

4. Combinational Logic Circuits (Half Adder, Full Adder), MUX and DeMUX

COMBINATIONAL LOGIC CIRCUITS

Half adder
Addition of 2 bits

Full adder
Addition of 3 bits

Combinational circuits' output depends on present input only whereas sequential circuits' output depends on the present input as well as the past output.

Eg: Sequential → traffic lights, registers, counters

Combinational →

HALF ADDER



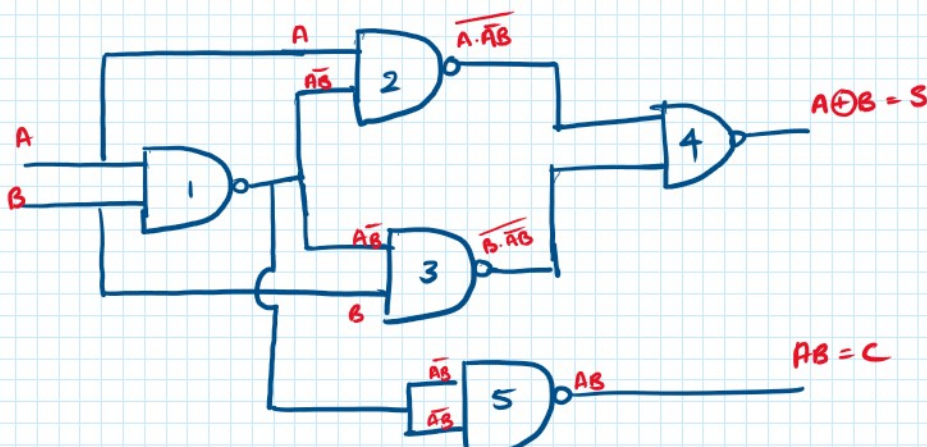
Inputs		Outputs	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

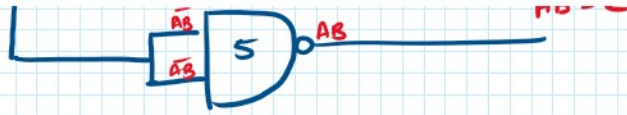
$$\text{Sum: } \bar{A}B + A\bar{B} = A \oplus B$$

$$\text{Carry: } AB$$

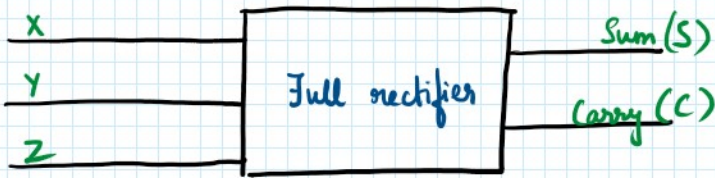
$$\begin{array}{r} + 1 \\ \hline 10 \end{array} \rightarrow \begin{array}{l} \text{Carry } C \\ \text{Sum } S \end{array}$$

Half adder using only NAND gates





Full Adder

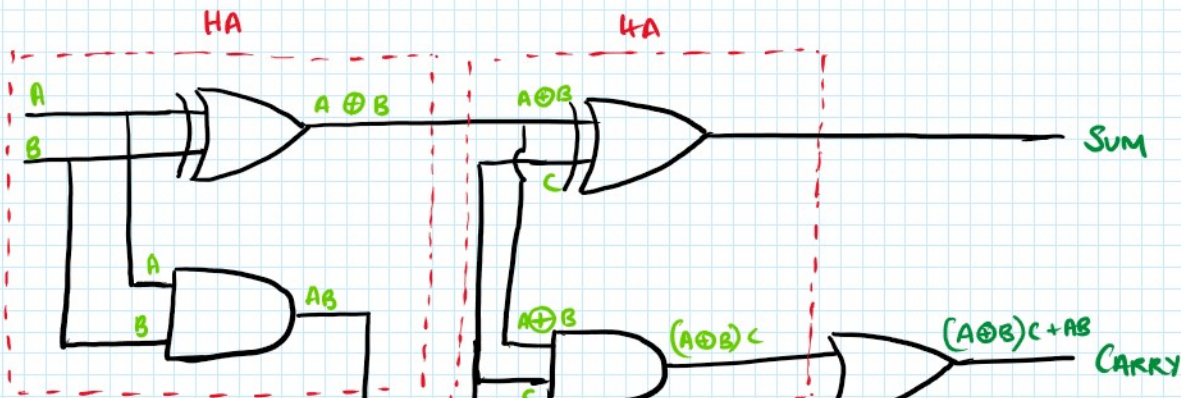
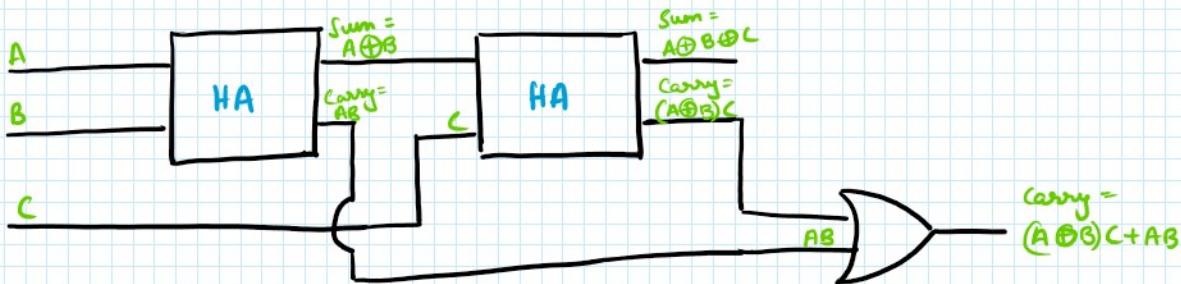


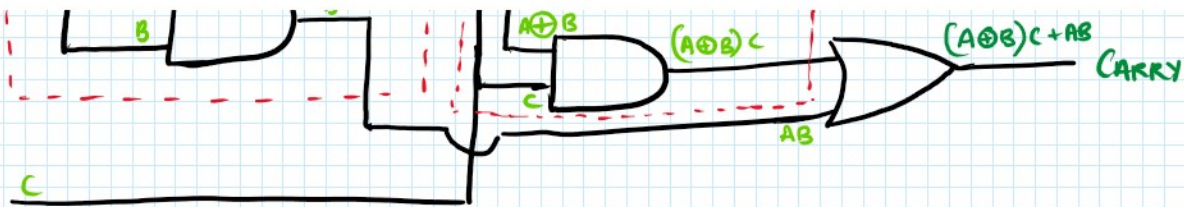
Inputs			Outputs	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$\underline{\text{Sum}}: \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz = x \oplus y \oplus z$$

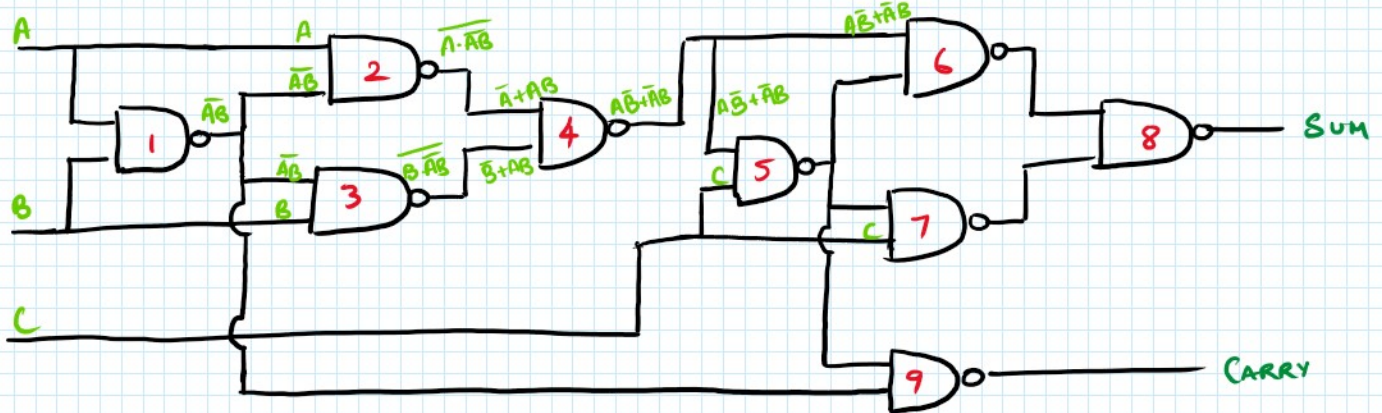
$$\underline{\text{Carry}}: xy + yz + xz = (x \oplus y)z + xy$$

Implementation of full adder using two half adders

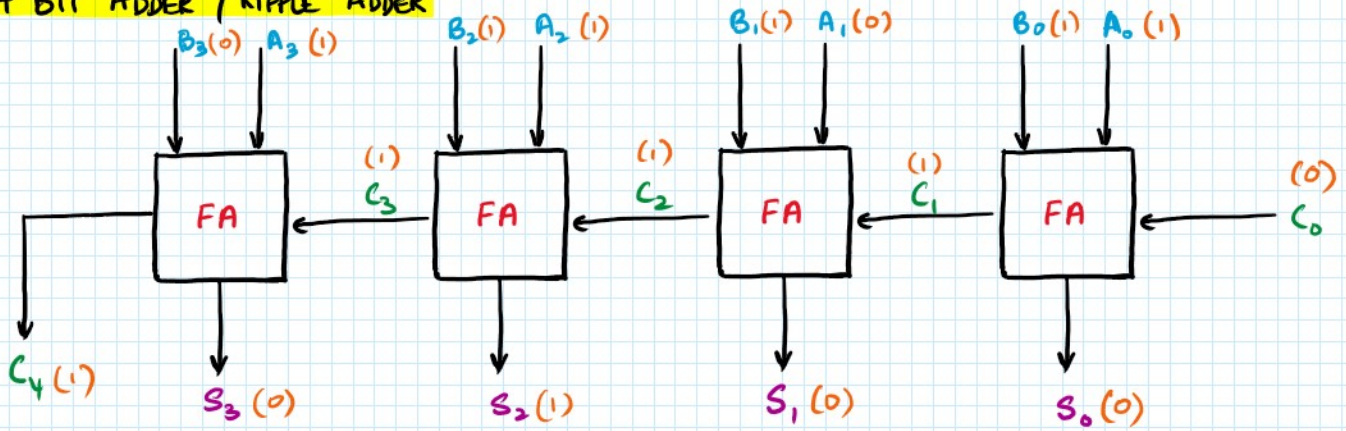




Full adder circuit using only NAND gates



4 Bit Adder / Ripple Adder



$$\begin{array}{r}
 C_3 \ C_2 \ C_1 \ C_0 \\
 1 \ 1 \ 1 \ 0 \\
 A = 1 \ 1 \ 0 \ 1 \\
 B = 0 \ 1 \ 1 \ 1 \\
 \hline
 C_4 \rightarrow 1 \ 0 \ 1 \ 0 \ 0
 \end{array}$$

MULTIPLEXER & DEMULTIPLEXER
 MUX DeMUX

} Data processing units

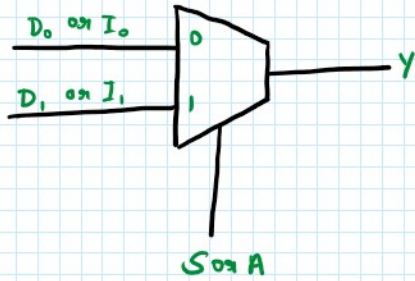
Select line / Control signal

Acts like a token system; controls which input goes through MUX at any time

MUX Input: Output

$2^N:1$ select lines

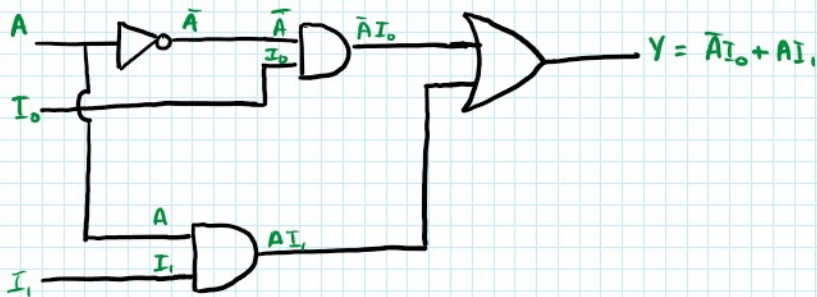
① 2:1 MUX



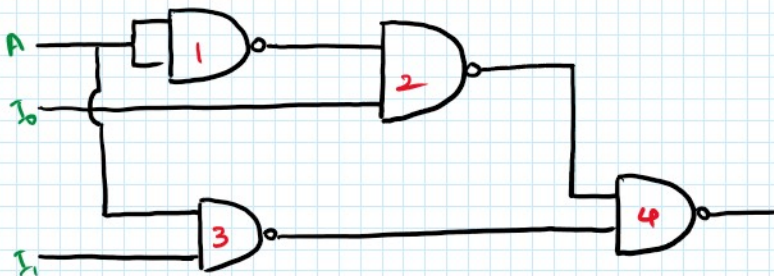
A	Y
0	I_0
1	I_1

$Y = \bar{A}I_0 + AI_1$	
If $Y = 0$, $Y = I_0$	If $Y = 1$, $Y = I_1$

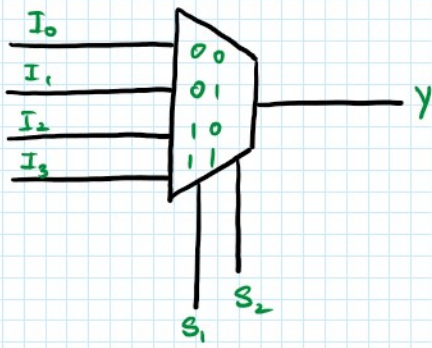
Using basic gates



Using NAND gates



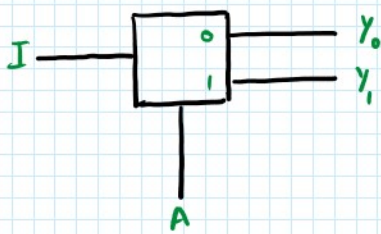
② 4:1 MUX



S_1	S_2	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Y = \bar{S}_1 \bar{S}_2 I_0 + \bar{S}_1 S_2 I_1 + S_1 \bar{S}_2 I_2 + S_1 S_2 I_3$$

③ 1:2 DEMUX

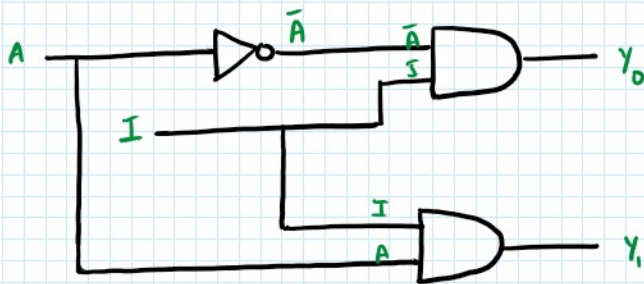


A	Y_0	Y_1
0	I	0
1	0	I

$$Y_0 = \bar{A}I$$

$$Y_1 = AI$$

Using basic gates



Using NAND gates

