DEMULTIPLEXERS [COMPLETE]

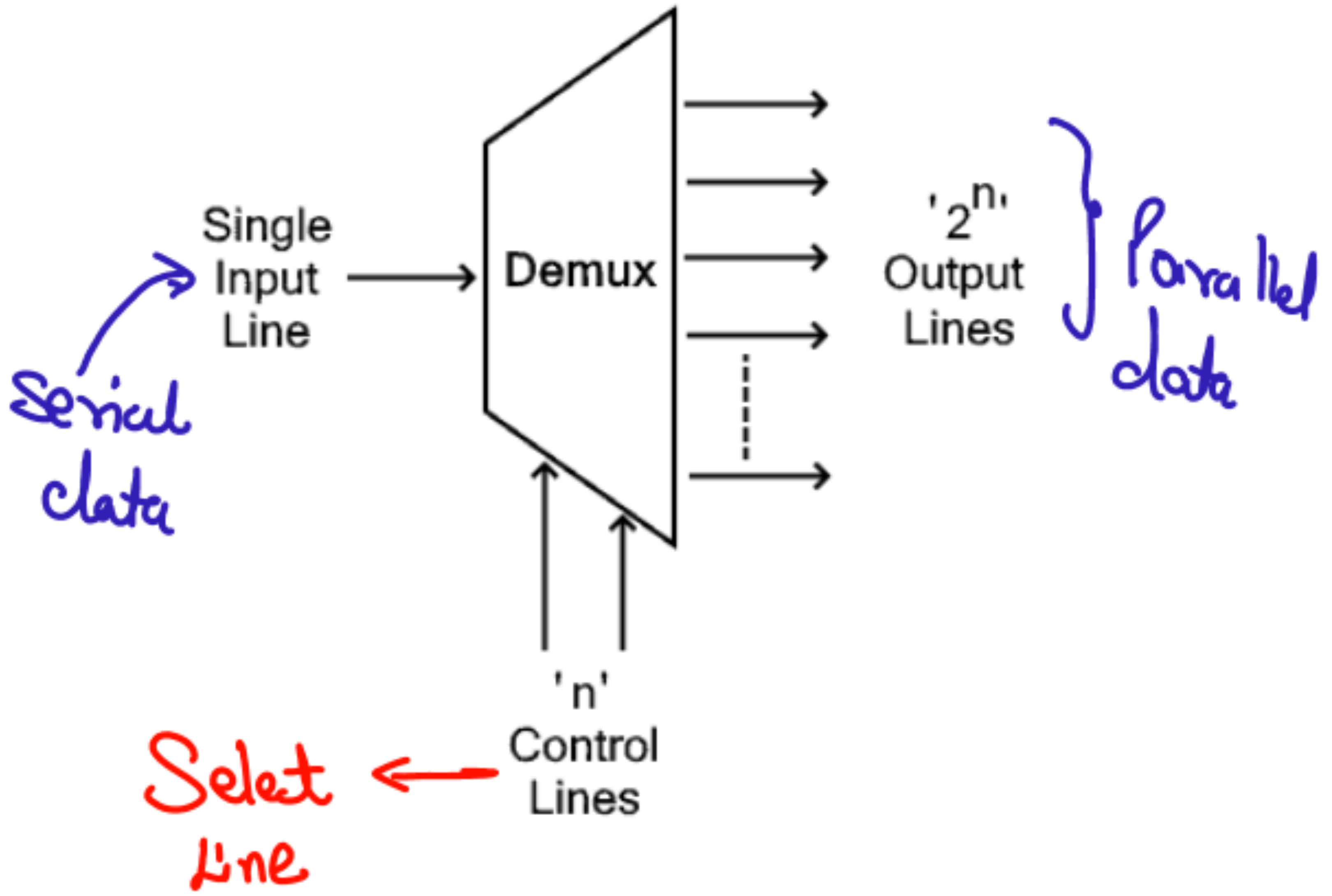
Demultiplexers (DEMUX)

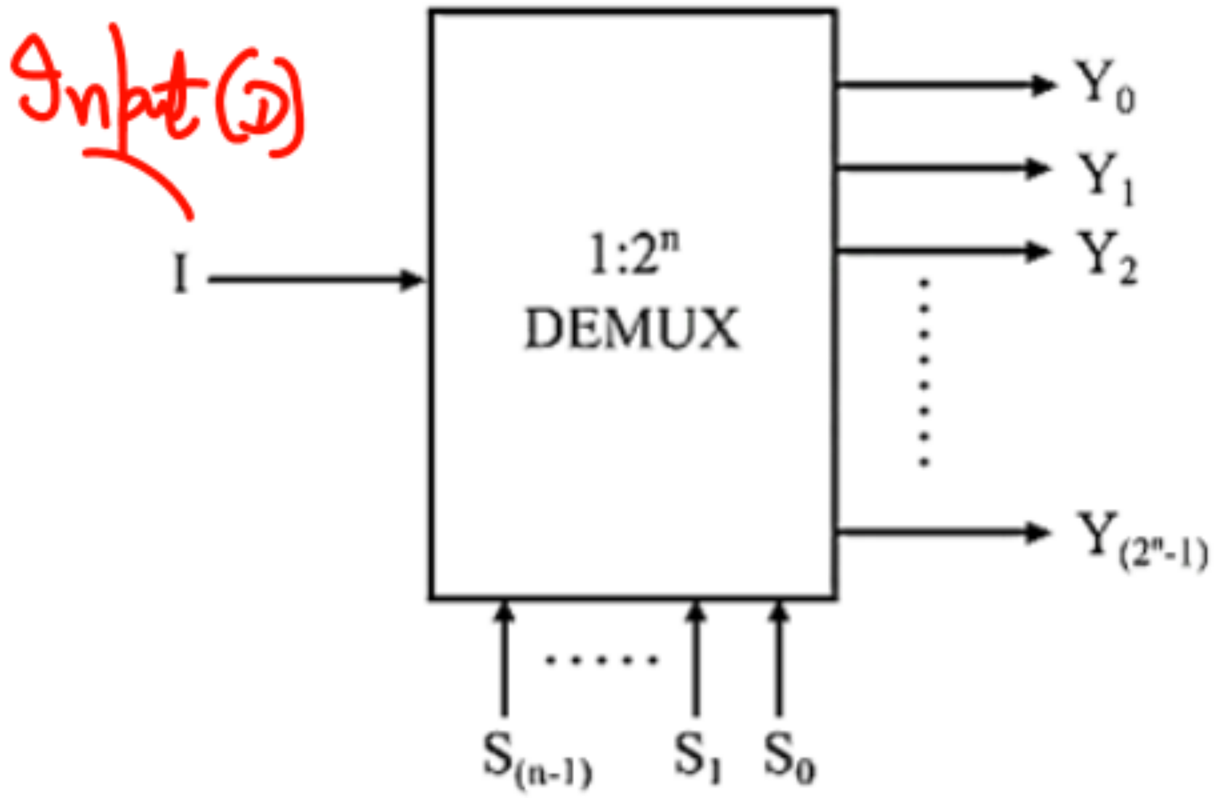
A demultiplexer is a combinational logic circuit that takes one input and routes it to one of many output lines, based on the values of select lines (control inputs).

- Opposite of a multiplexer (MUX).
- It distributes data from one input line to multiple output lines.
- The output line activated depends on the combination of select inputs.
- It convert the serial signals into parallel data output lines.

Data on one line where bits transferred successively.
↓
one after one
Eg 101101 →

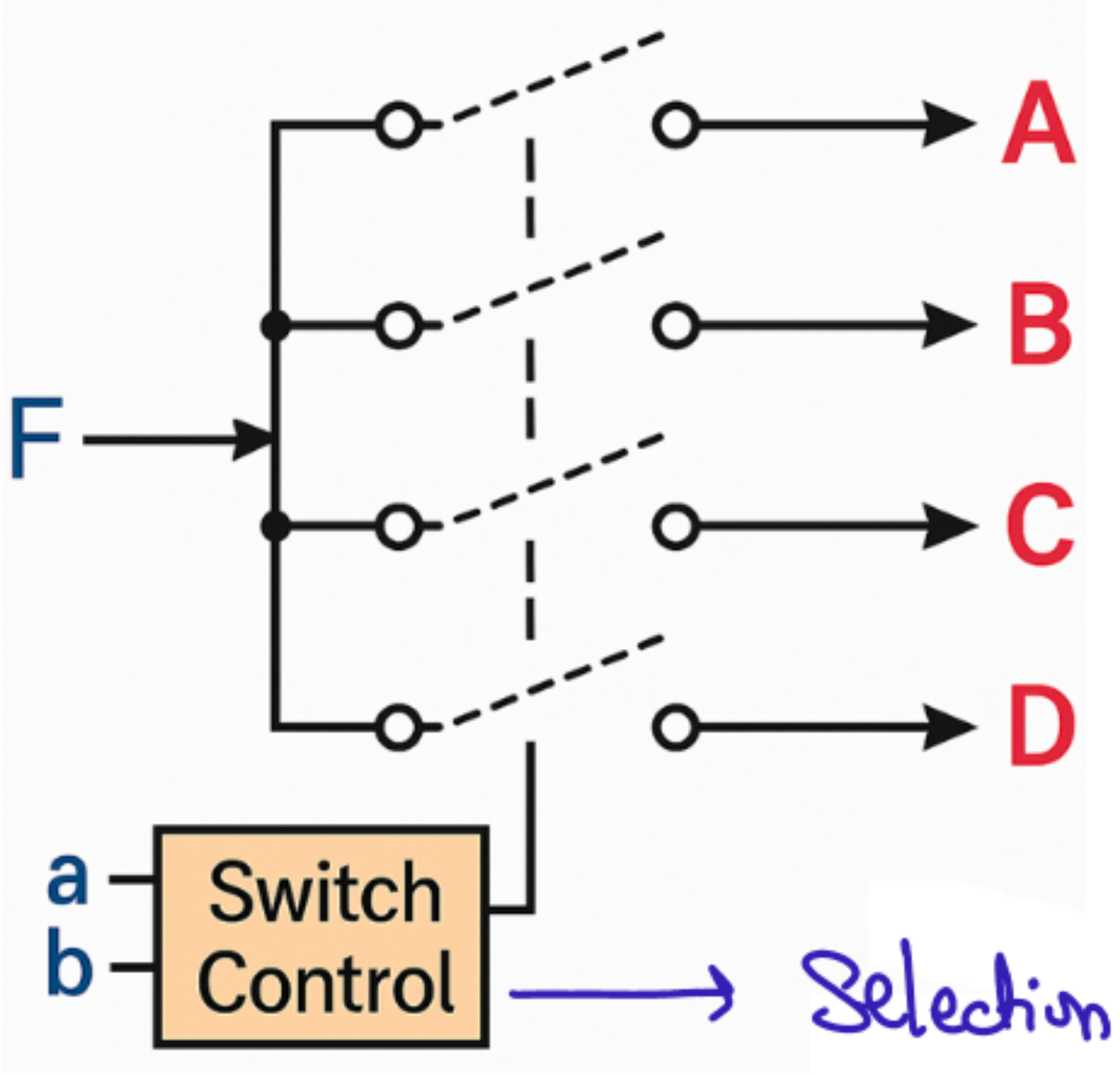
All bits travels in parallel or separate line





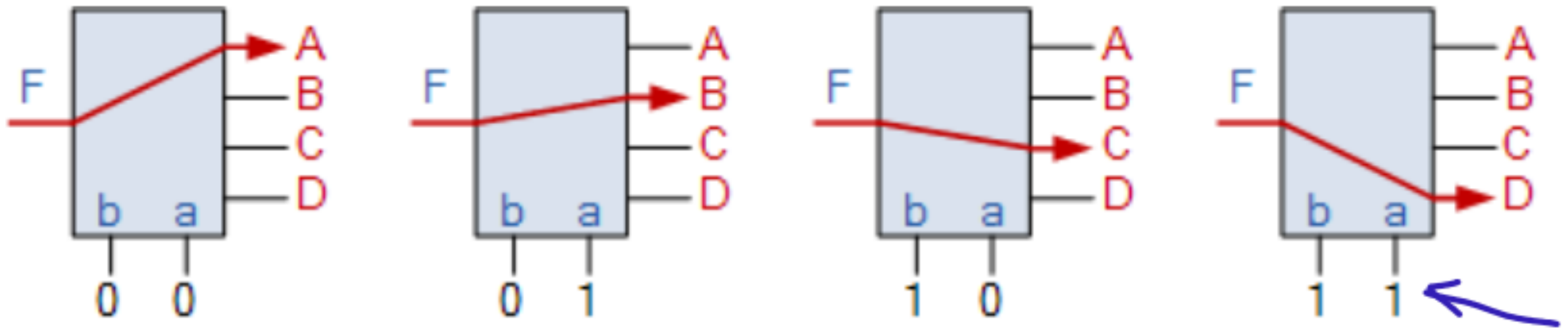
Visual representation of Demultiplexers

Common Input



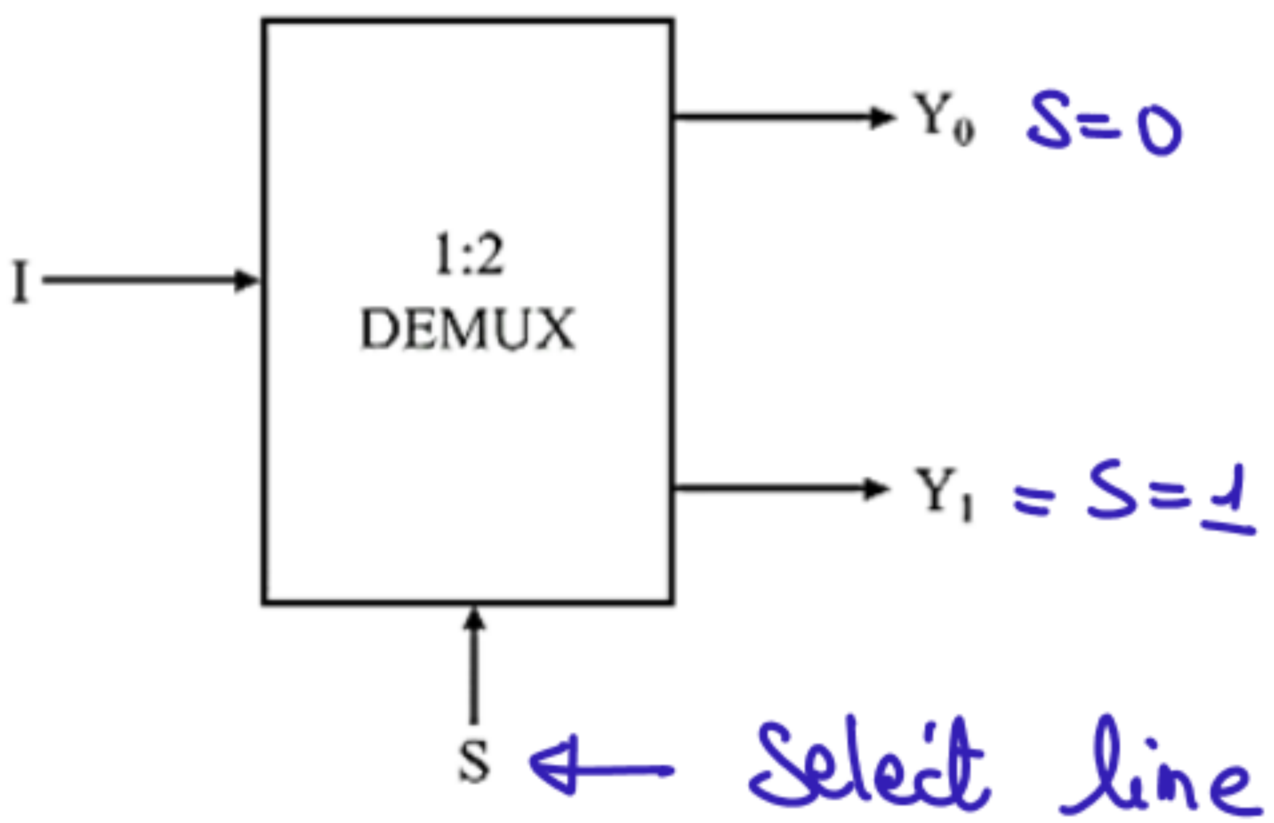
Data Outputs

Selection switch → tells us that which output has to be connect to input

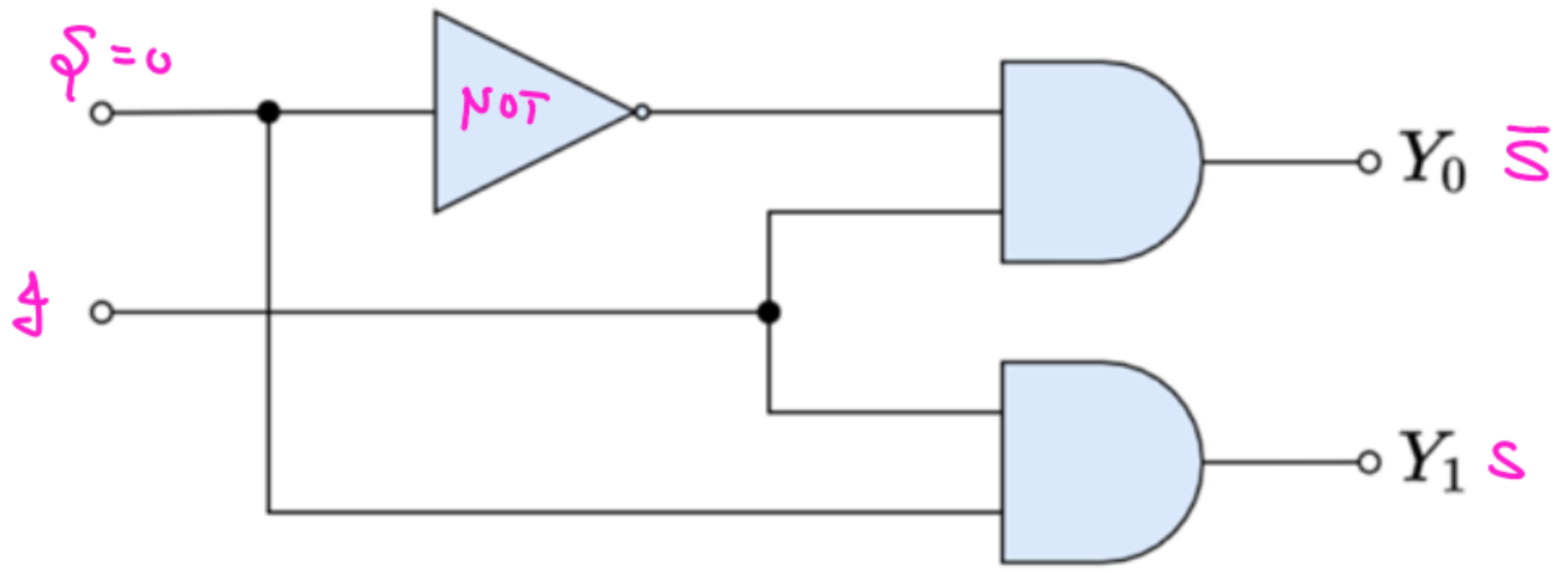


select line combinations.

1x2 Demultiplexer

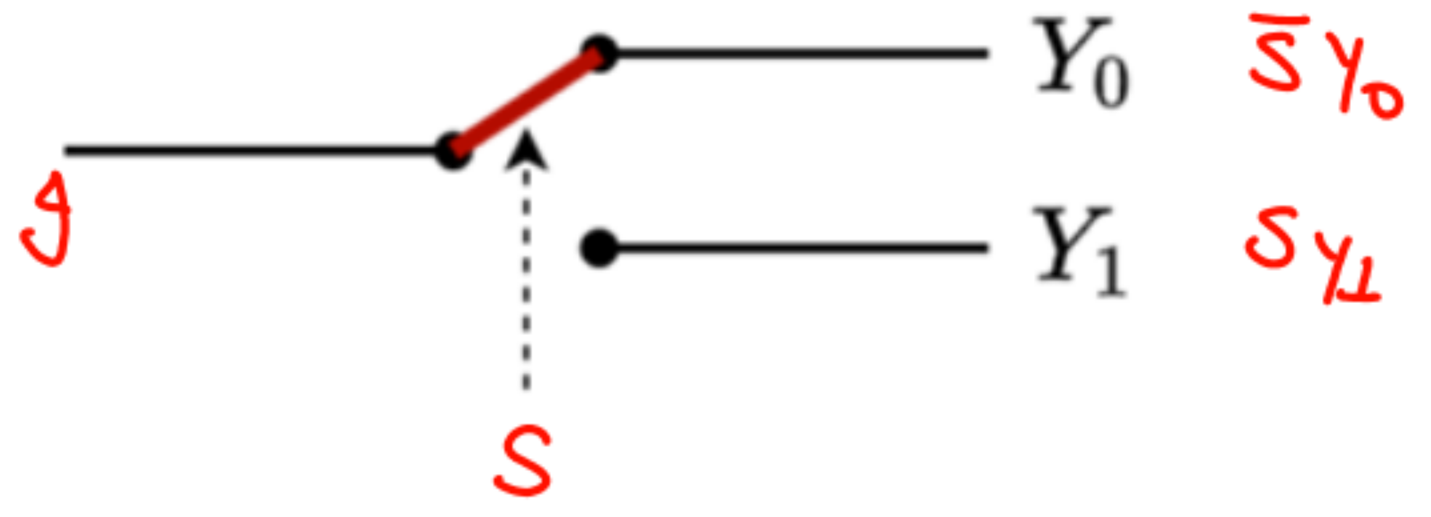


Select line is associated with output lines



Select Line	Outputs	
S	Y_1	Y_0
0	0	I
1	I	0

$\bar{S}Y_0$
SY_1
Parallel output

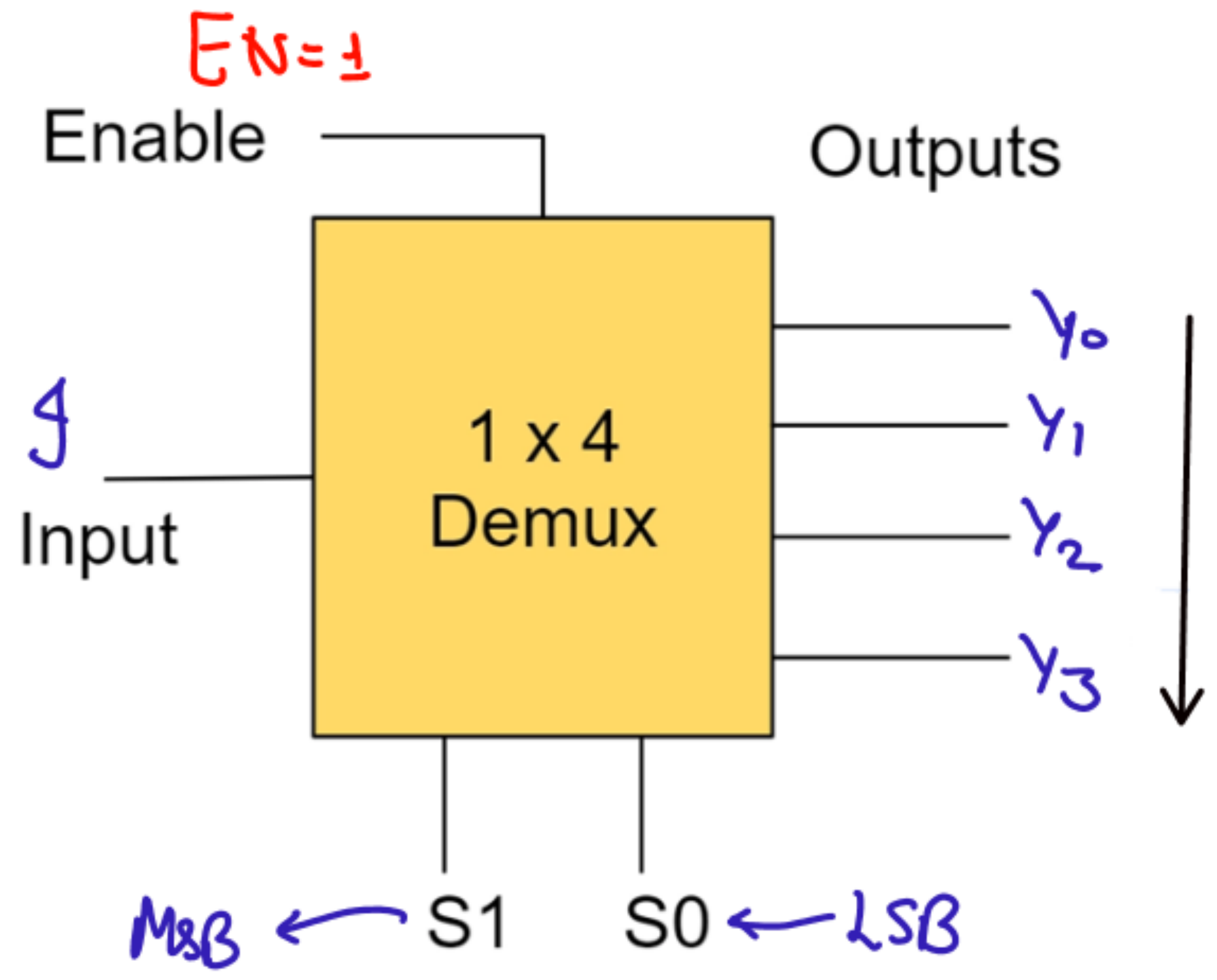


Switch Equivalent

Select	Input	Outputs Y_1	Y_0
0	0	X	0
0	1	X	1
1	0	0	X
1	1	1	X

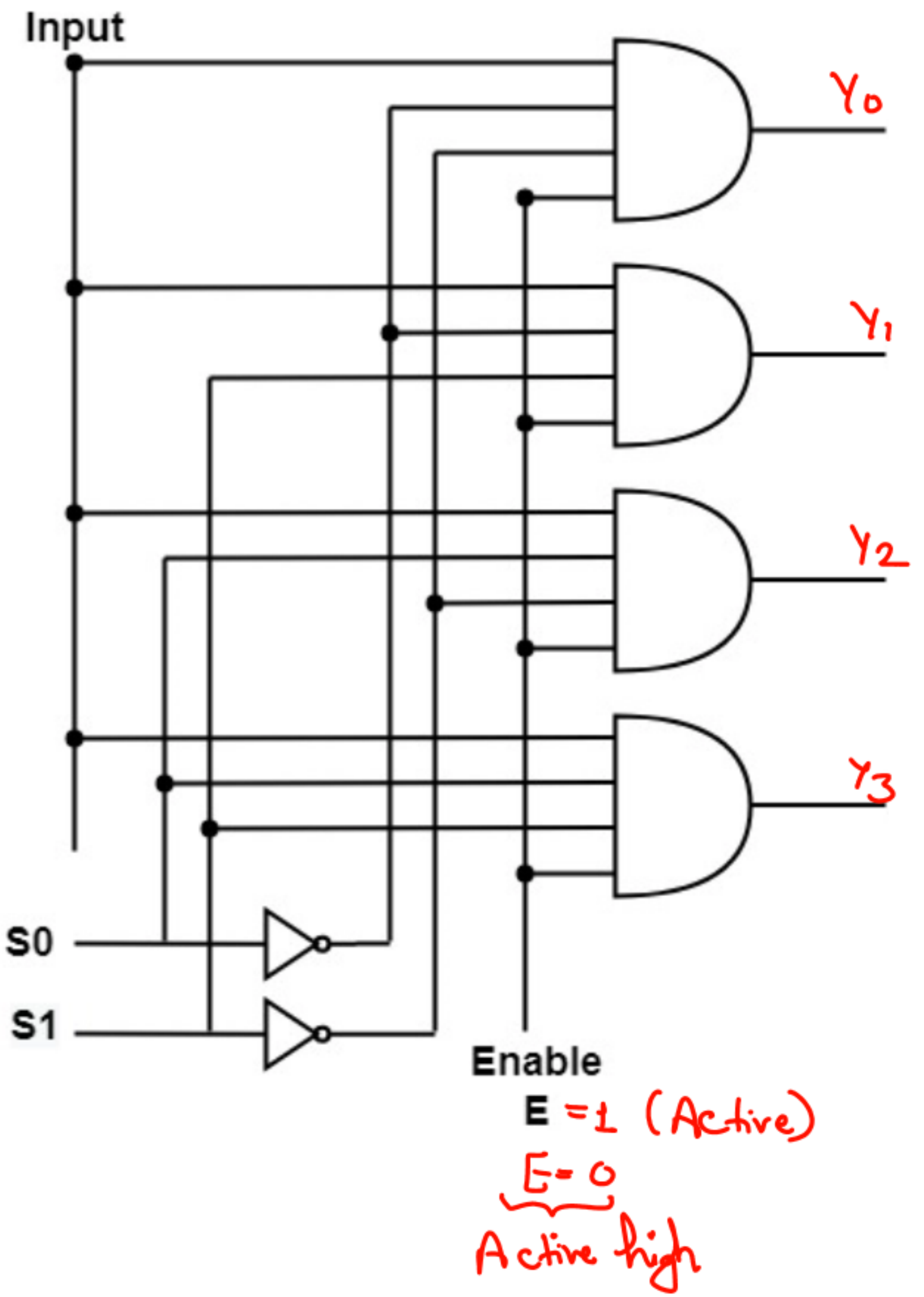
don't Care

1x4 Demultiplexer



* Each input is a product term hence represented by AND gate.

Select Line		Outputs				
S1	**S0**	**Y3**	**Y2**	**Y1**	**Y0**	
0	0	0	0	0	I	$\bar{S_1}\bar{S_0}Y_0$
0	1	0	0	I	0	$\bar{S_1}S_0Y_1$
1	0	0	I	0	0	$S_1\bar{S_0}Y_2$
1	1	I	0	0	0	$S_1S_0Y_3$



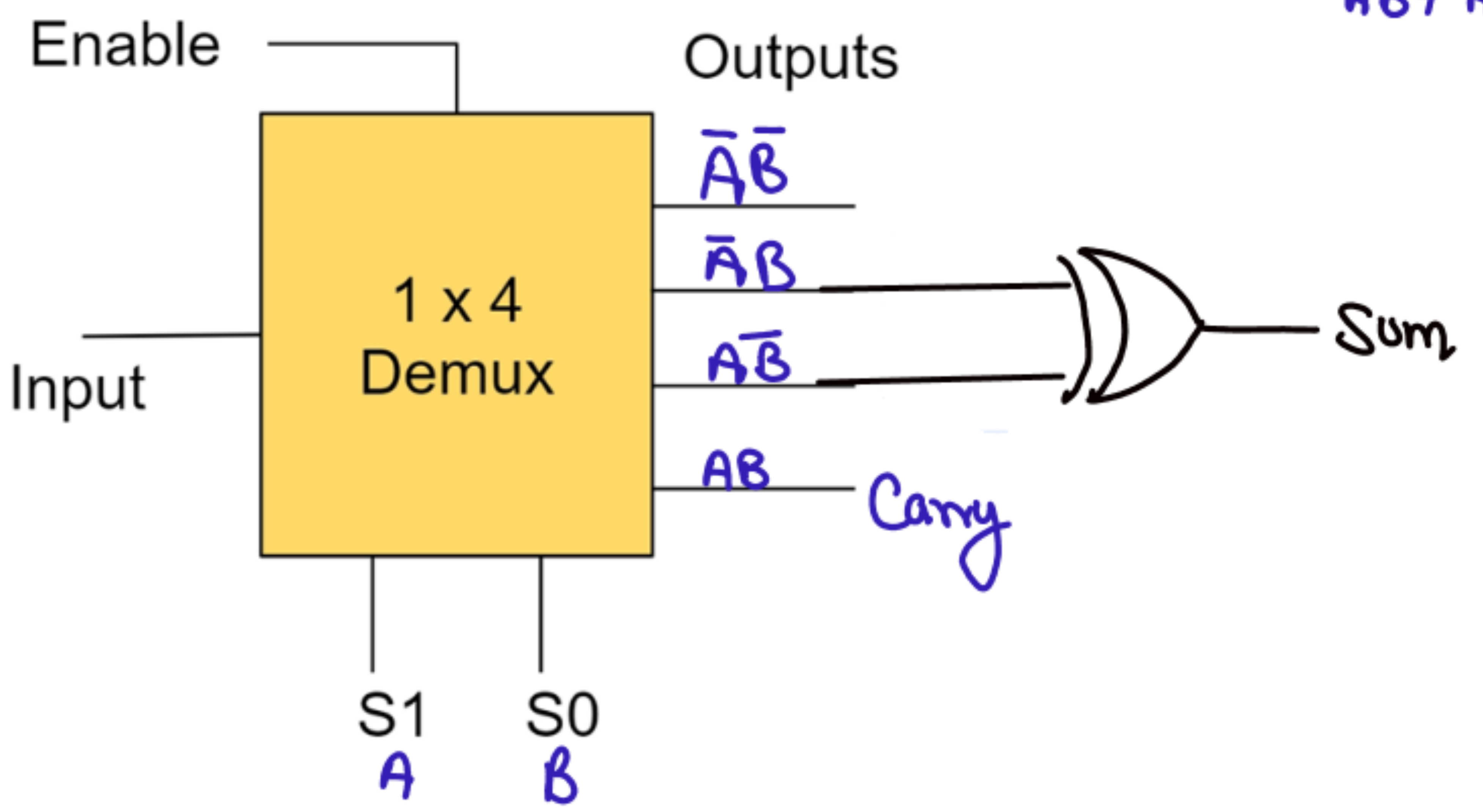
Half adder and Half subtractor using DeMux

- Half Adder

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$\text{Sum} = A \oplus B$
$= A\bar{B} + \bar{A}B$

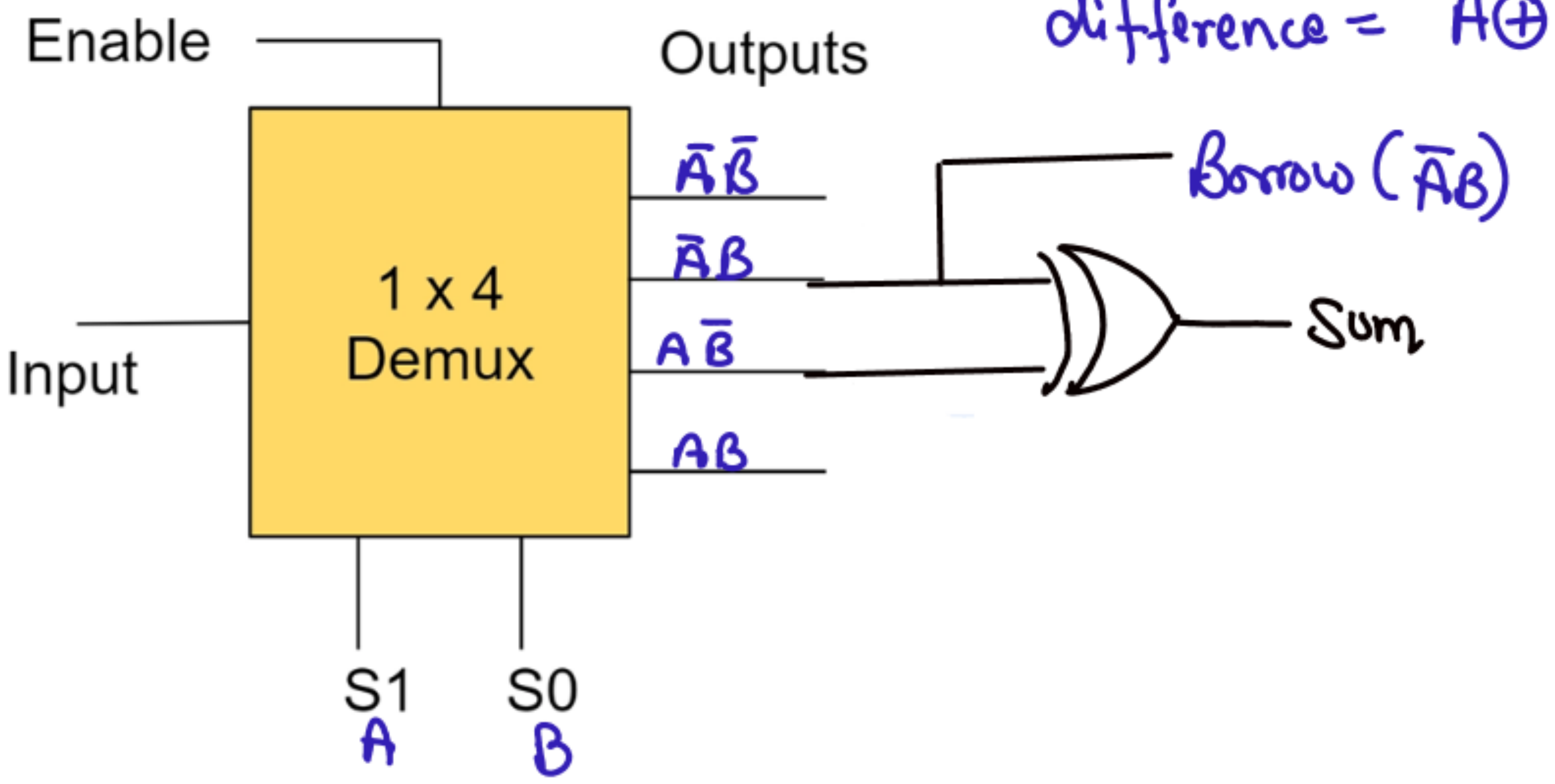
$\text{Carry} = AB$

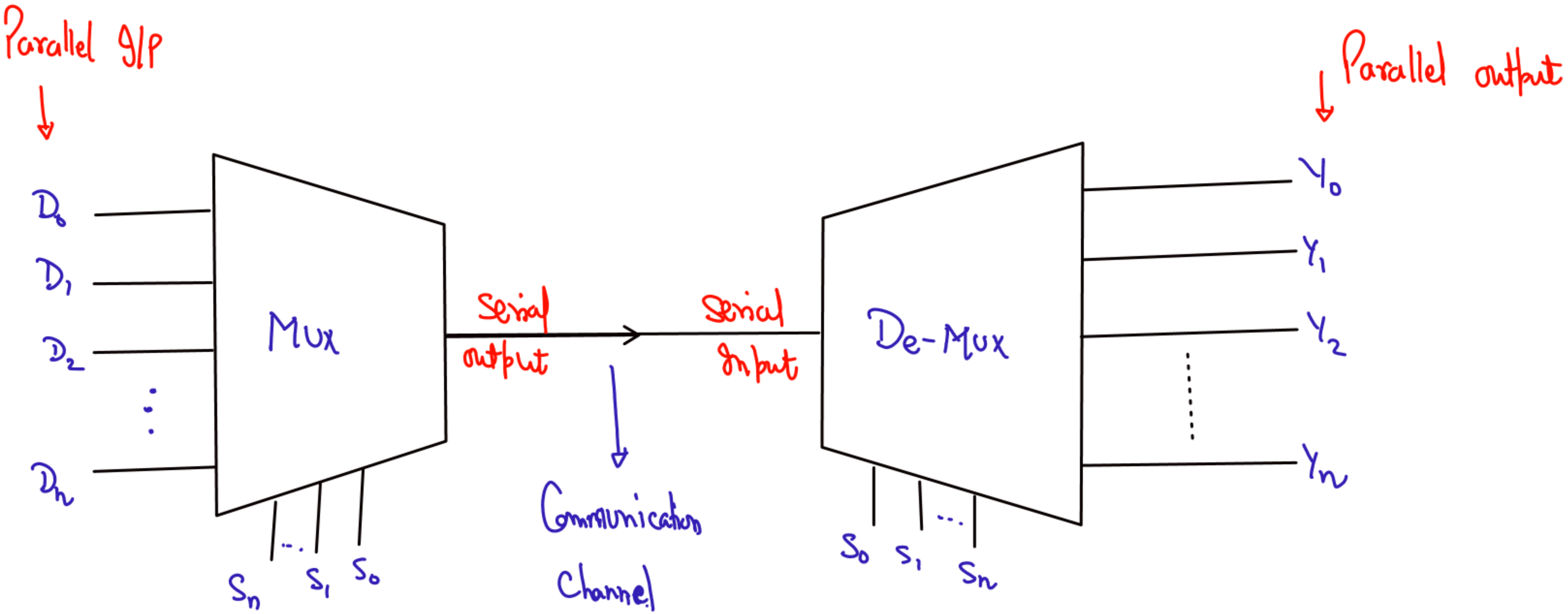


- Half Subtractor

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$\text{difference} = A \oplus B$ $\quad \text{Borrow} = \bar{A}B$





Parallel I/P
D0
D1
D2
Dn
Mux
Sn
S1
S0
Serial output
Communication Channel
Serial input
De-Mux
S0
S1
Sn
Parallel output
Y0
Y1
Y2
Yn

1x8 DeMux using 1x4 DeMux

- We can build a 1×8 Demultiplexer using two 1×4 Demultiplexers and an extra select line as a control.
- we requireTWO 1×4 Demultiplexers in SECOND stage in order to get the final EIGHT outputs.

LSB

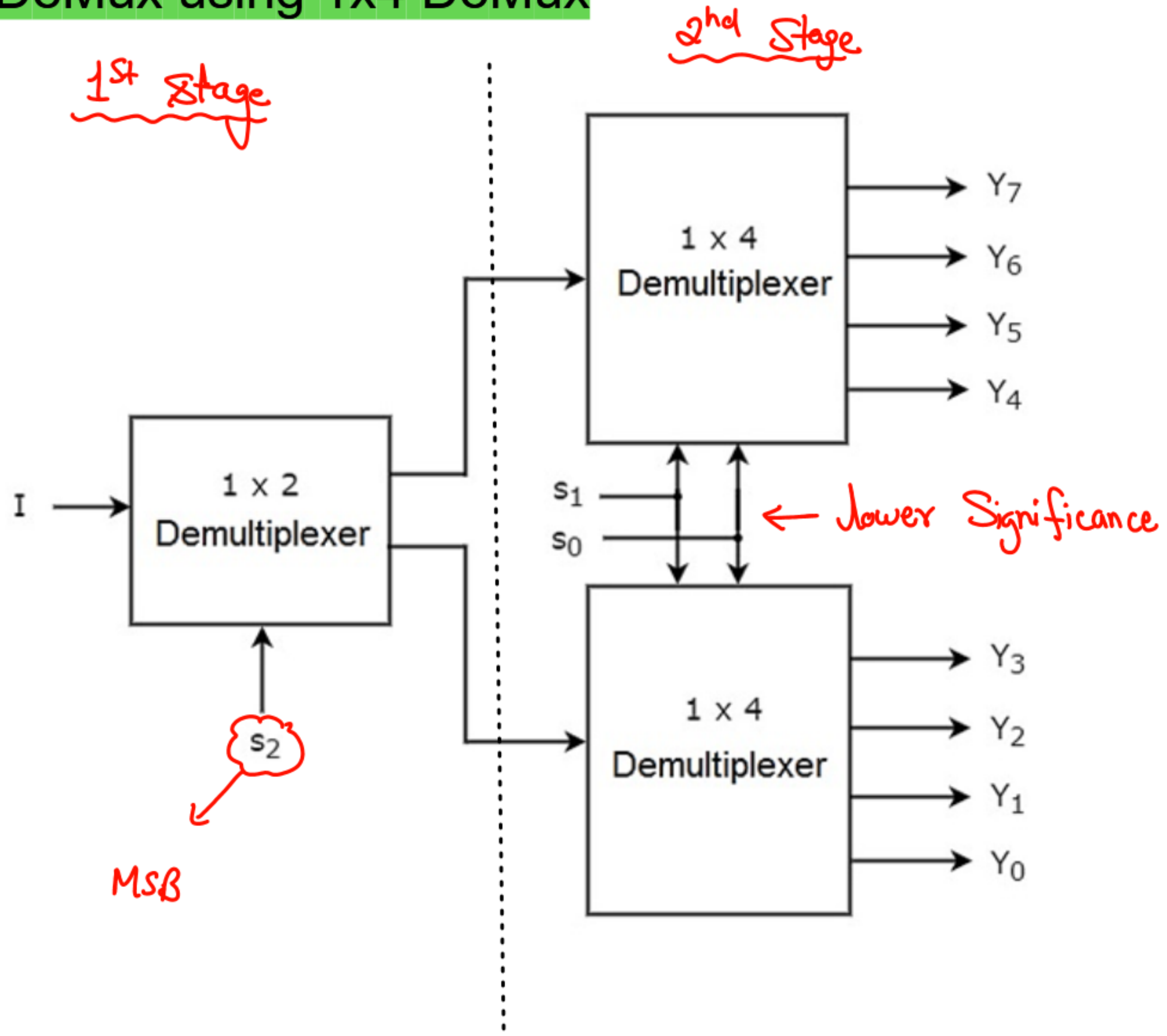
MSB

Selection Inputs			Outputs							
s2	s1	s0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
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

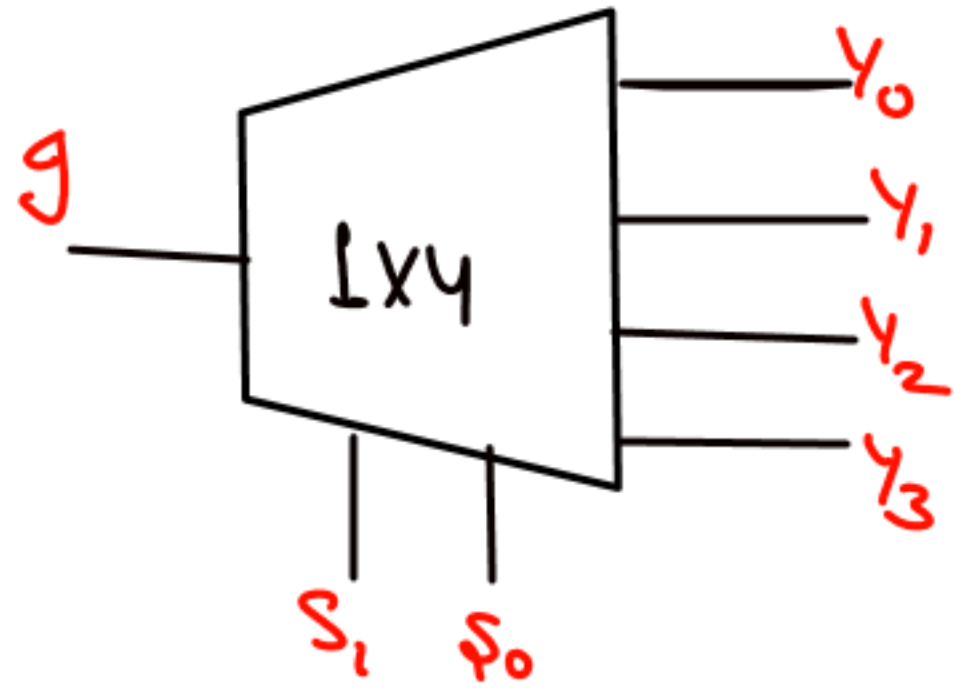
S_2 as 1:2 Demux at first stage

use S_1 & S_0 at second stage.

1x8 DeMux using 1x4 DeMux



1 x 4 Demux



let $S_1 = 0$

Possible o/p

S_1	S_0	
0	0	$\rightarrow Y_0$
0	1	$\rightarrow Y_1$

let $S_0 = 0$

Possible outputs

S_1	S_0	
0	0	$- Y_0$
1	0	$\rightarrow Y_2$

High order DeMux to Low order DeMux : GENERAL FORMULA

- In MUX, MSB is used as select line at last stage but in DeMux, MSB is used as select line at first stage.

In general, to implement 1 : B Demux using 1 : A Demux then

$B/A \longrightarrow K_1$ (n^{th} stage) $\longleftarrow$ last stage

$K_1/A \longrightarrow K_2$ ($n-1^{th}$ Stage)

$K_2/A \longrightarrow K_3$ ($n-2^{th}$ Stage)

$\vdots$

$K_{n-1}/A = K_n = 1$ (first Stage)

* If $K_{n-1}/A = K_n \neq 1$, then implement 1 : A Demux at first stage.

Total No. of DeMux $= K_1 + K_2 + K_3 + \dots K_n = \sum_{i=0}^{n} K_i$

Q.1 How many 1:4 DeMuxes are required in 2nd stage to implement 1:64 DeMux?

1 : A   1 : B

$B/A \rightarrow K_i$

$64/4 \rightarrow 16 \rightarrow$ last stage (n)

$16/4 \rightarrow 4 \rightarrow$ n-1 stage

$4/4 \rightarrow 1$ (1st stage)

total 1:4 demux $\Rightarrow$ $\underset{\text{1st Stage}}{1} + \underset{\text{2nd Stage}}{4} + \underset{\text{3rd Stage}}{16}$

ANS

Q.2 How many 1:2 DeMuxes are required to implement 1:64 DeMux?

1:A   1:B

$B/A \rightarrow K_i$

$\frac{64}{2} \rightarrow 32$ (n^{th} Stage)

$32/2 \rightarrow 16$ $(n-1)$ Stage

$16/2 \rightarrow 8$ $(n-2)$ Stage

$8/2 \rightarrow 4$ $(n-3)$ Stage

$4/2 \rightarrow 2$ $(n-4)$ Stage

$2/2 \rightarrow 1$ (1^{st} Stage)

Total DeMuxes required $= 1+2+4+8+16+32$

$= 63$ deMuxes required