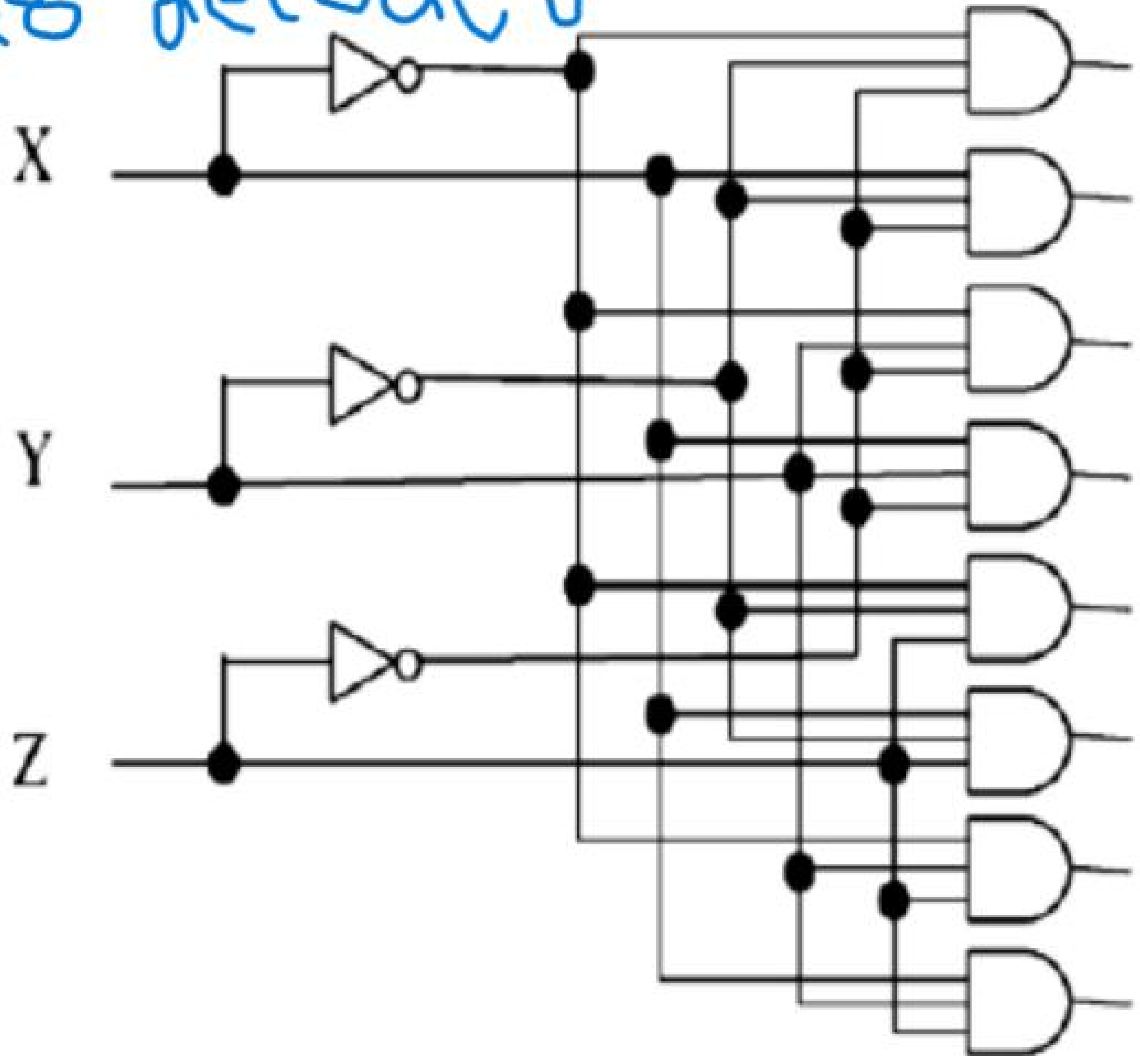
3x8 decoder



$D_0 = x'y'z'$

$D_1 = x'y'z$

$D_2 = x'yz'$

$D_3 = x'yz$

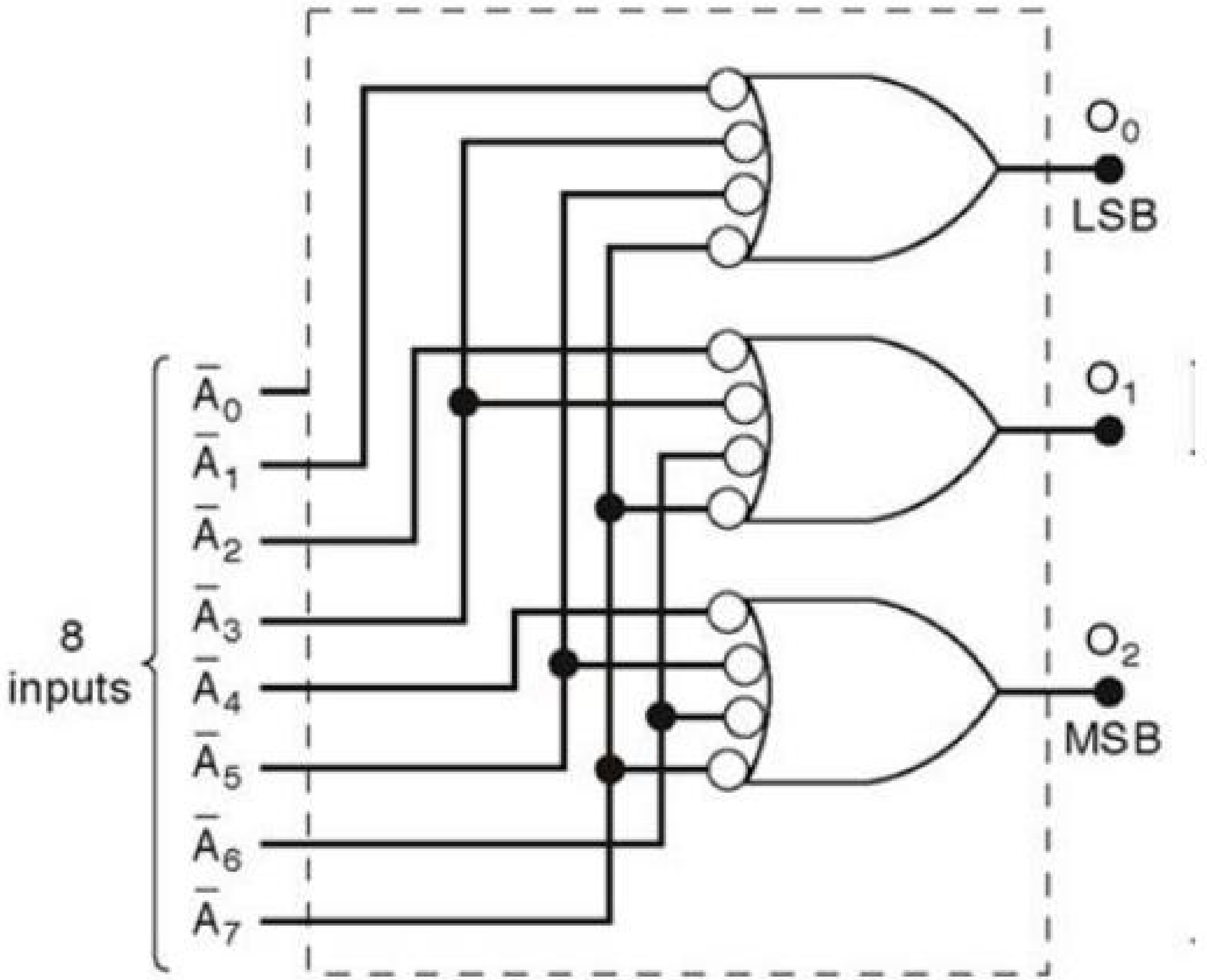
$D_4 = xy'z'$

$D_5 = xy'z$

$D_6 = xyz'$

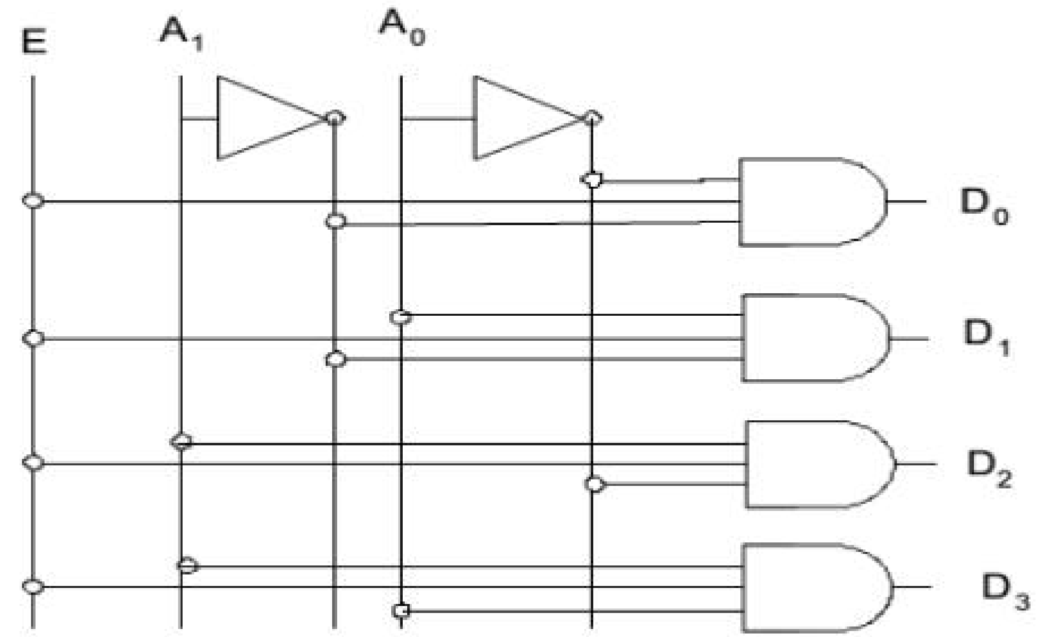
$D_7 = xyz$

Encoders



*Only one LOW input at a time

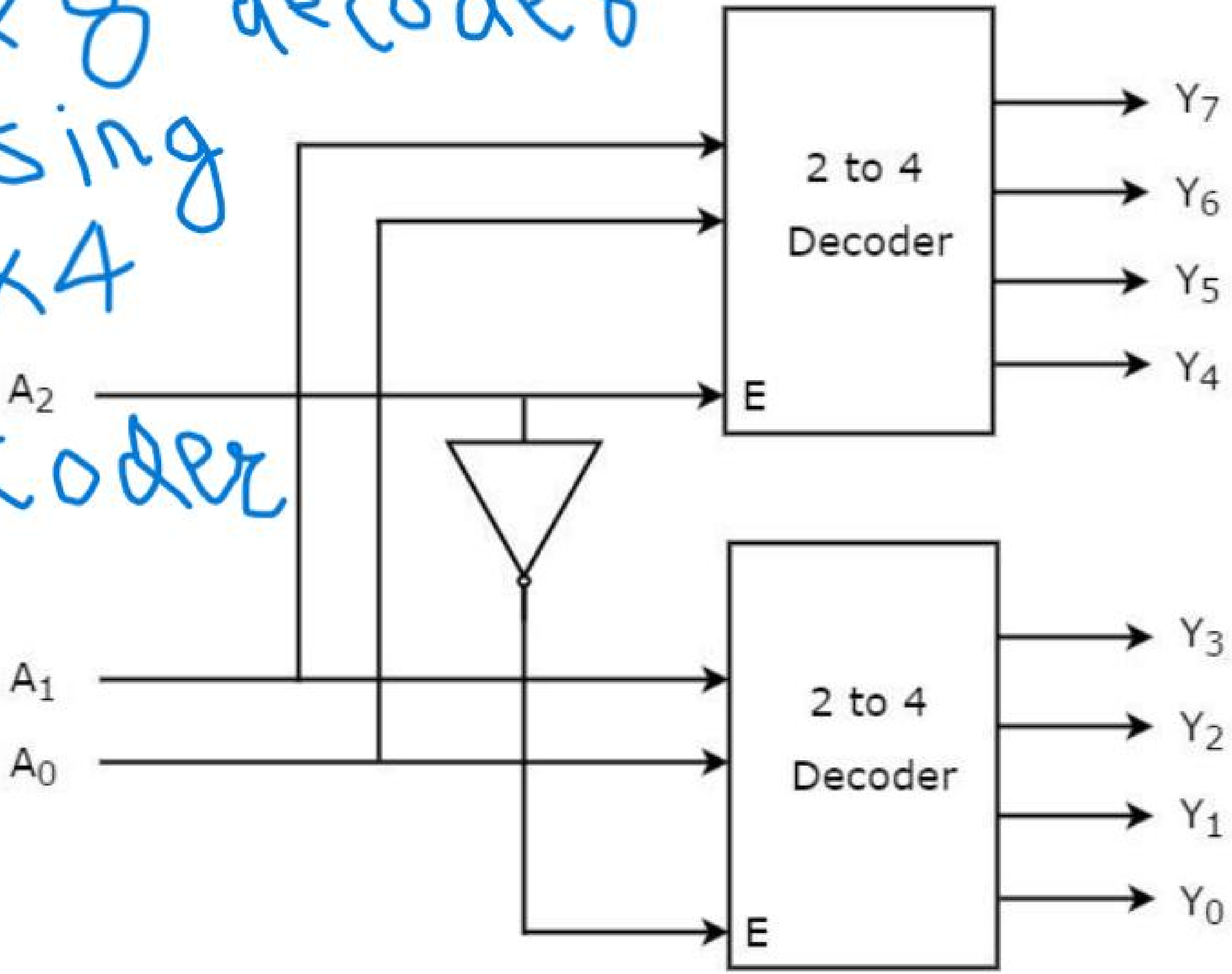
2-4 decoder with enable



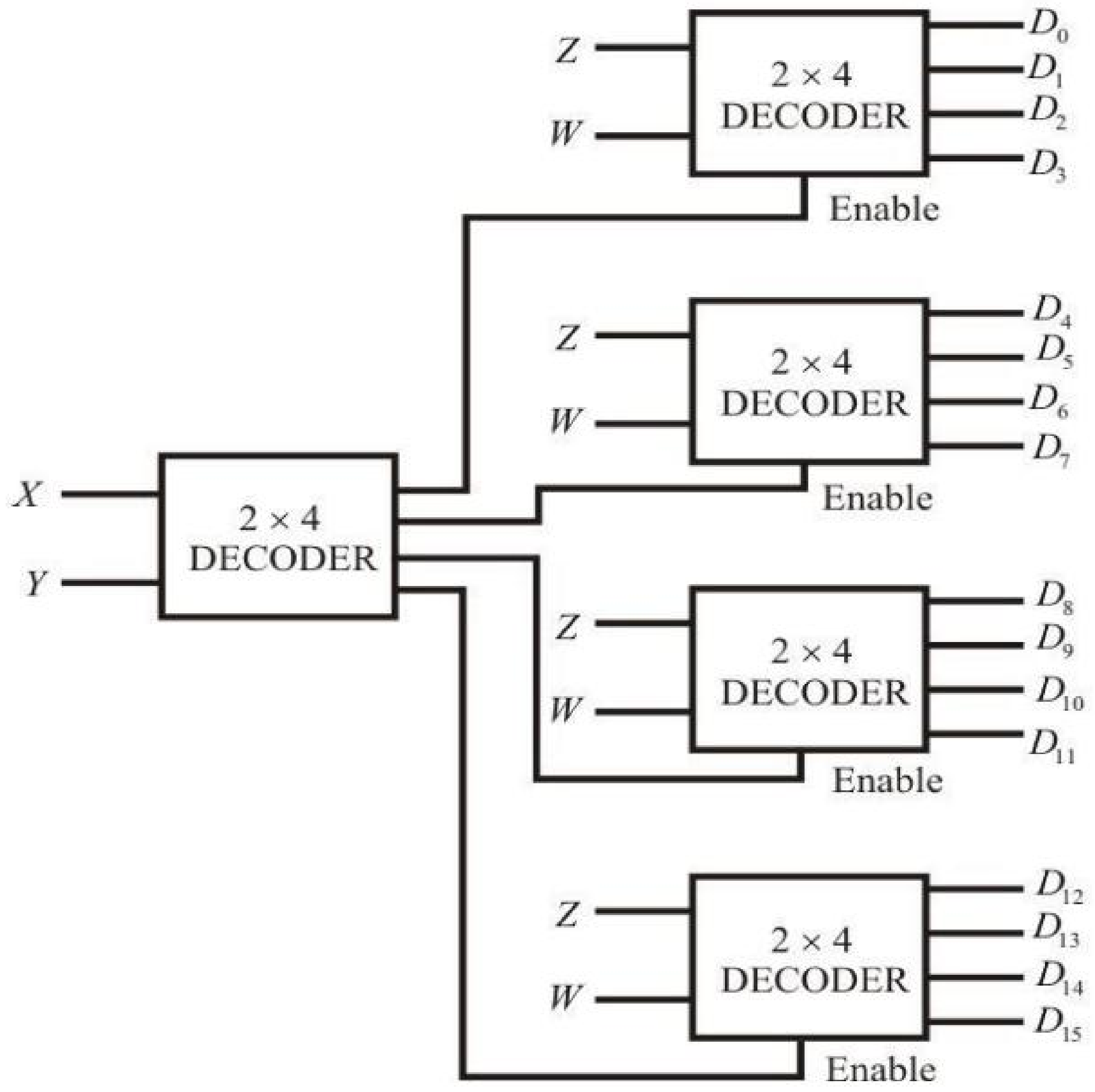
Implementation 2-to-4 decoder with enable

Decimal value	Enable	Inputs		Outputs			
	E	A_1	A_0	D_0	D_1	D_2	D_3
	0	X	X	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	0
3	1	1	1	0	0	0	1

3 x 8 decoder using 2x4 decoder

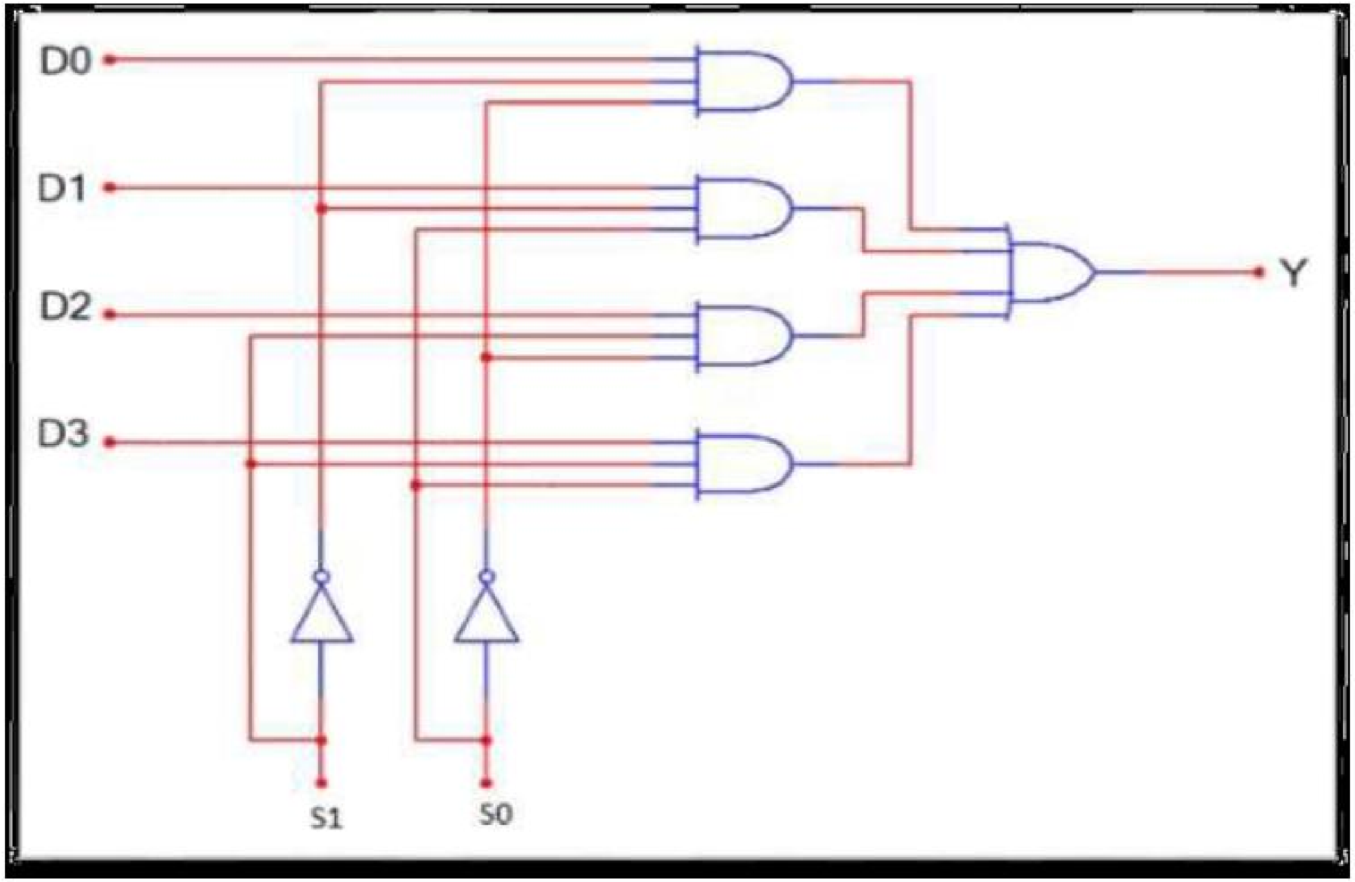


4 to 16 decoder using 2 to 4 decoder



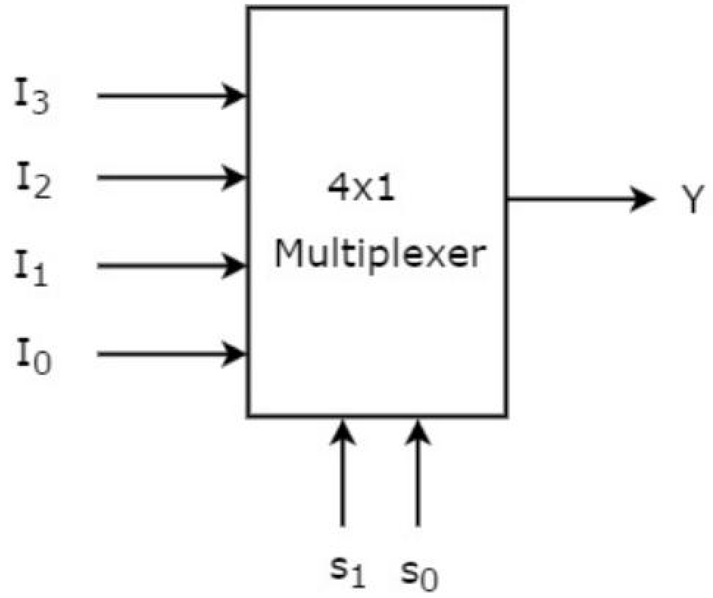
Multiplexer

- The logical level applied to the S input determines which AND gate i
  its data input passes through the OR gate to the output.
- The output, Y=S1’S0’D0+S1’S0D1+S1SO’D2+S1S0D3



Multiplexer

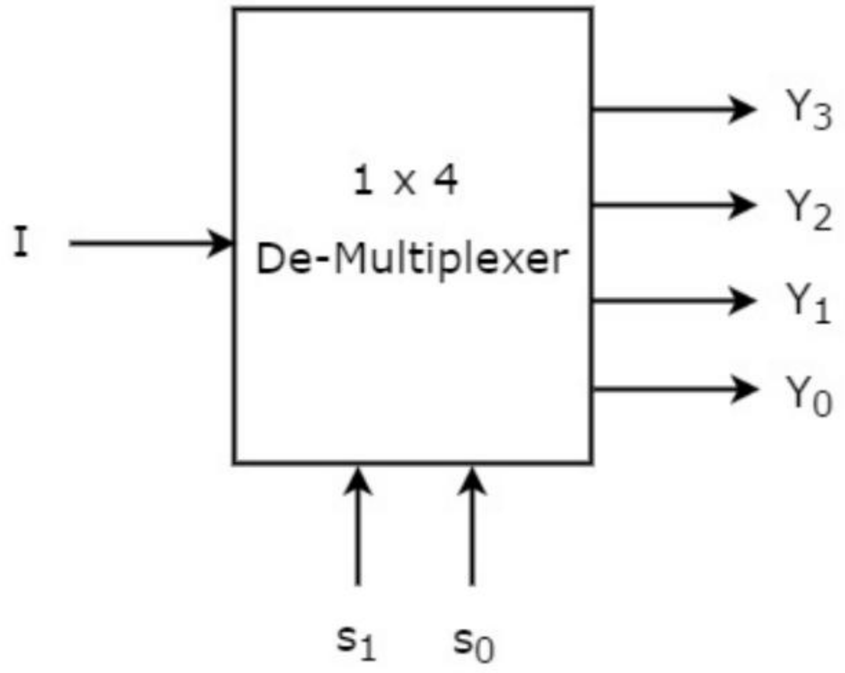
Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



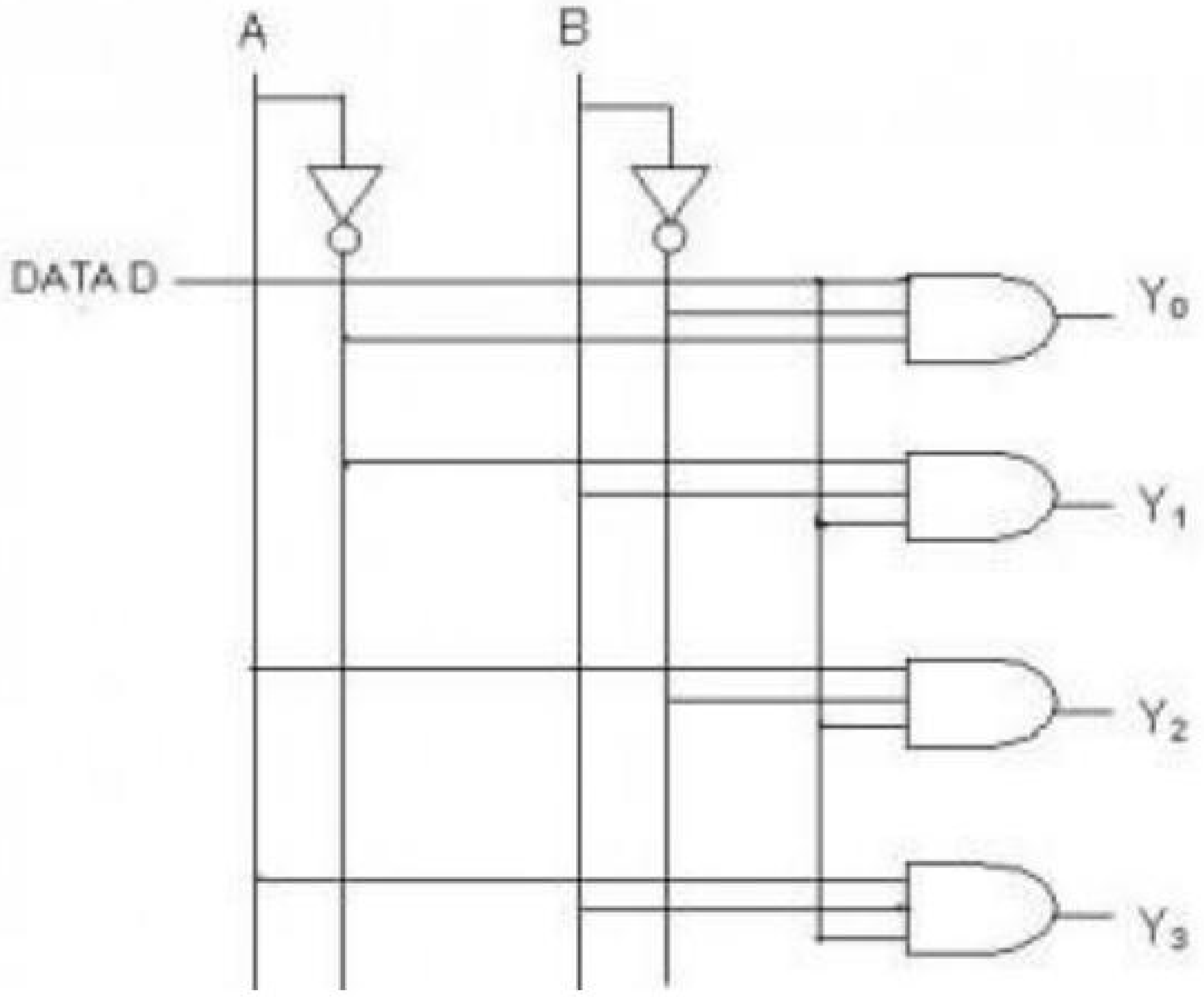
$$Y = S_1{}'S_0{}'I_0 + S_1{}'S_0I_1 + S_1S_0{}'I_2 + S_1S_0I_3$$

Demultiplexer

1-4 Demultiplexer : The 1-to-4 demultiplexer comprises 1- input bit, 4-output bits, and control bits. The 1x4 demultiplexer circuit diagram is shown below.



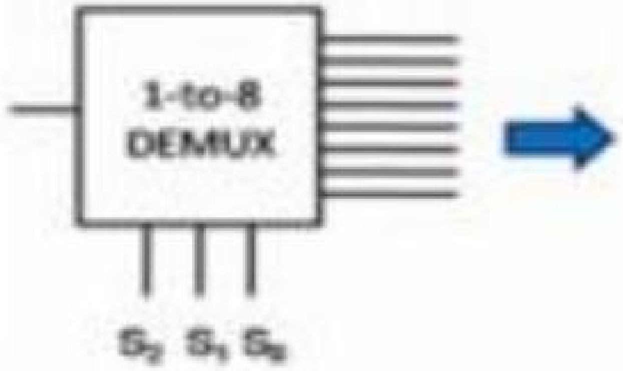
Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0



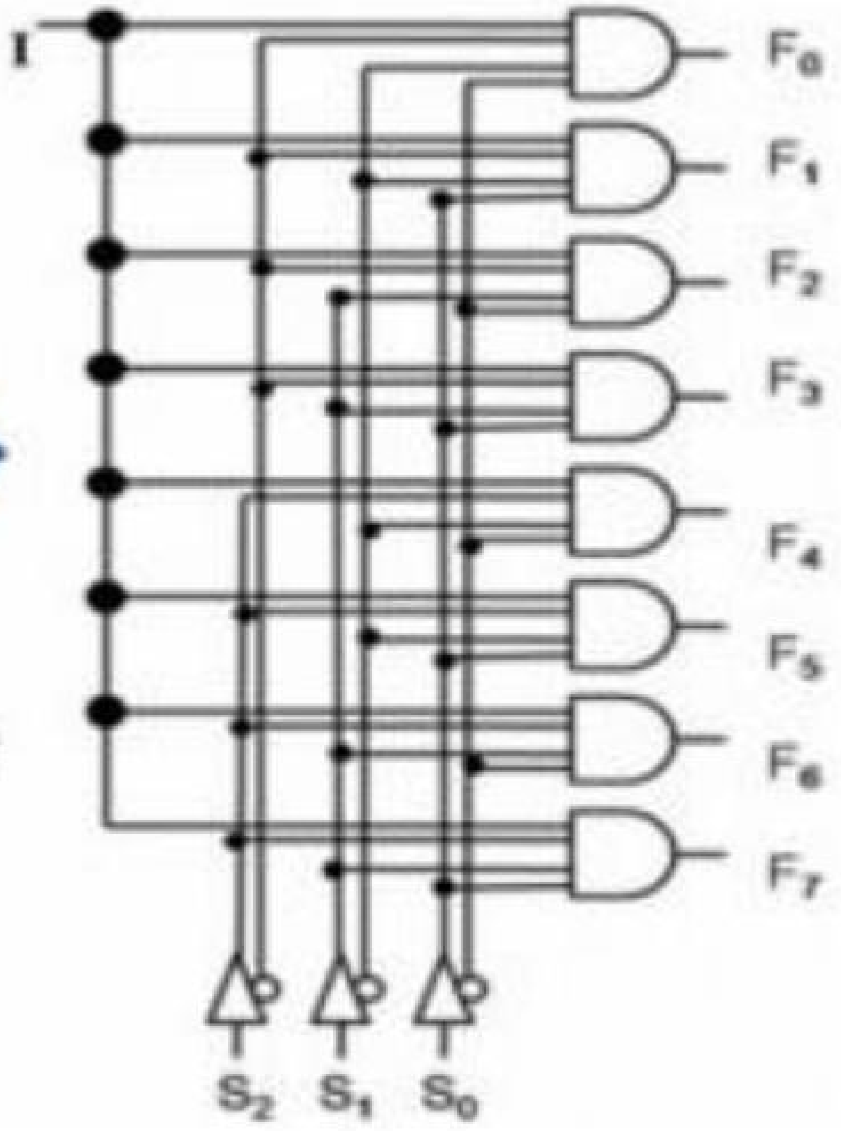
1X4 Demux

3 outputs.

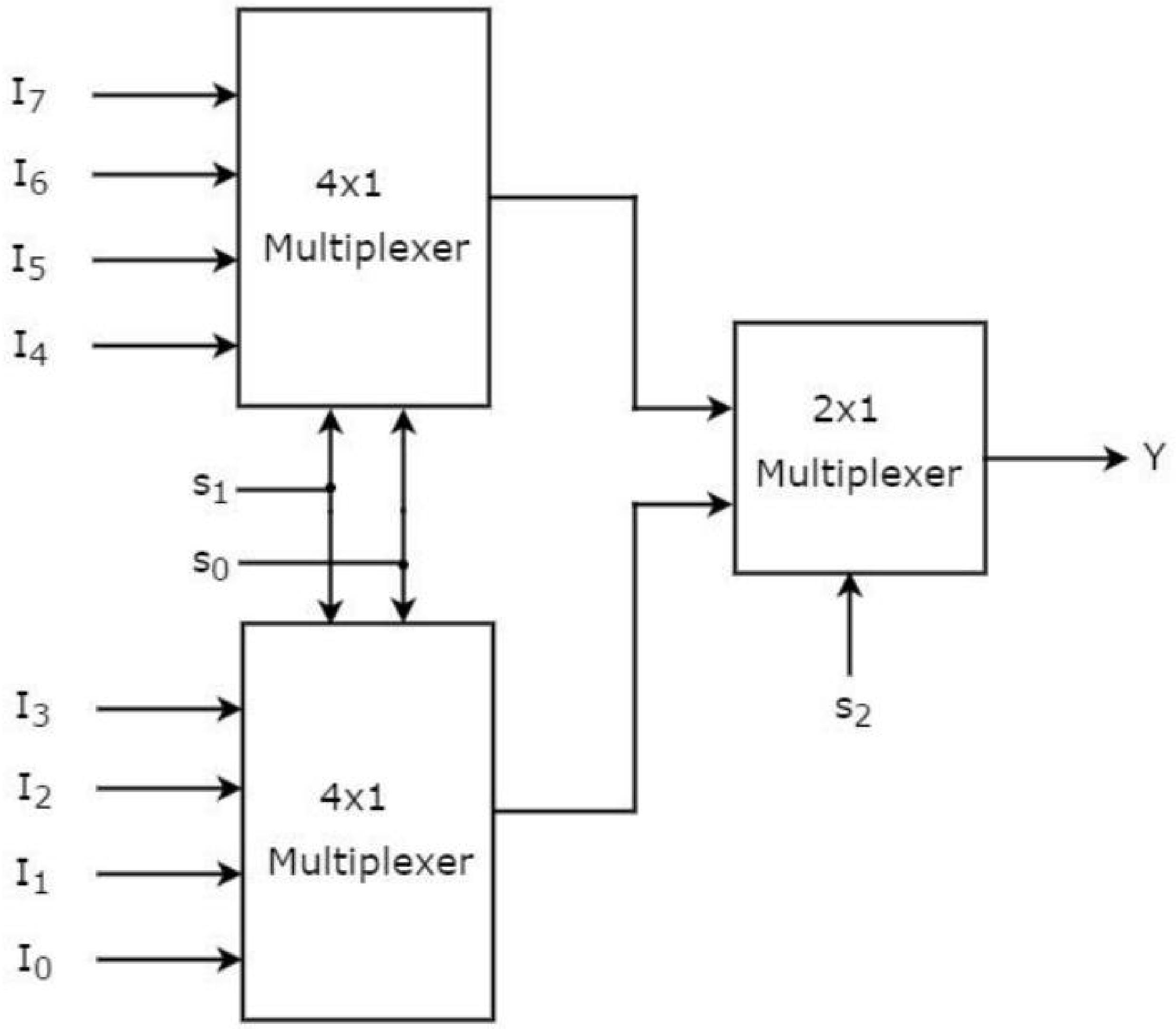
1X8 Demux



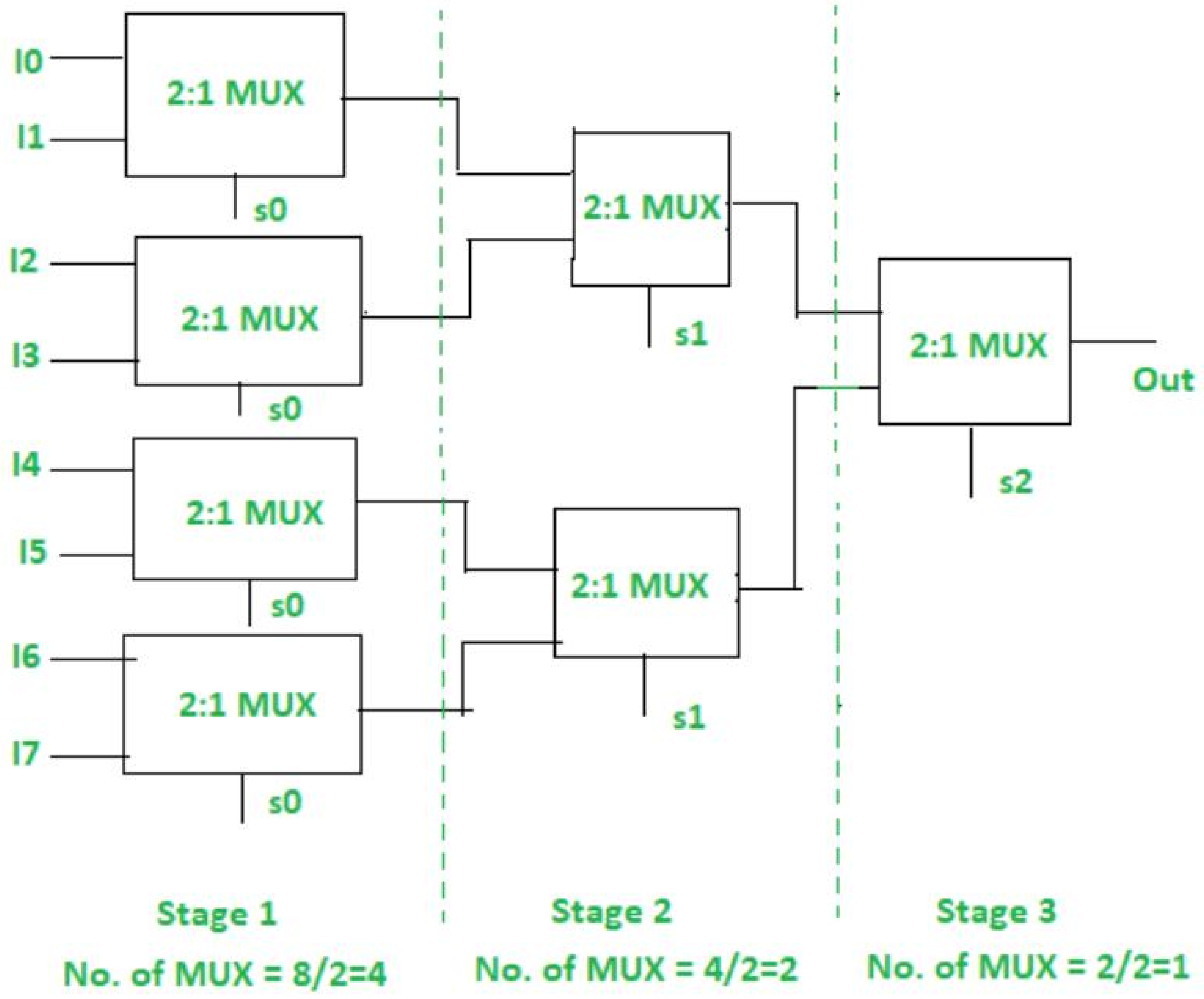
1-to-8 line demultiplexer



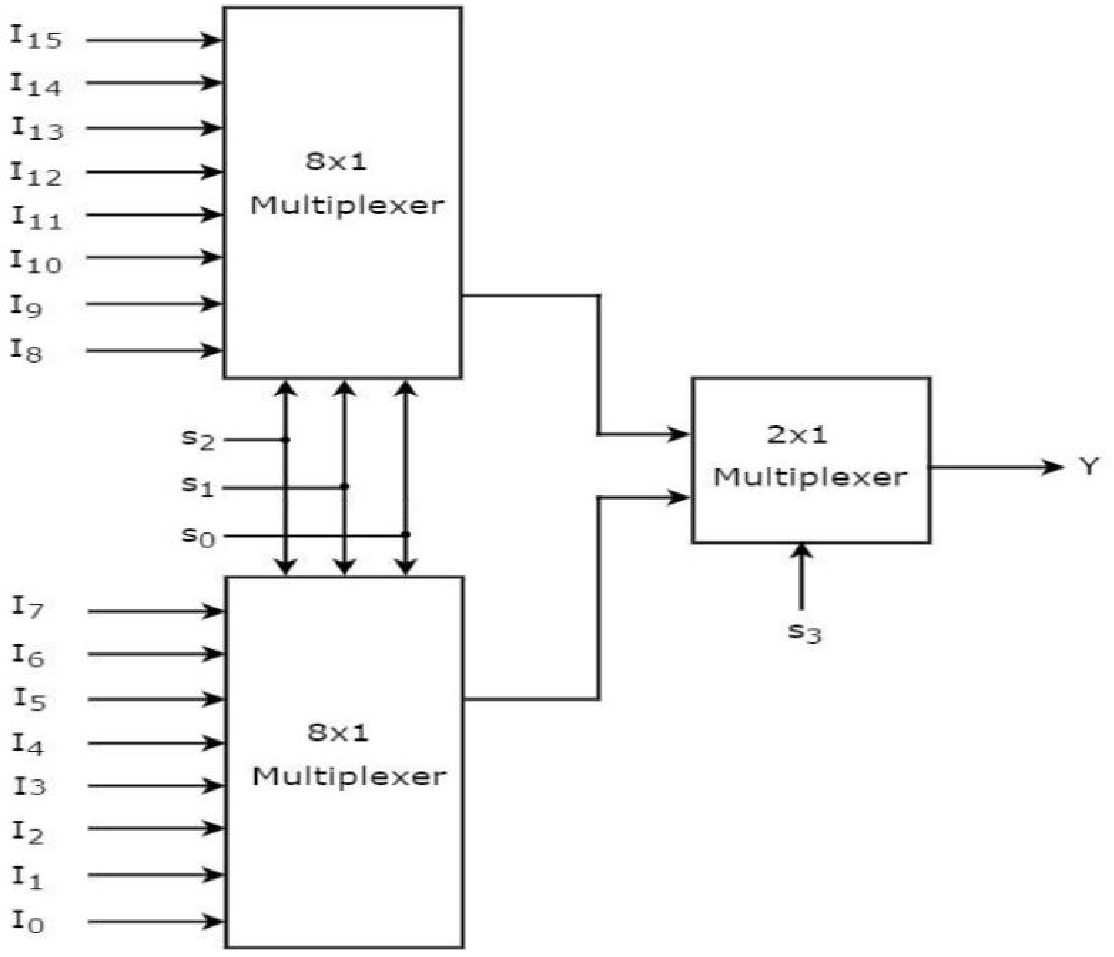
Implement 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer.



Implement 8x1 Multiplexer using 2x1 Multiplexer.



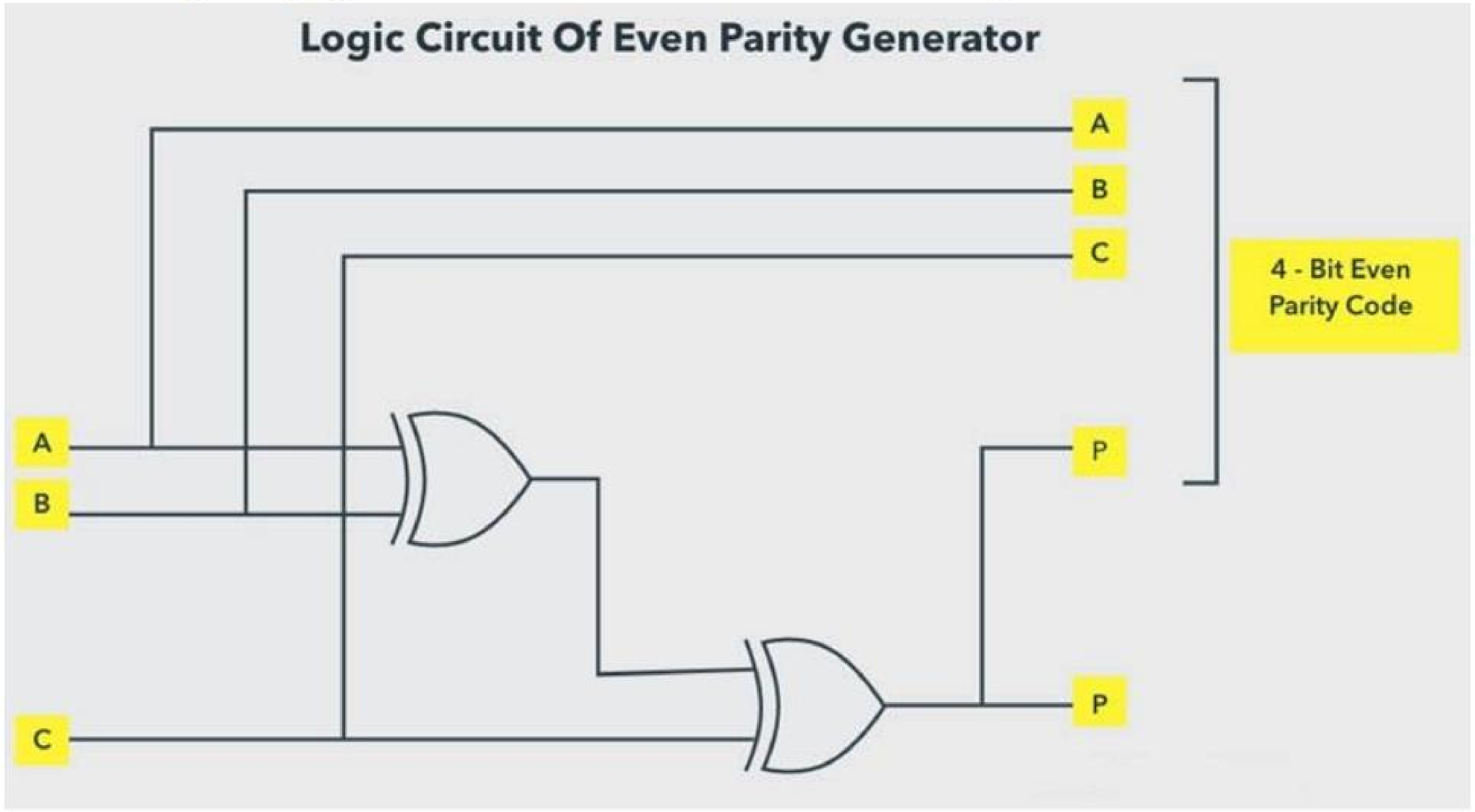
16x1 Multiplexer



Selection Inputs				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

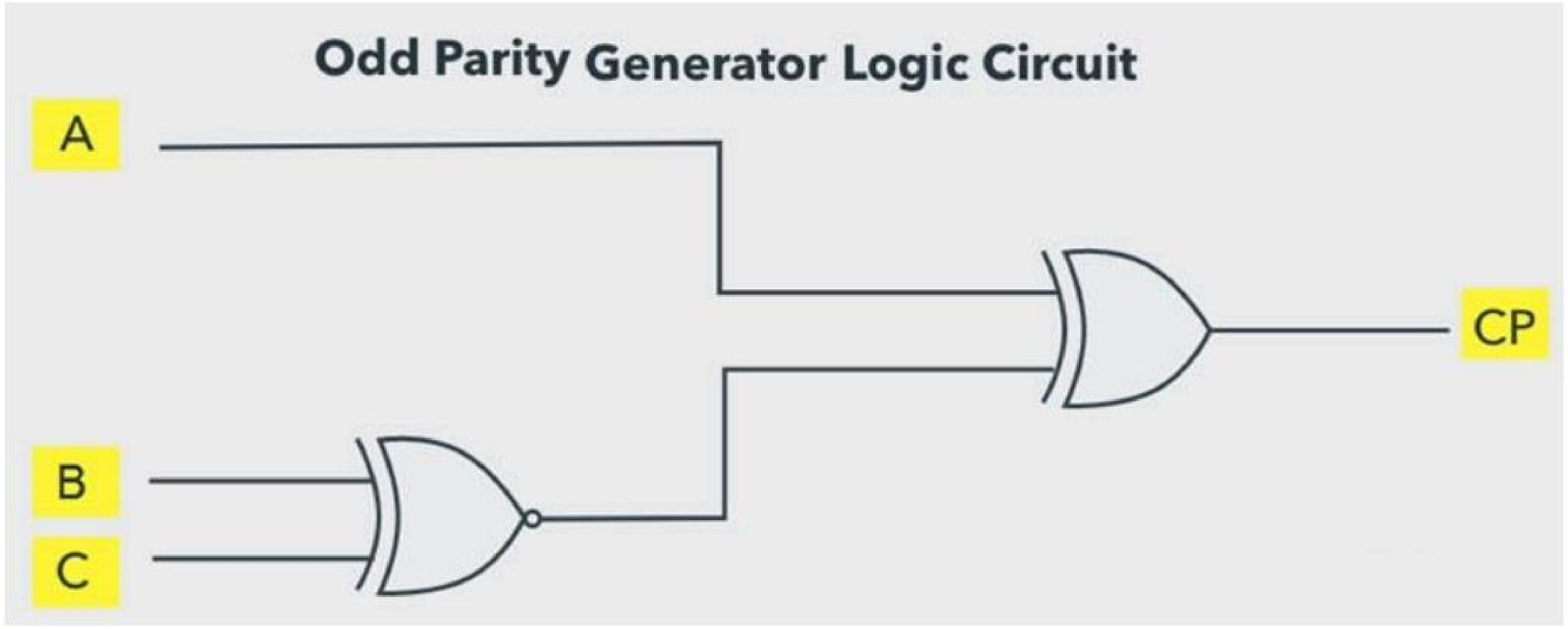
Even Parity Generator

- The above expression can be implemented by using two Ex-OR gates. The logic diagram of even parity generator with two Ex – OR gates is shown below.



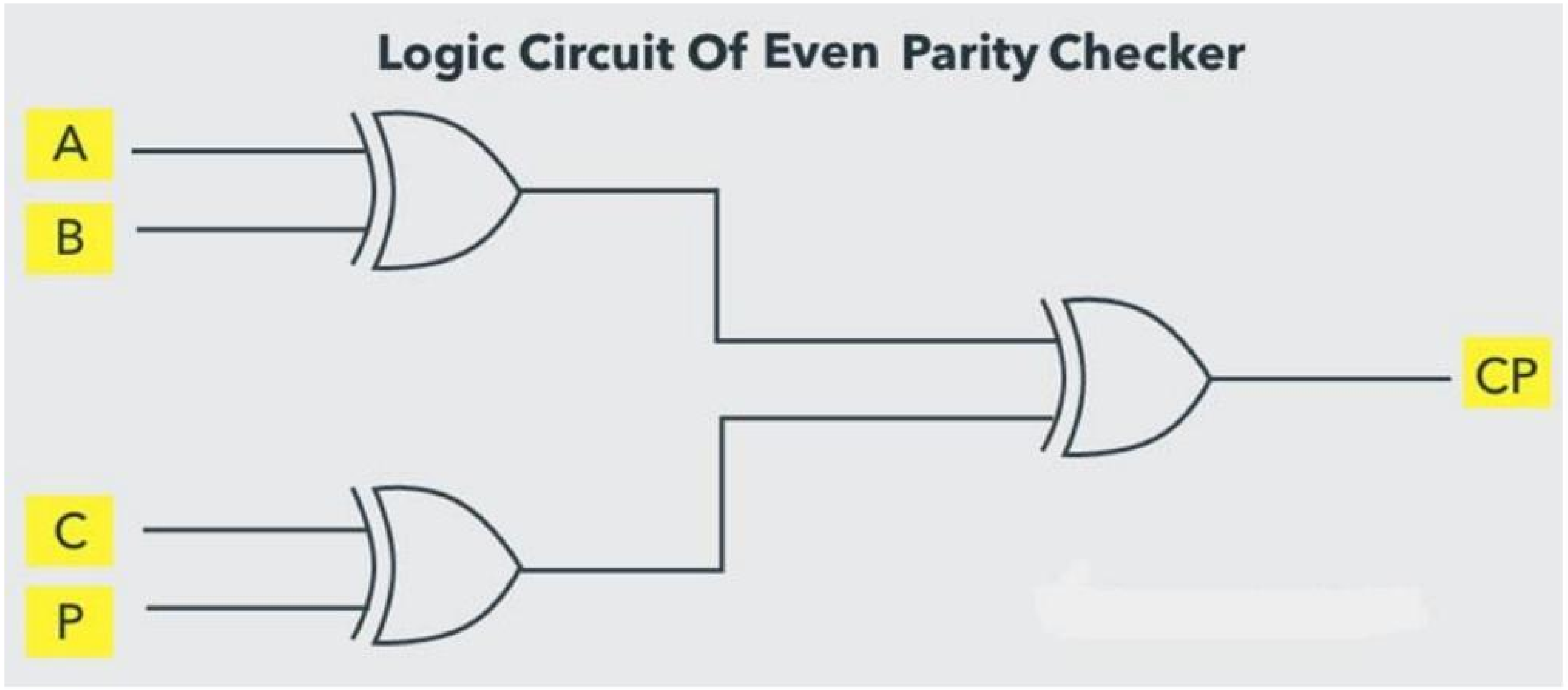
Odd Parity Generator

- The output parity bit expression for this generator circuit is obtained as
- $P = A \oplus (B \oplus C)'$



Even Parity Checker

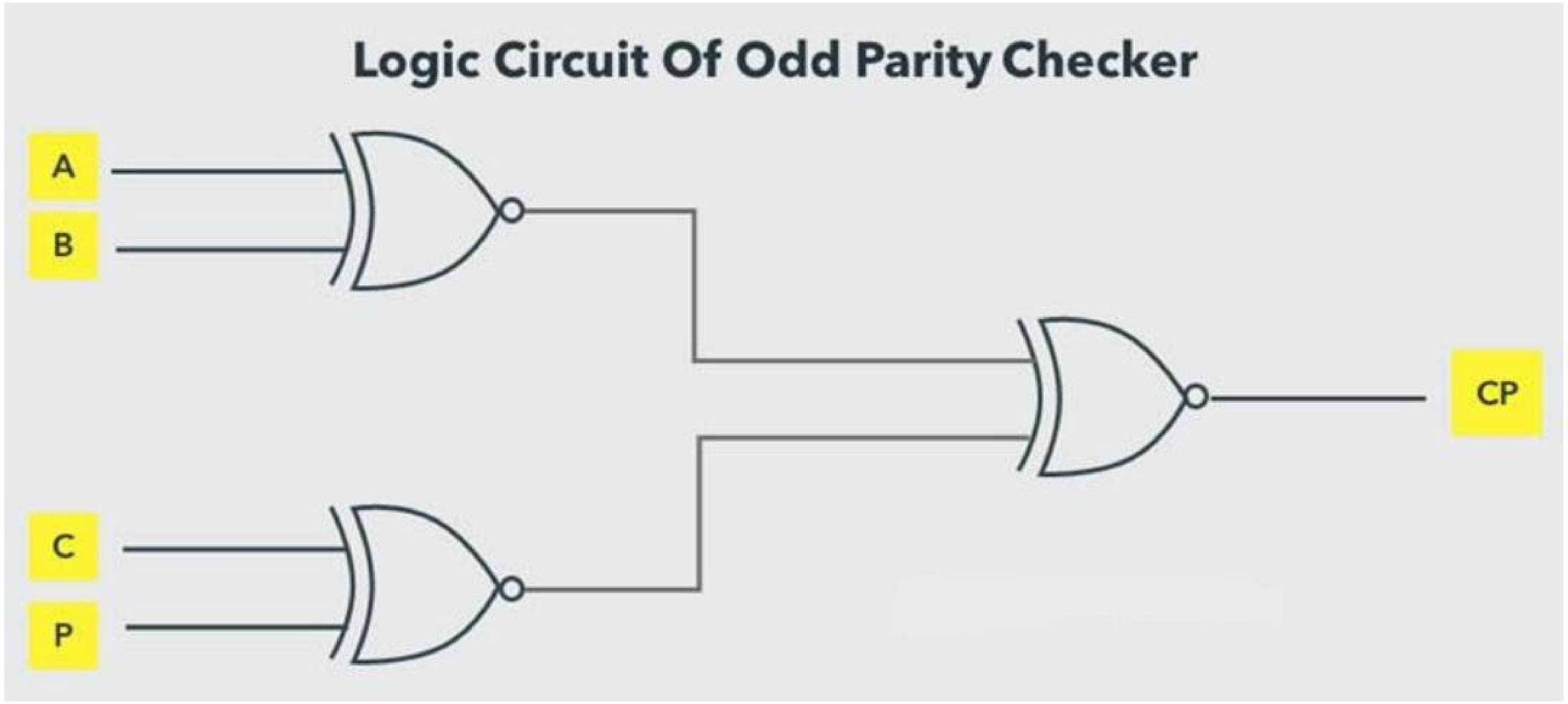
- The above logic expression for the even parity checker can be implemented by using three Ex-OR gates a shown in figure. If the received message consists of five bits, then one more Ex-OR gate is required for th even parity checking.



Odd Parity Checker

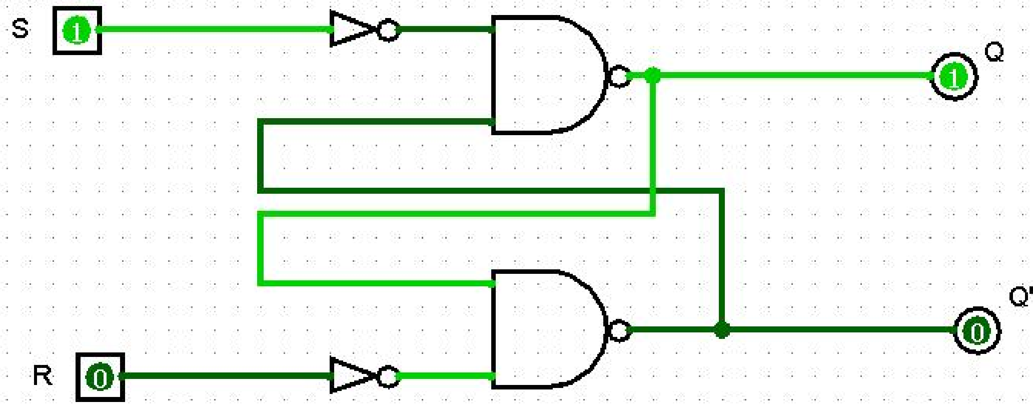
➢ After simplification, the final expression for the PEC is obtained as

PEC = (A Ex-NOR B) Ex-NOR (C Ex-NOR P)

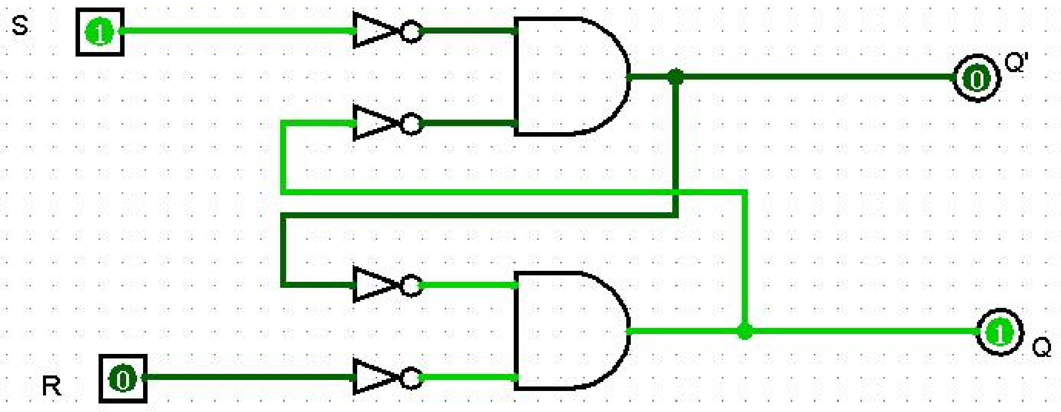


SR LATCHES USING DIFFERENT GATES

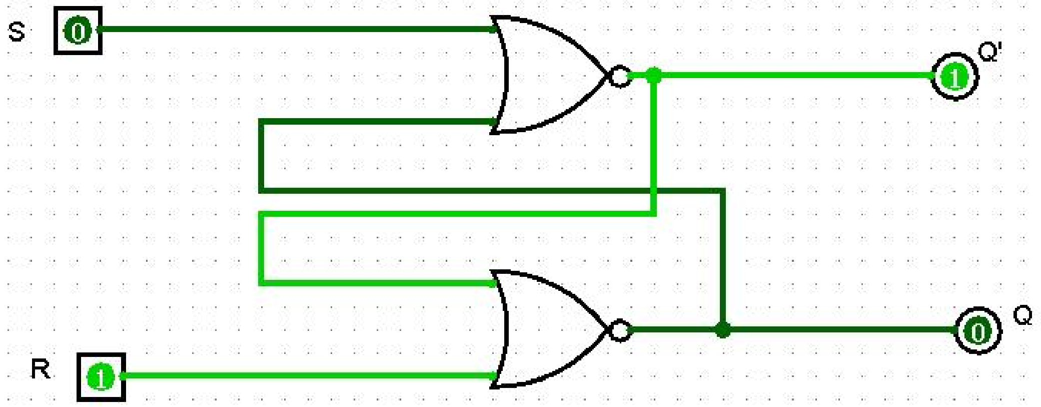
D LATCH OR GATED LATCHES WITH DIFFERENT GATES CAN BE APPLIED IN A SIMILAR WAY



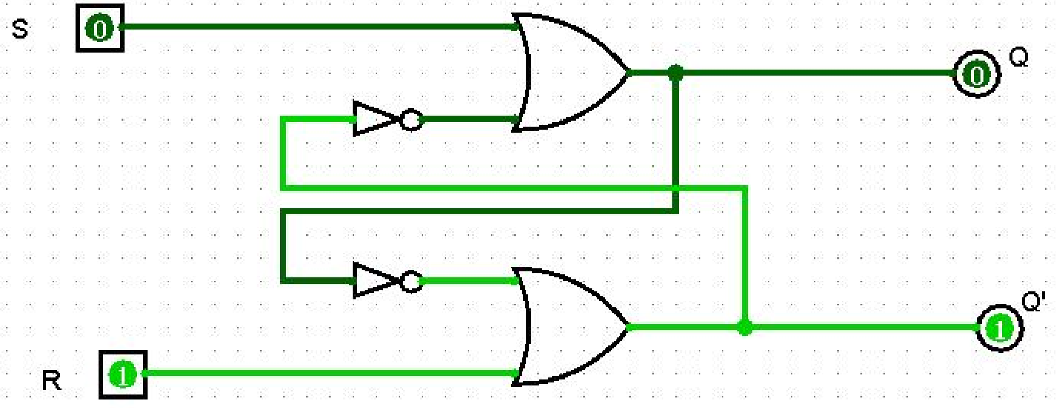
NAND



AND



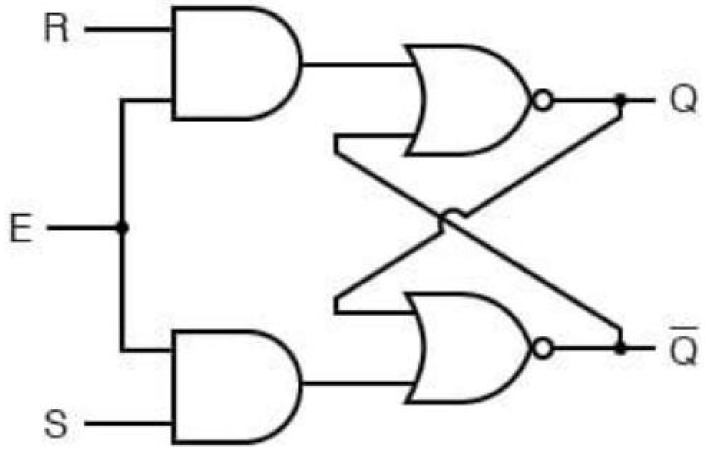
NOR



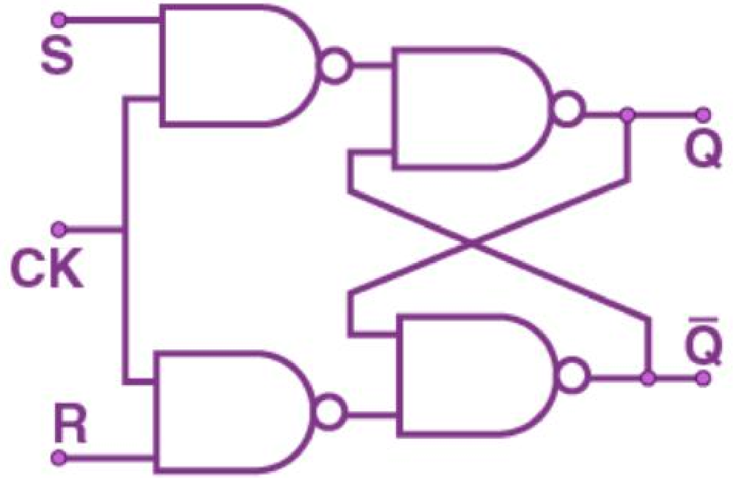
OR

Gated SR Latch

- A Gated SR latch (clocked SR Latch) is a SR latch with enable input which works when enable is 1 and retain the previous state when enable is 0.



E	S	R	Q	$\overline{Q}$
0	0	0	latch	latch
0	0	1	latch	latch
0	1	0	latch	latch
0	1	1	latch	latch
1	0	0	latch	latch
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

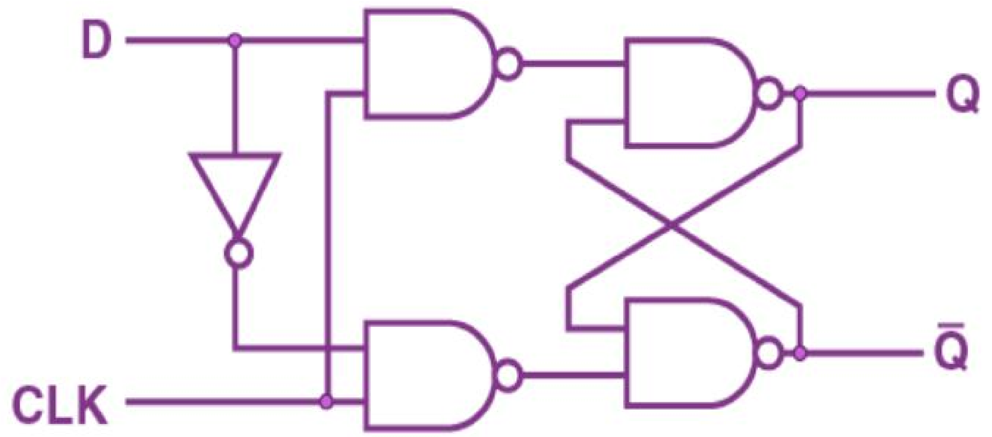


S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Characteristics Equation for SR Flip Flop: $Q_{N+1} = Q_N R' + S$

D Flip-Flop

D flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal. That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal.

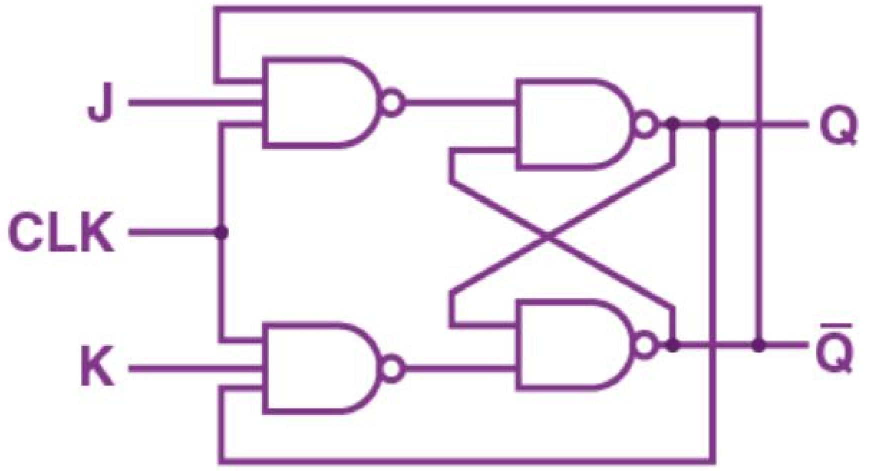


Q	D	$Q_{(t+1)}$
0	0	0
0	1	1
1	0	0
1	1	1

Characteristics Equation for D Flip Flop: $Q_{t+1} = D$

JK Flip-Flop

JK flip-flop is the modified version of SR flip-flop. It operates with only positive clock transitions or negative clock transitions.

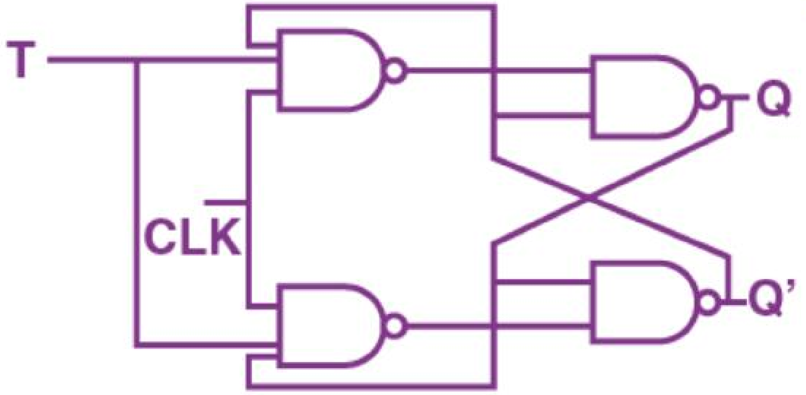


J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristics Equation for JK Flip Flop: $Q_{N+1} = JQ_N' + K' Q_N$

T Flip-Flop

T flip-flop is the simplified version of JK flip-flop. It is obtained by connecting the same input ‘T’ to both inputs of JK flip-flop. It operates with only positive clock transitions or negative clock transitions.



T	Q_N	Q_{N+1}
0	0	0
0	1	1
1	0	1
1	1	0

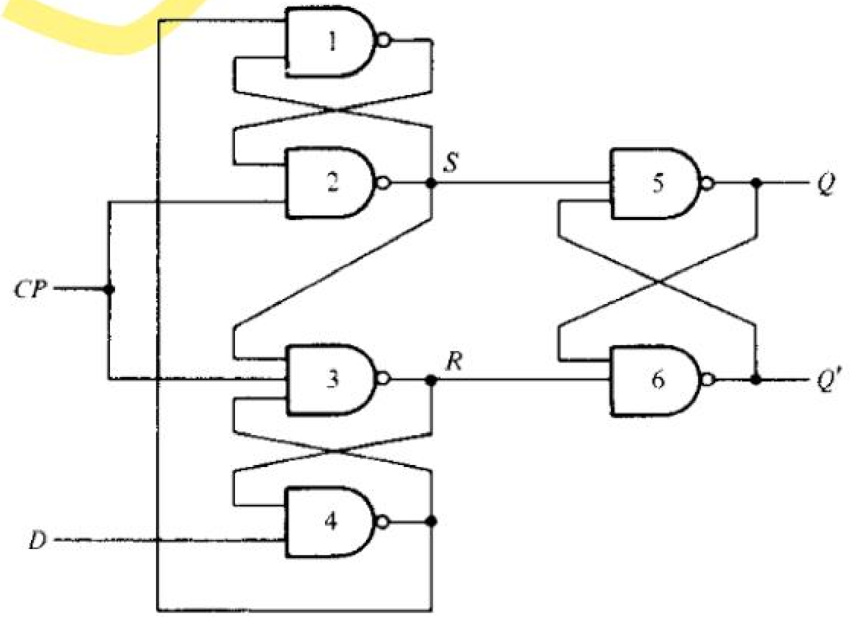
Characteristics Equation for JK Flip Flop: $Q_{N+1} = TQ_N' + T' Q_N = T \oplus Q_N$

Conversion for Flip-Flops

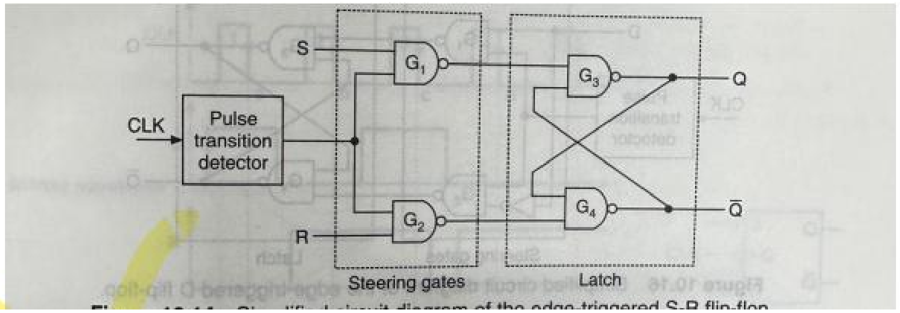
EXCITATION TABLE

Q_N	Q_{N+1}	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

Edge-Triggered D Flip-Flop



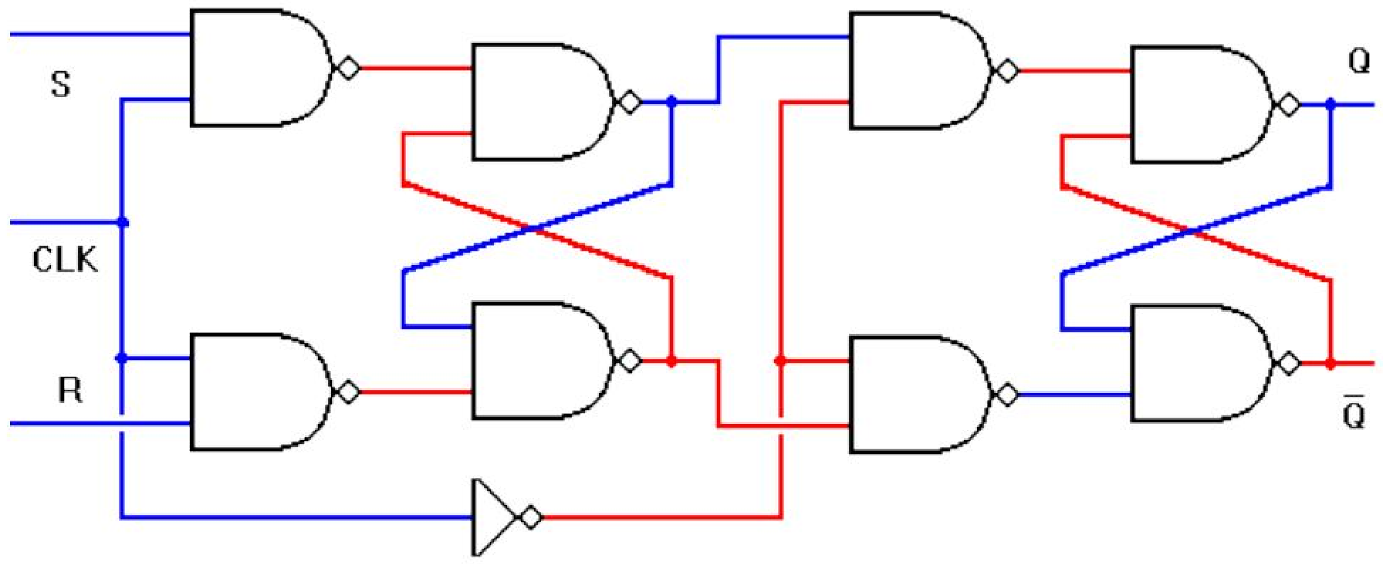
- CP = 0 => S & R = 1 => STEADY STATE OUTPUT
- D = 0 & CP = 1 => S = 1, R = 0 => Q = 0
- D = 1 & CP = 1 => S = 0, R = 1 => Q = 1

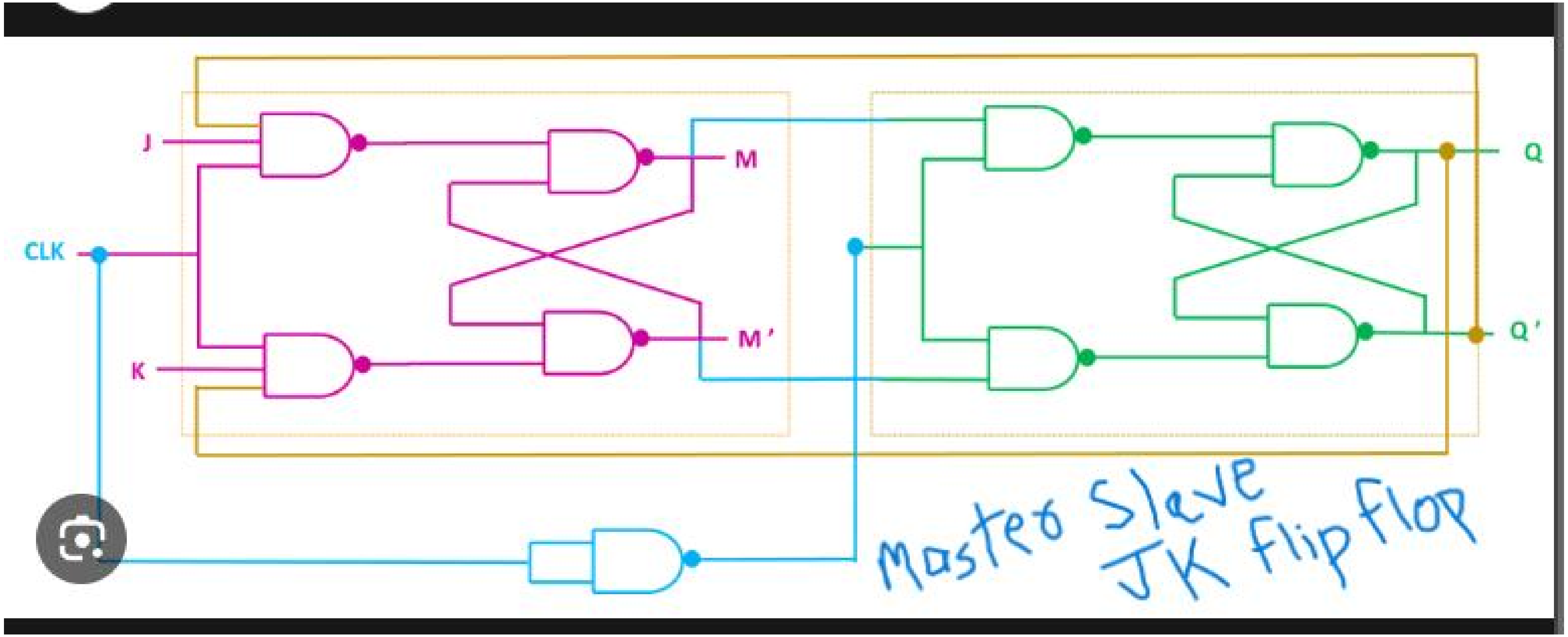


Edge-Triggered SR Flip-Flop

- To adjust the clocked RS latch for edge triggering, we must actually combine two identical clocked latch circuits, but have them operate on opposite halves of the clock signal.

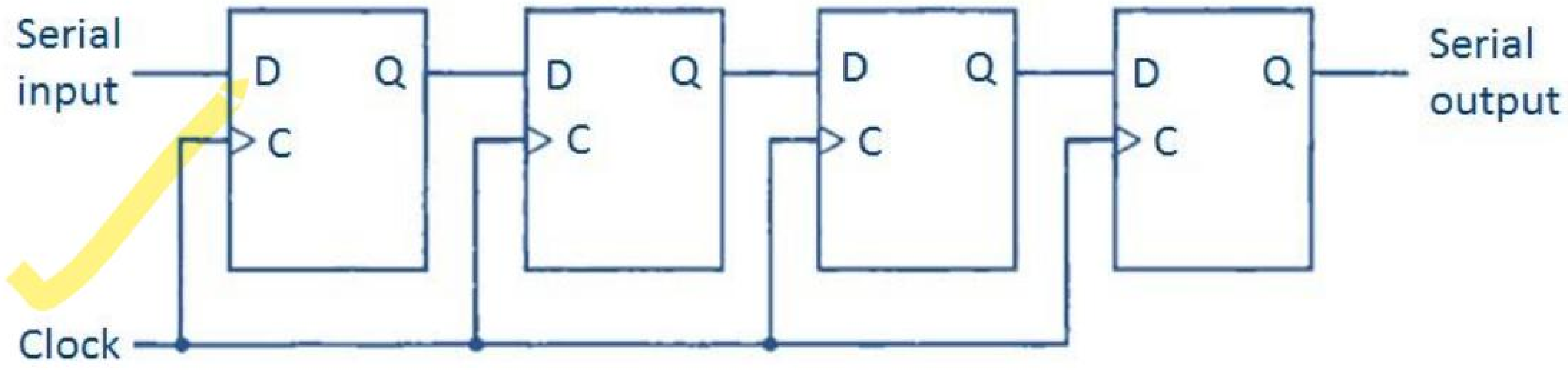
- The edge-triggered SR NAND flip-flop is shown below.




J
K
CLK
M
M'
Q
Q'
Master Slave
JK flip flop

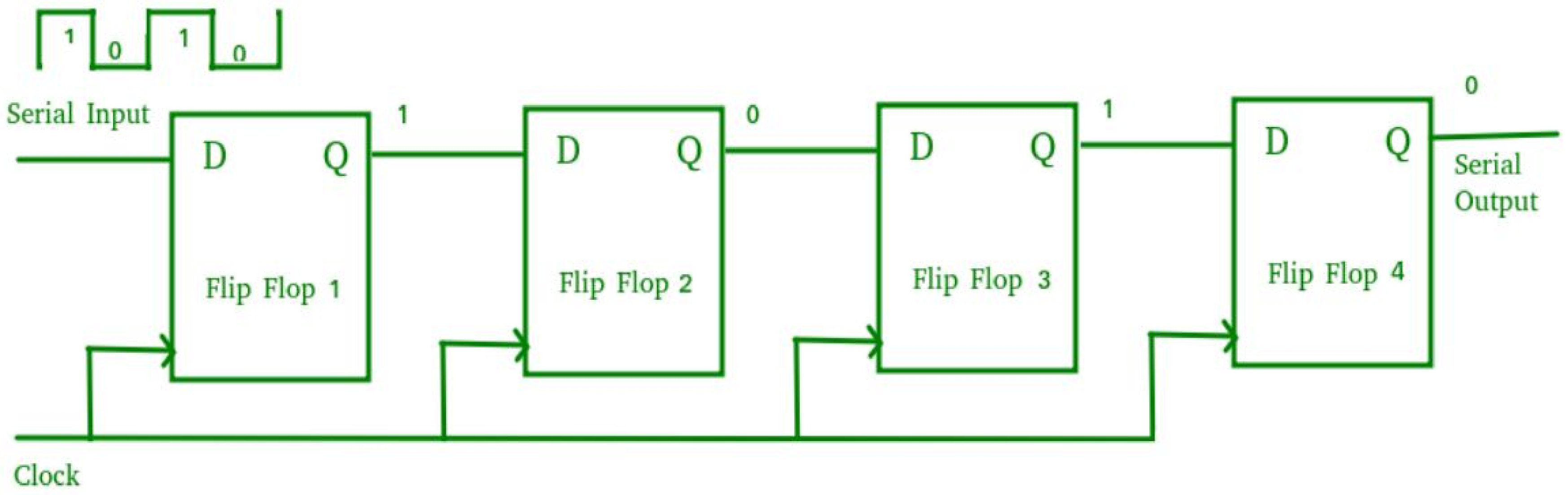
4-bit shift register using d flip flop

SERIAL INPUT SERIAL OUTPUT



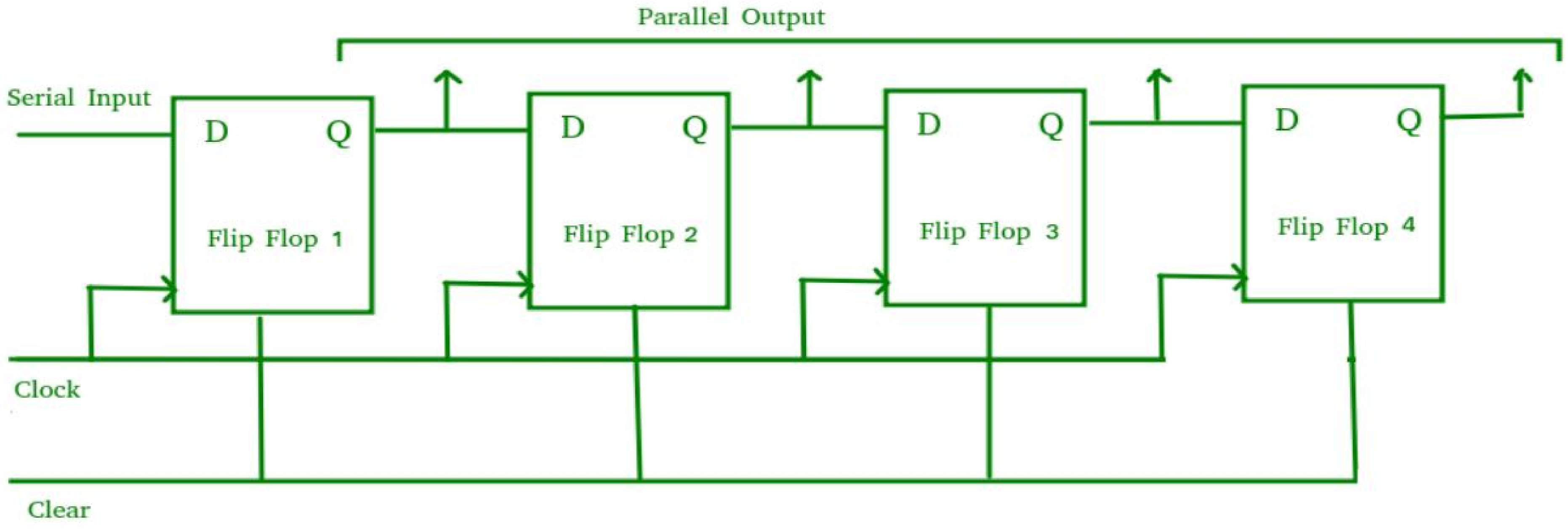
Serial-In Serial-Out Shift Register (SISO)

- The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as a Serial-In Serial-Out shift register.
- Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register.
- The main use of a SISO is to act as a delay element.

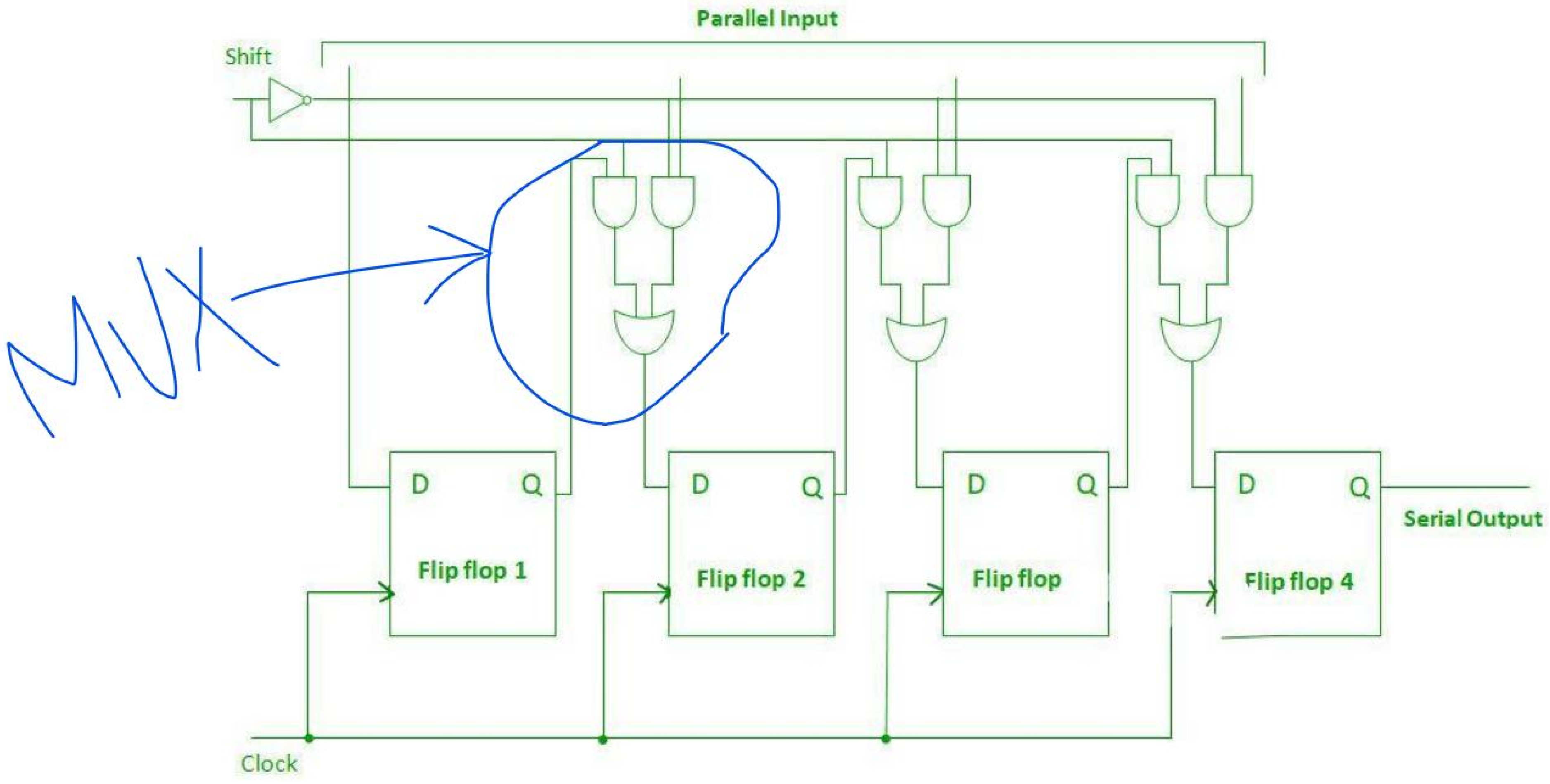


Serial-In Parallel-Out Shift Register (SIPO)

- The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as the Serial-In Parallel-Out shift register
- Used in communication lines where demultiplexing of a data line into several parallel lines is required because the main use of the SIPO register is to convert serial data into parallel data.

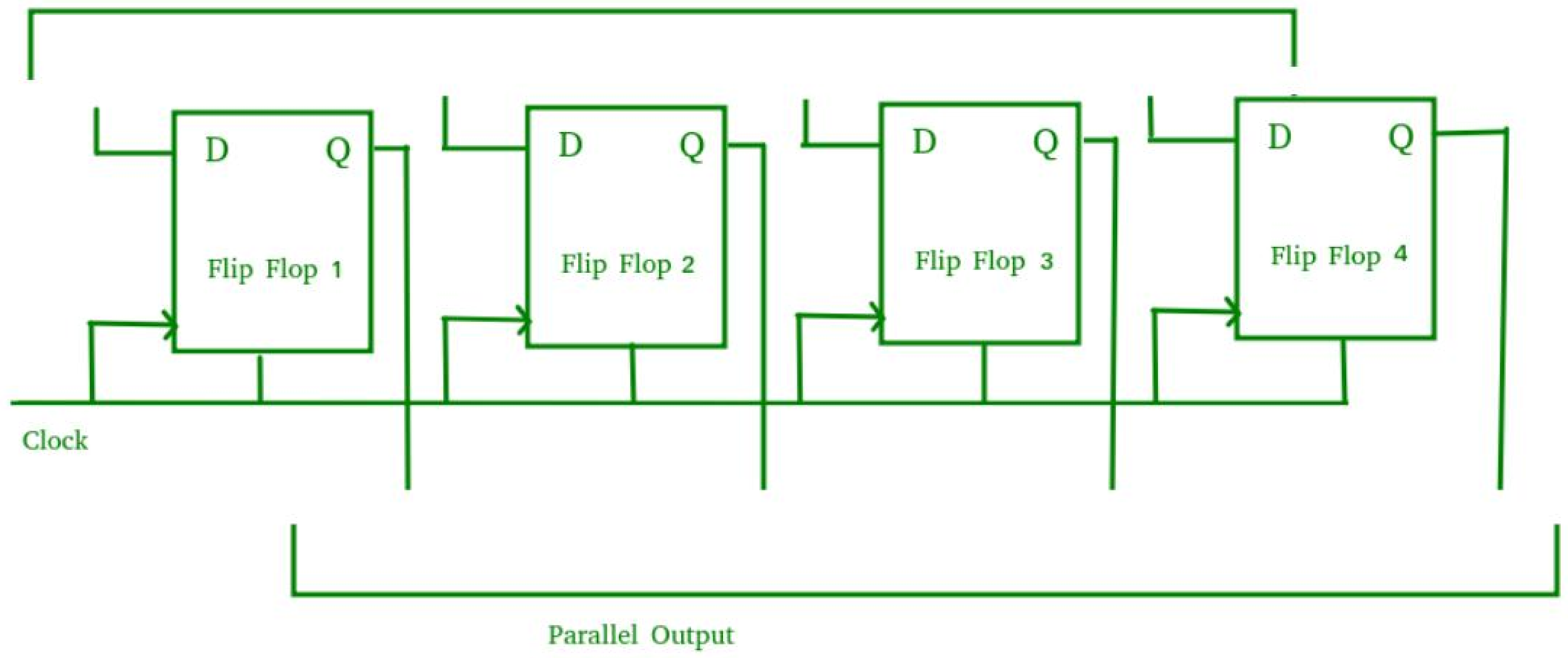


Parallel-In Serial-Out Shift Register (PISO)

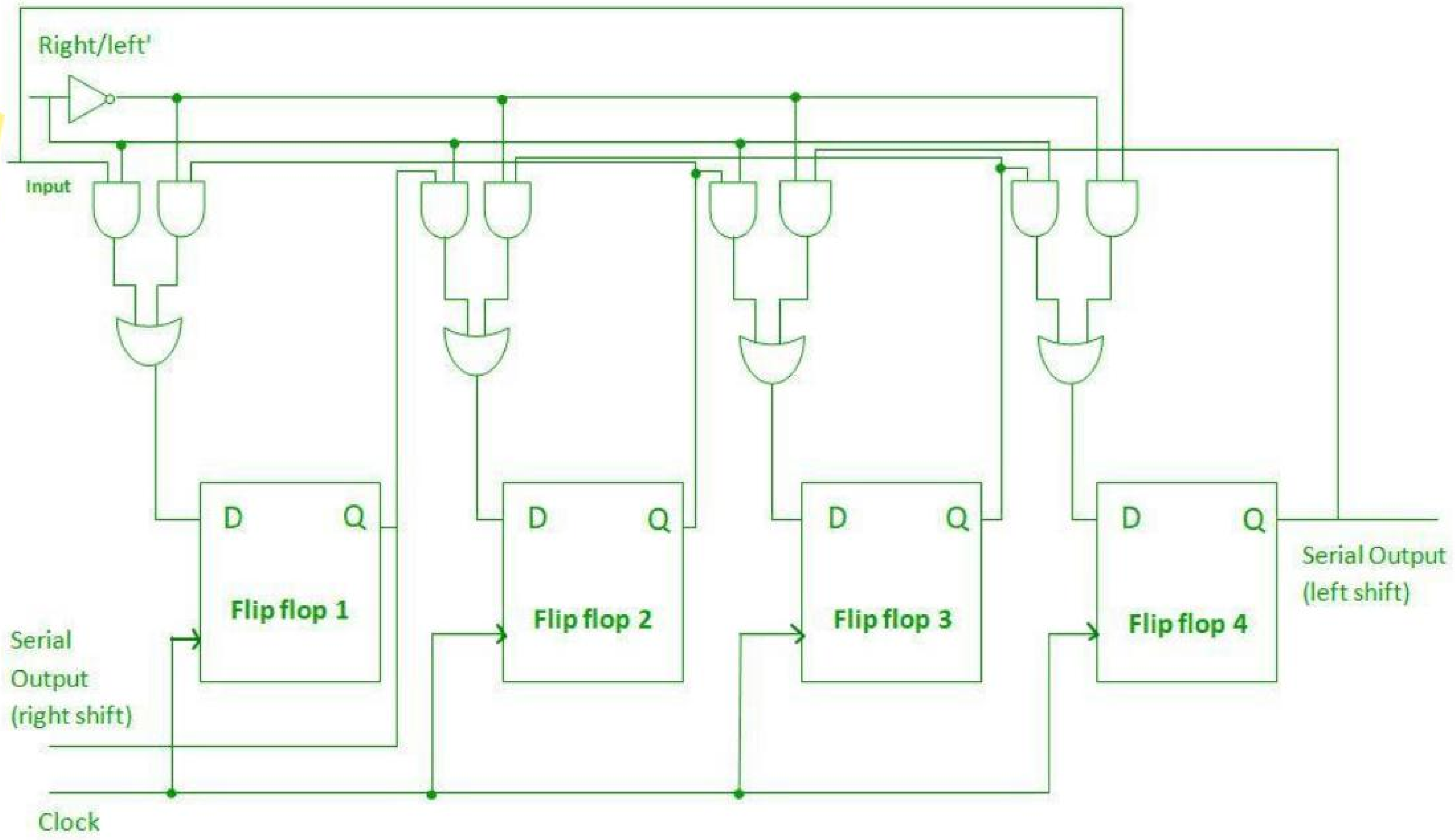


Parallel-In Parallel-Out Shift Register (PIPO)

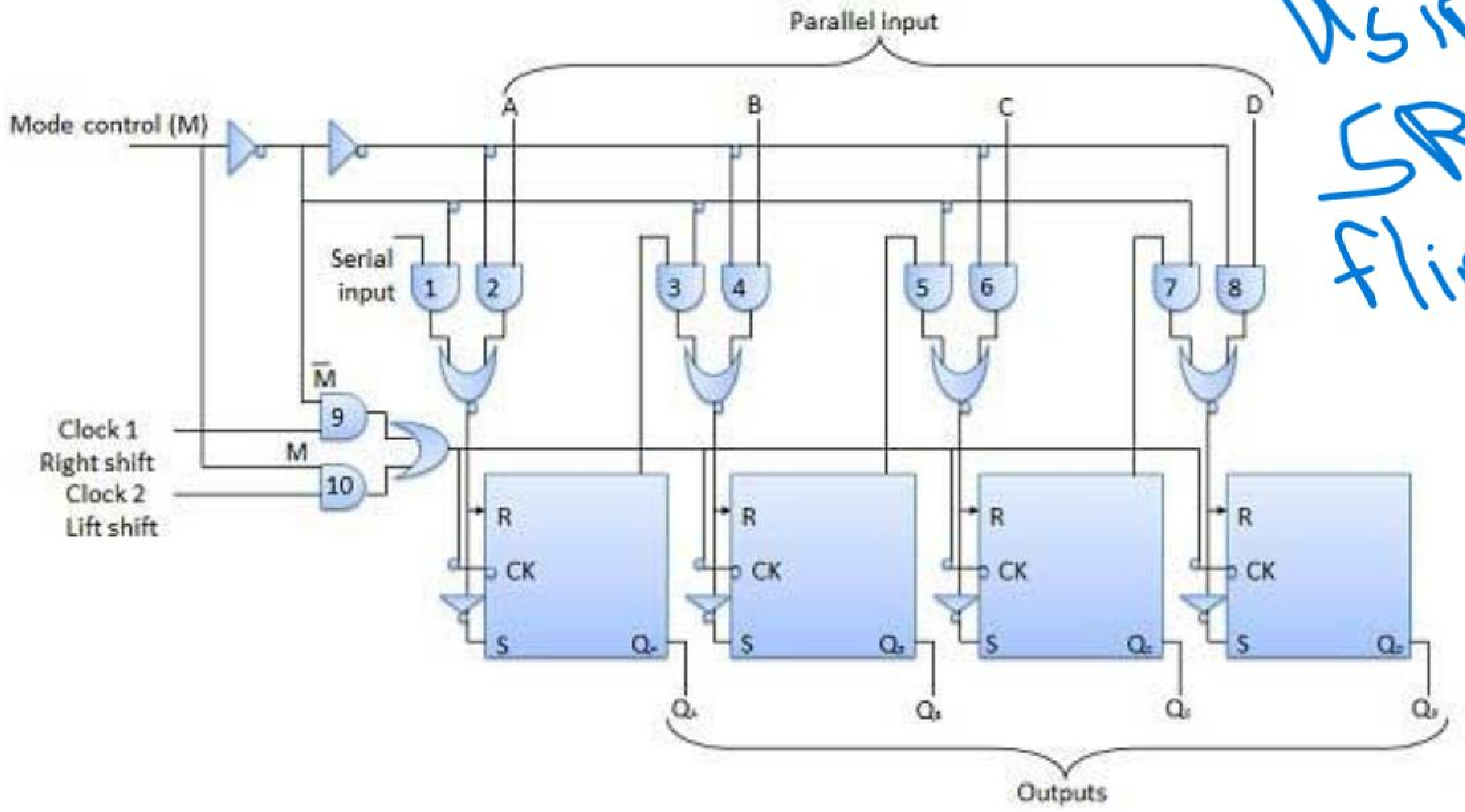
- The shift register, which allows parallel input and also produces a parallel output is known as Parallel-In parallel-Out shift register.
- A Parallel in Parallel out (PIPO) shift register is used as a temporary storage device and like SISO Shift register it acts as a delay element.



Bidirectional Shift Register



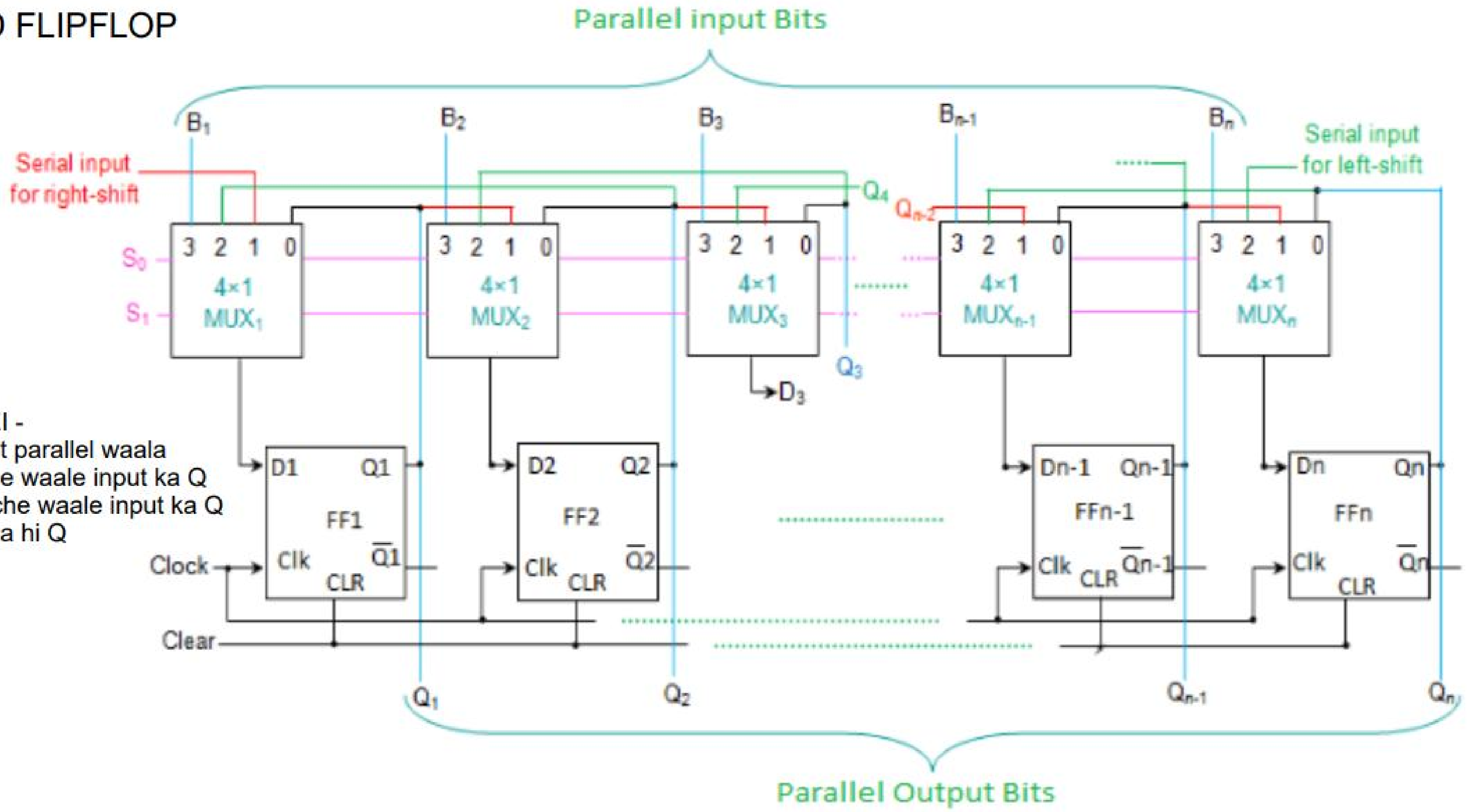
Universal Shift Register



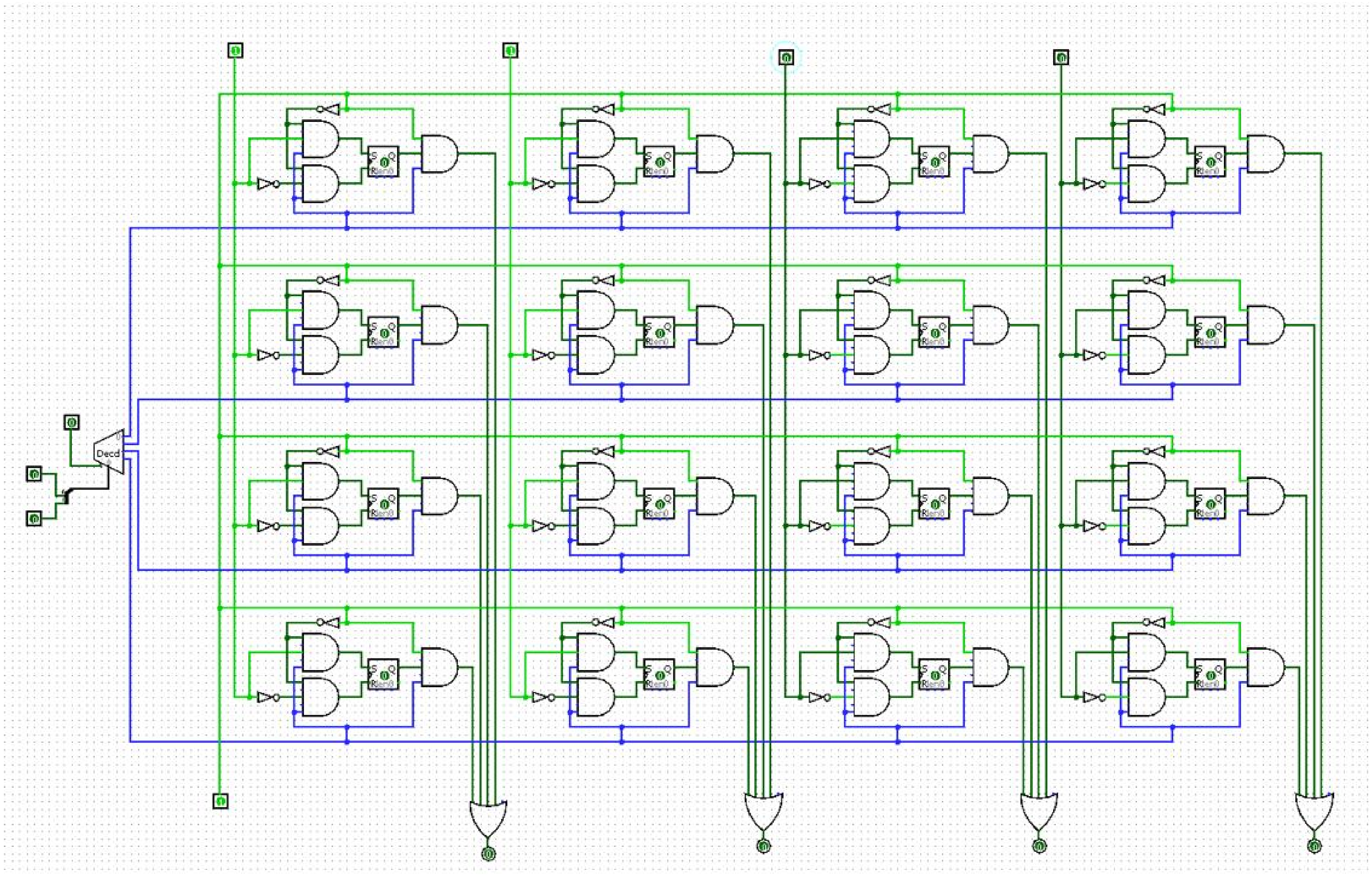
Using SR flipflop

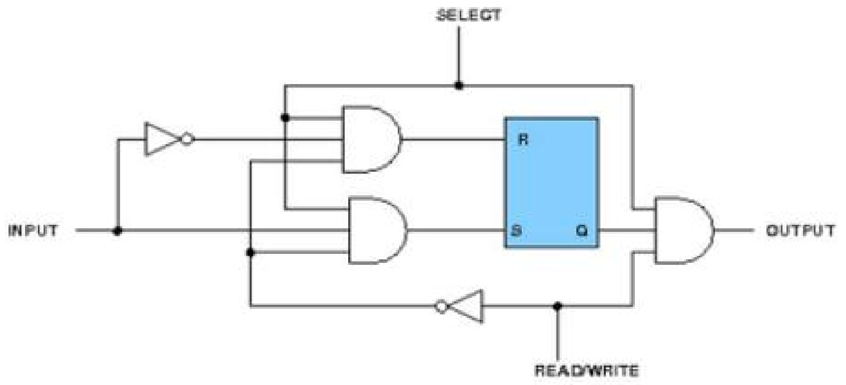
Universal Shift Register

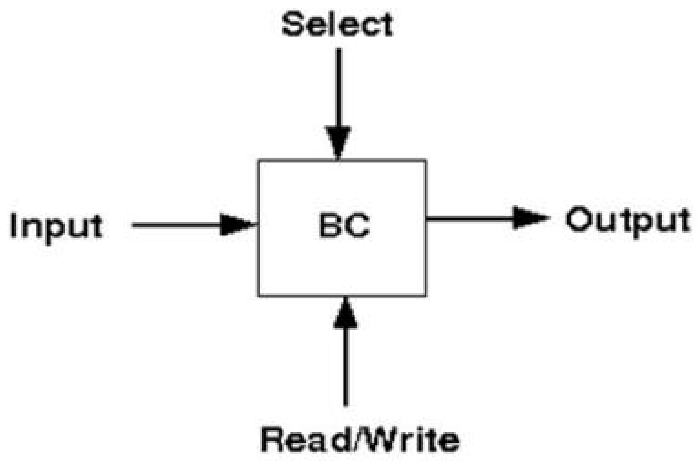
USING D FLIPFLOP



4X1 MUX MEI -
pin 3 me input parallel waala
pin 2 mei aage waale input ka Q
pin 1 mei picche waale input ka Q
pin 0 mei apna hi Q


Decd





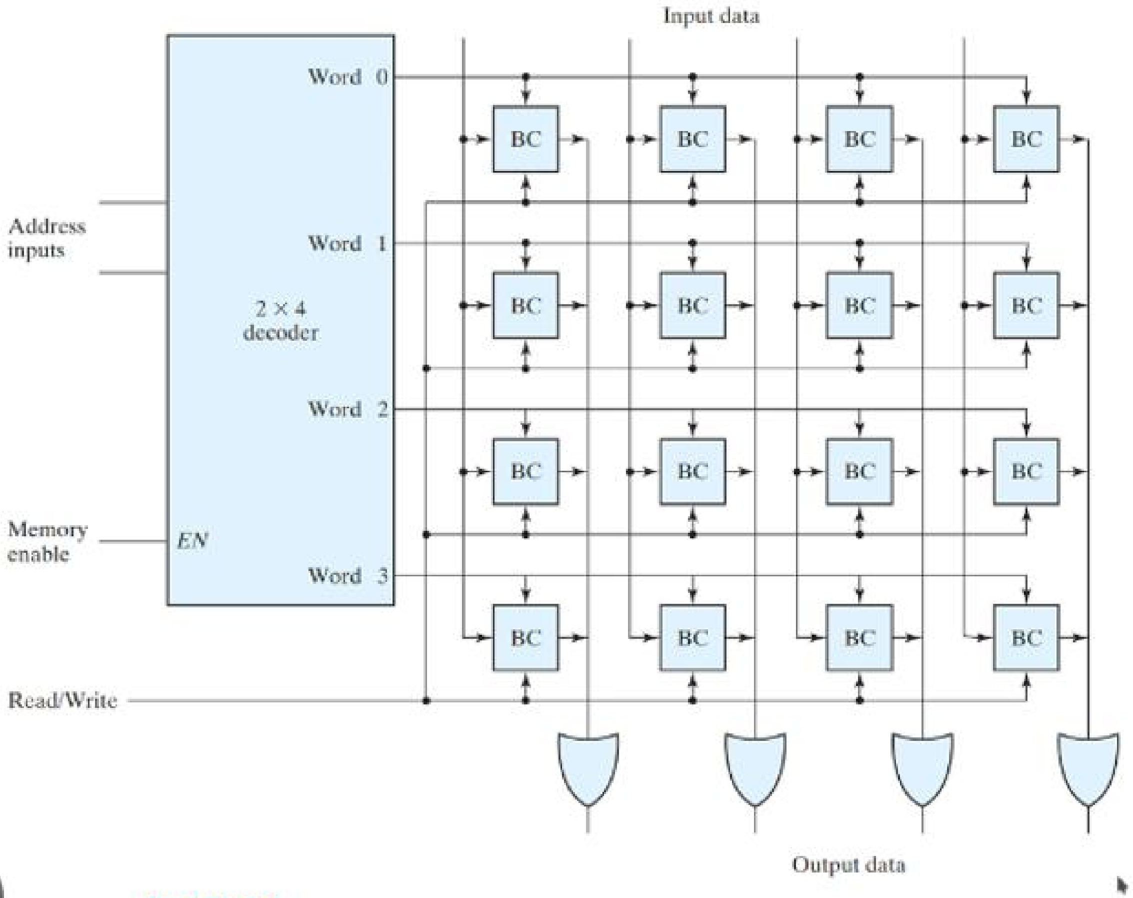
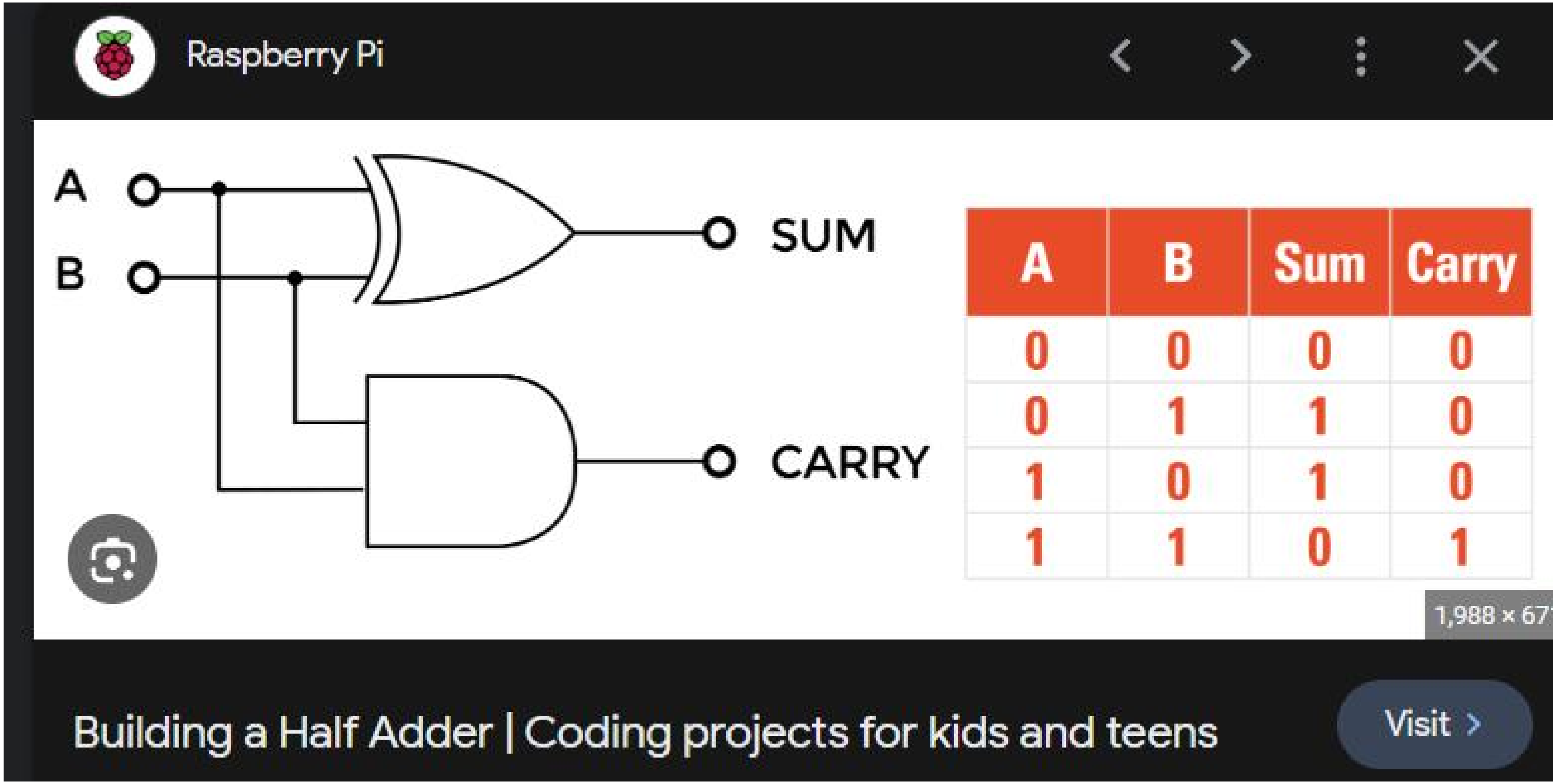


FIGURE 7.6
Diagram of a 4 × 4 RAM

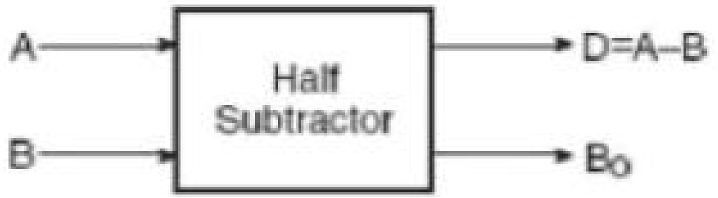


A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Building a Half Adder | Coding projects for kids and teens

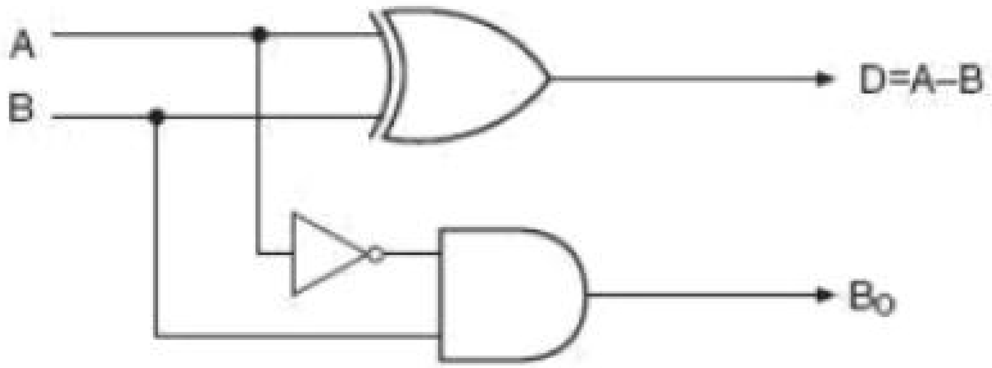
$$D = \overline{A}.B + A.\overline{B}$$

$$B_o = \overline{A}.B$$



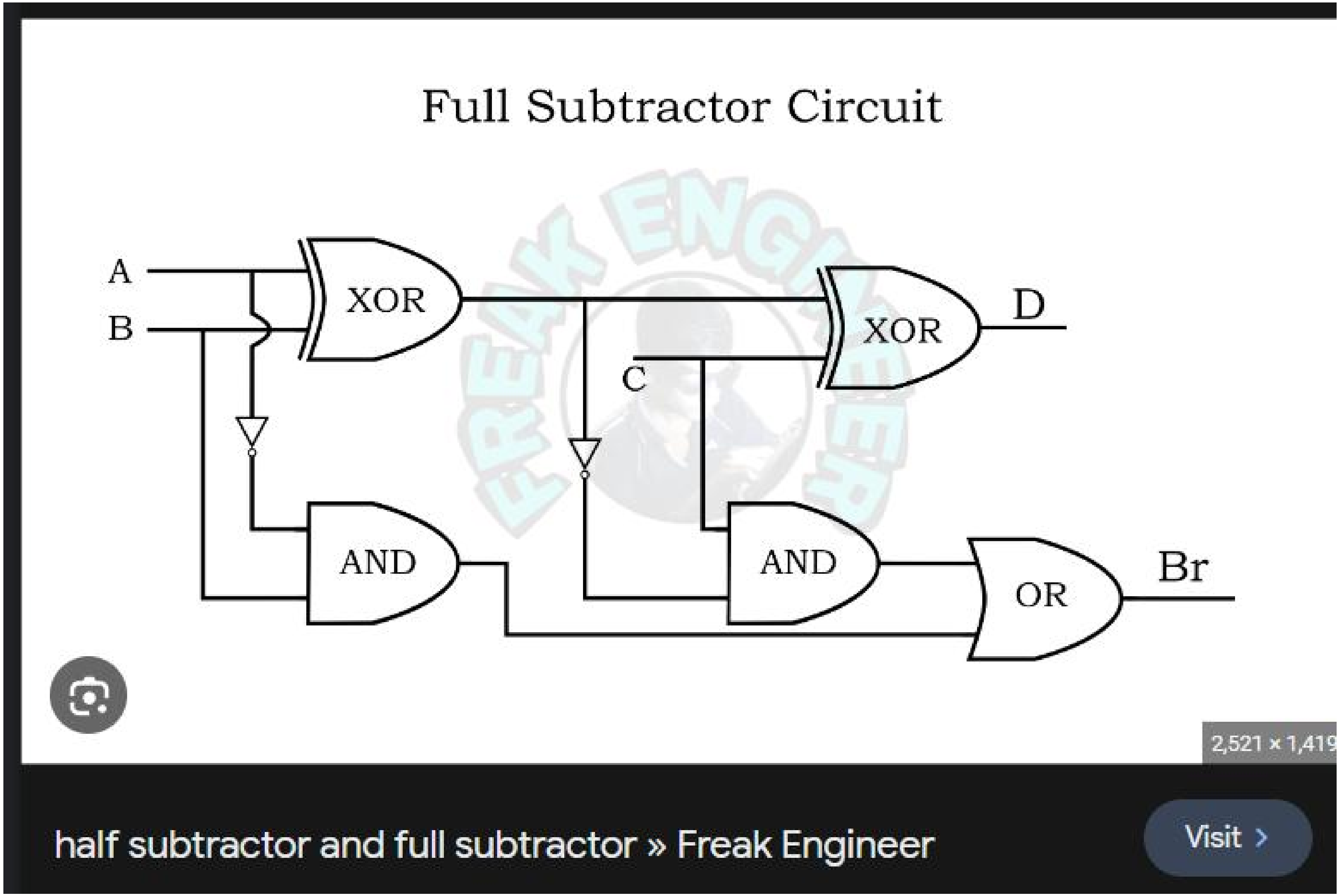
A	B	D	B_O
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor



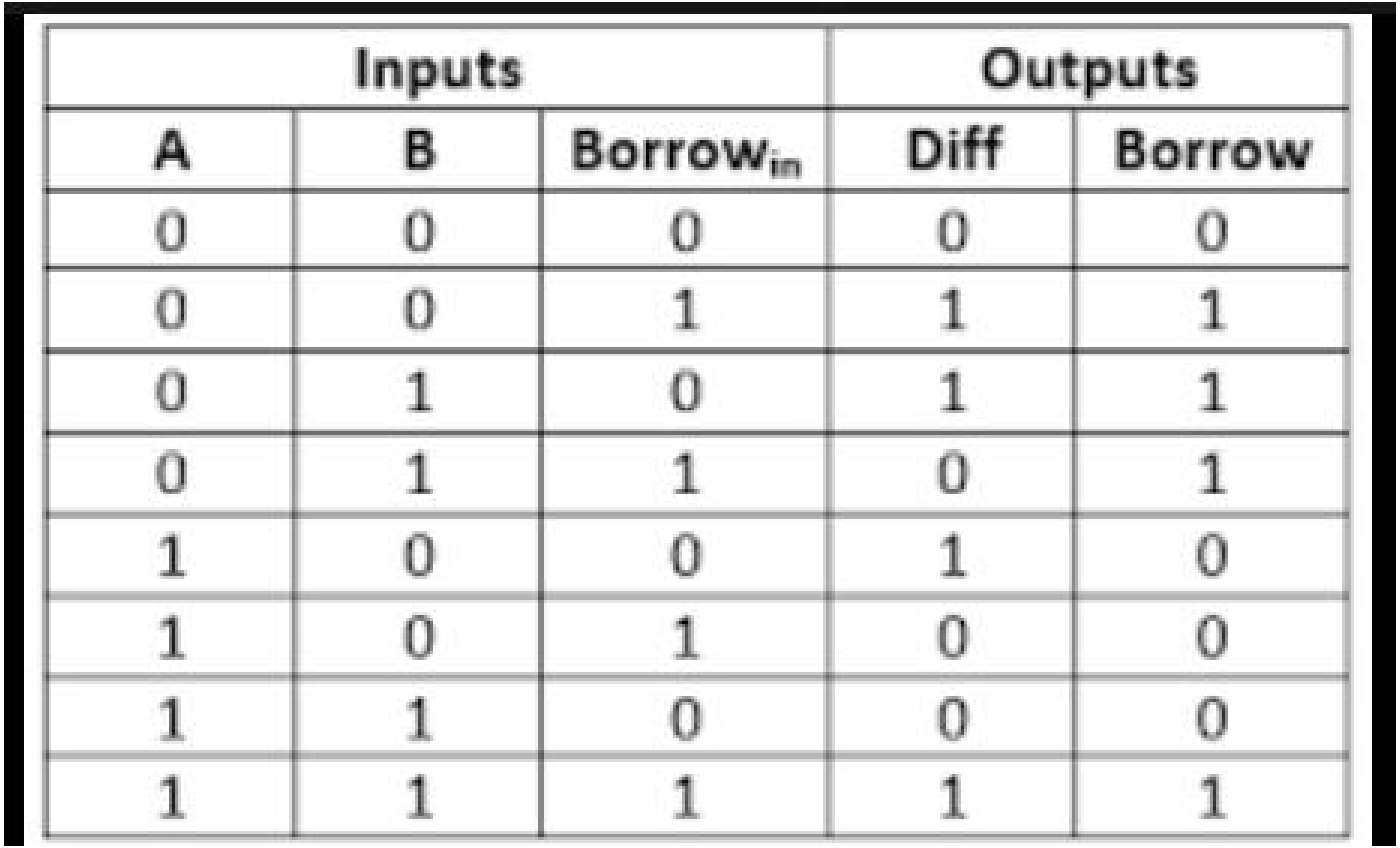
Half Subtractor

Visit >

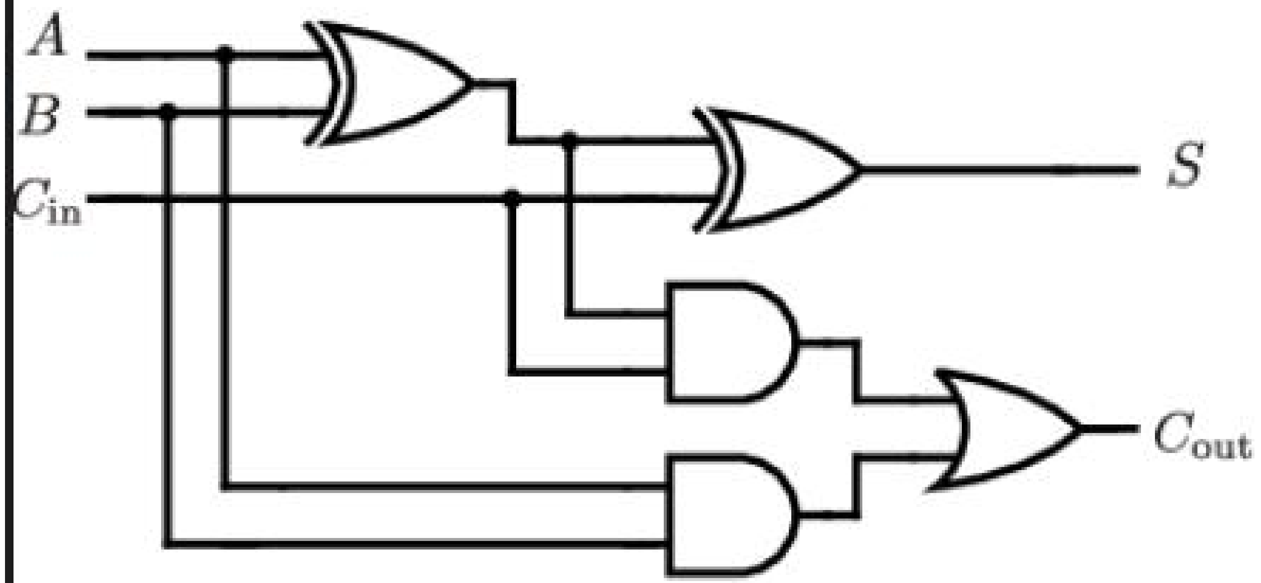


Full Subtractor Circuit

half subtractor and full subtractor » Freak Engineer



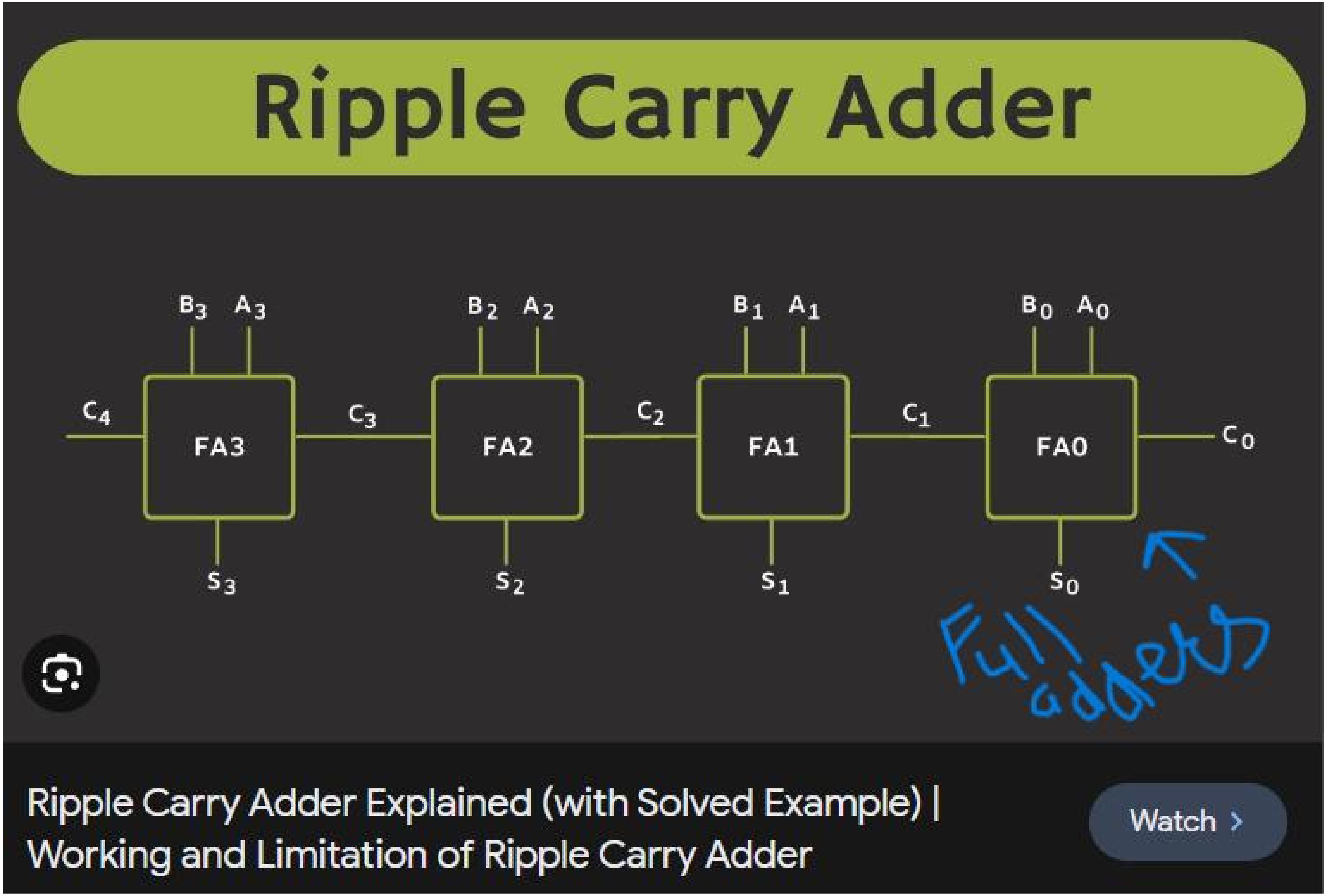
Inputs			Outputs	
A	B	Borrow_{in}	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



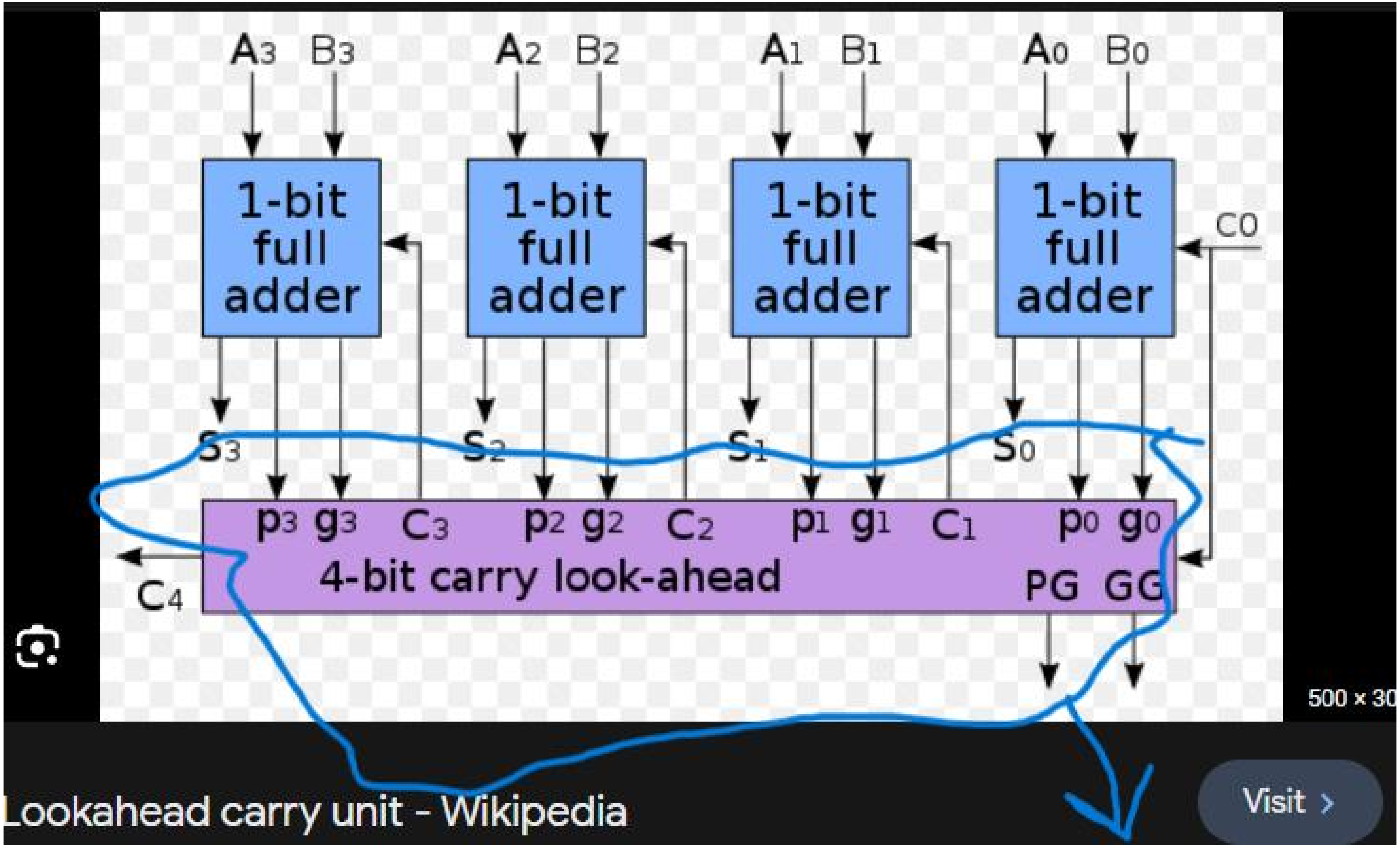
Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full-adder circuit diagram and truth table, where A, B, and C in are... | Download Scientific Diagram

Visit ›

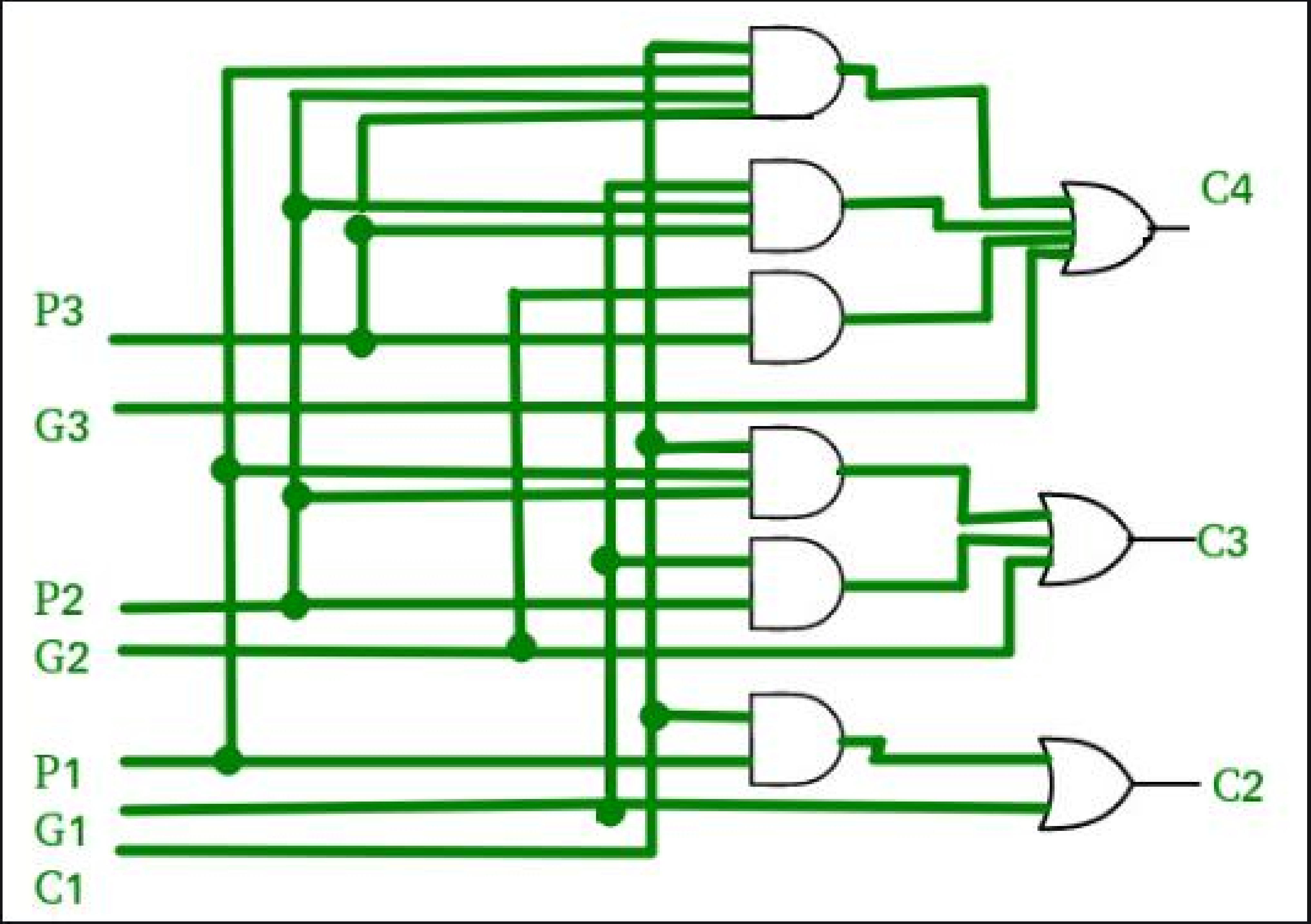


Ripple Carry Adder Explained (with Solved Example) | Working and Limitation of Ripple Carry Adder



Lookahead carry unit - Wikipedia

To understand carry look ahead adder - https://www.geeksforgeeks.org/carry-look-ahead-adder/


C4
P3
G3
C3
P2
G2
P1
C2
G1
C1