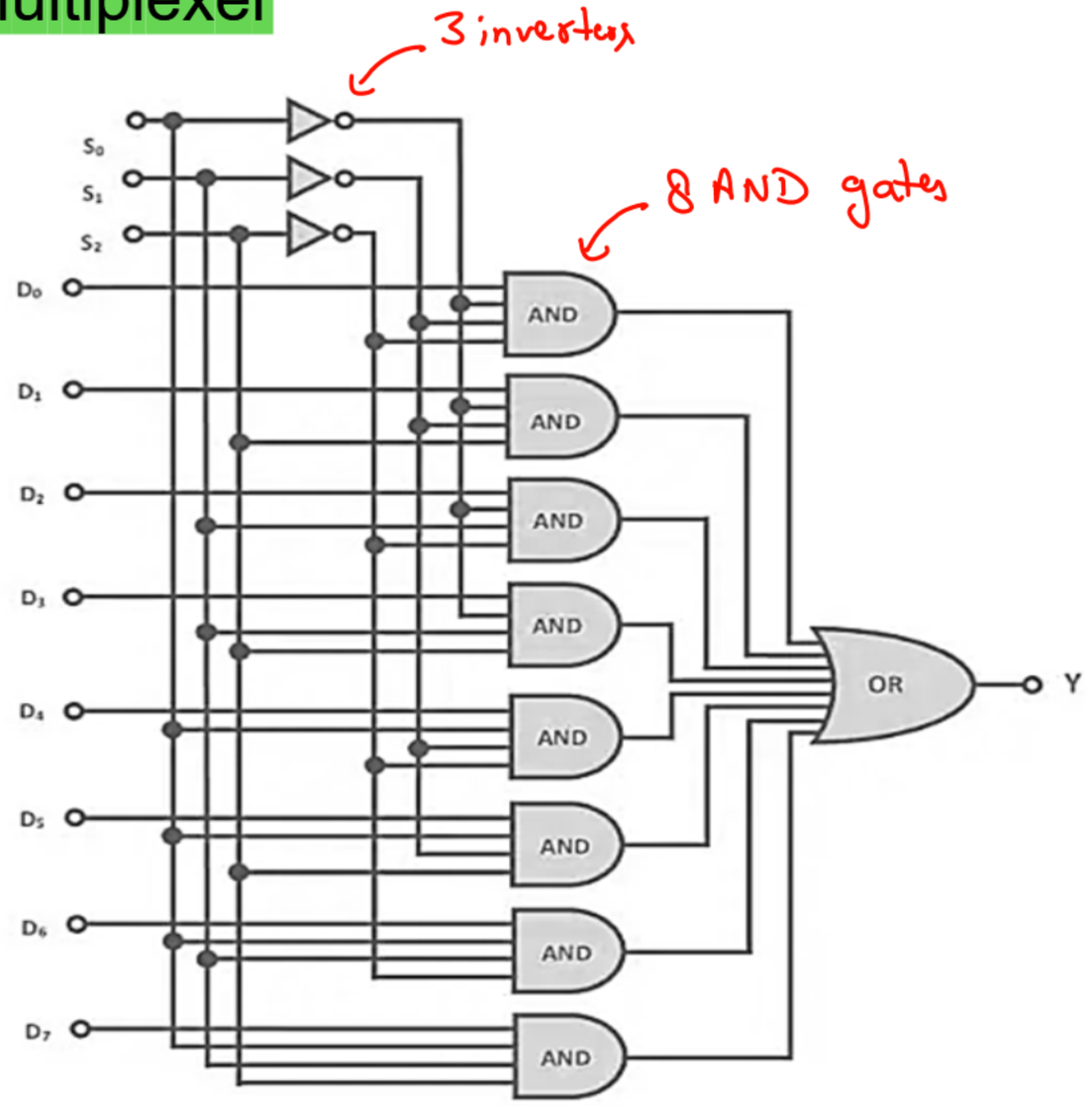
$\overline{S_2}\,S_1\,\overline{S_0}\,D_2$

$S_2\,S_1\,S_0\,D_7$



$$Y = \overline{S_2}\,\overline{S_1}\,\overline{S_0} \cdot D_0 + \overline{S_2}\,\overline{S_1}\,S_0 \cdot D_1 + \overline{S_2}\,S_1\,\overline{S_0} \cdot D_2 + \overline{S_2}\,S_1\,S_0 \cdot D_3 + S_2\,\overline{S_1}\,\overline{S_0} \cdot D_4 + S_2\,\overline{S_1}\,S_0 \cdot D_5 + S_2\,S_1\,\overline{S_0} \cdot D_6 + S_2\,S_1\,S_0 \cdot D_7$$

8×1 Multiplexer



12 NAND gates are required to design 8:1 mux

Similarly we can design 16:1 mux, 32 x 1 mux & So on.

↓

but It is not feasable

↓

minimize the MUX by distributing the high order multiplexer to the sequence of low order MUX.

Tomorrow →

Implementation of Logic gates using Mux

&

functions using mux