Unit-5:   Combinational Circuit

Features of Combinational Circuit –

1. In this output depends only upon present input.

2. Simplest design and Speed is fast

3. There is no feedback between input and output.

4. Elementary building blocks: Logic gates

5. Used for arithmetic as well as boolean operations.

6. Combinational circuits don’t have capability to store any state.

7. As combinational circuits don’t have clock, they don’t require triggering.

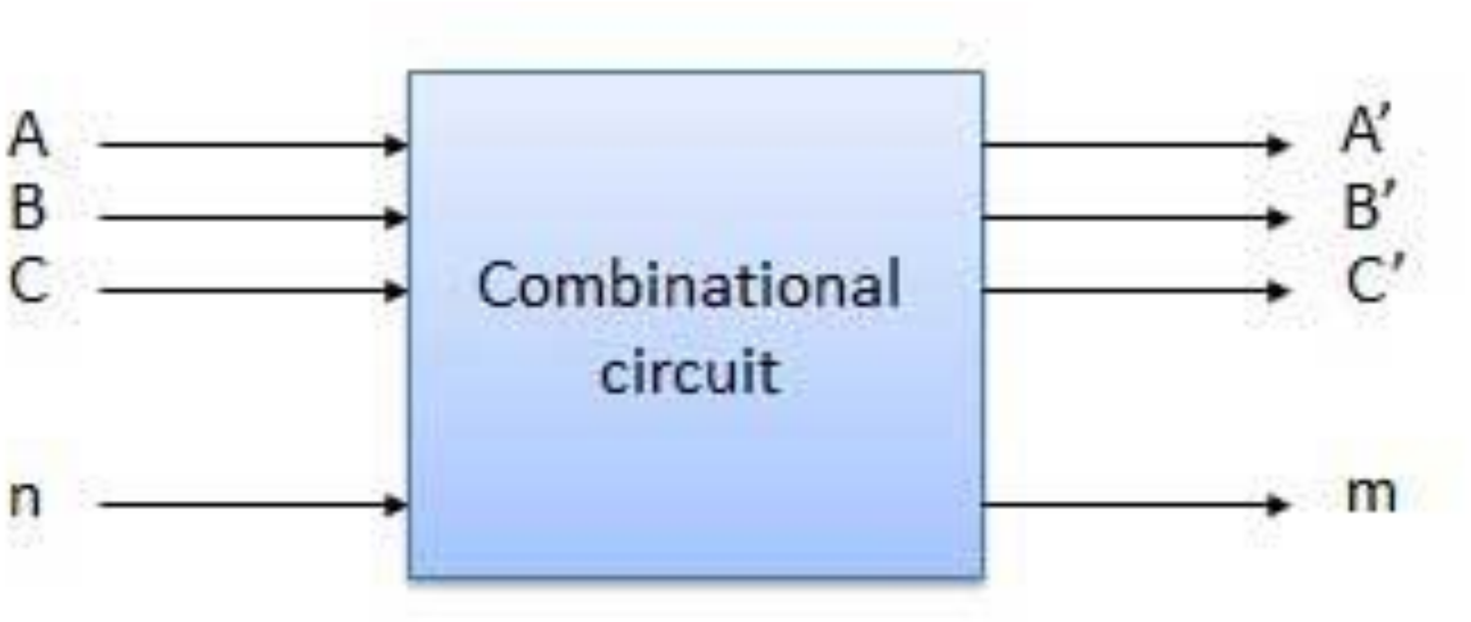
8. These circuits do not have any memory element.

Examples – Encoder, Decoder, Multiplexer, Demultiplexer

Adders
Subtractor
Comparator
Parity Circuit

Combinational Circuits

- output depends only on the present input
- The combinational circuit do not use any memory.
- The previous state of input does not have any effect on the present state of the circuit.

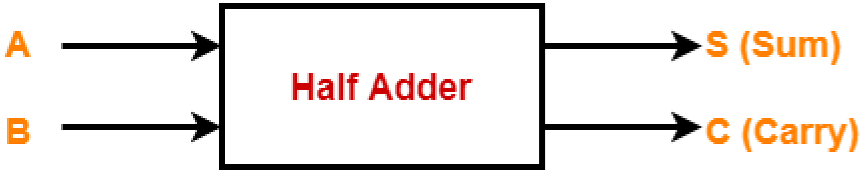


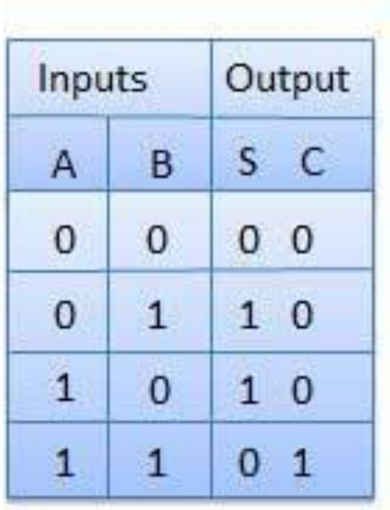
Half Adder

A combinational logic circuit with two inputs and two outputs.

The half adder circuit add two single bit Cary number

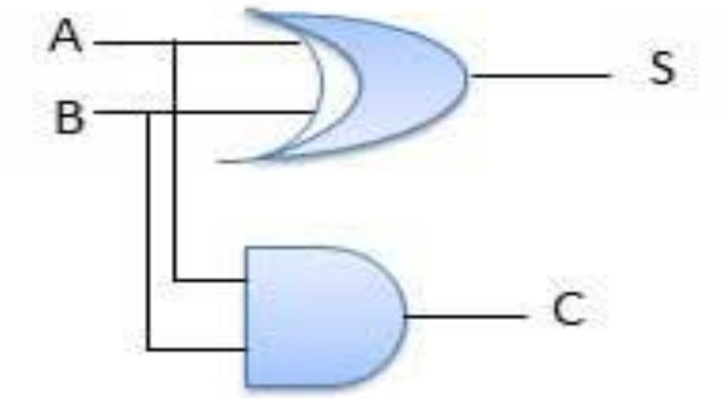
This circuit has two outputs **carry** and **sum**.





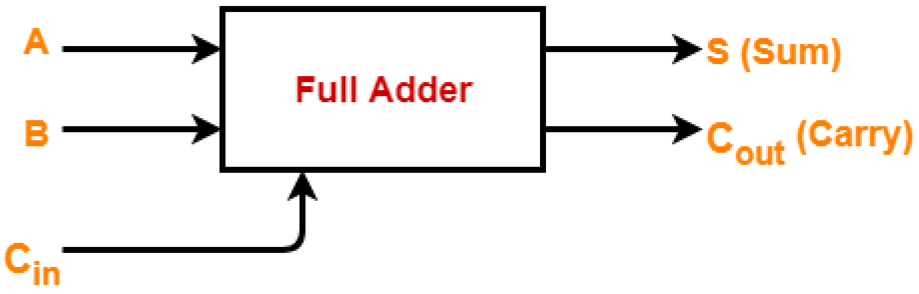
Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S(A, B) = ∑m (1, 2)
CY(A, B) = ∑m (3)



Full Adder (1-bit Adder)

A combinational logic circuit with 3 inputs and 2 outputs.
The Full adder circuit add 3 single bit Cary number
This circuit has two outputs **carry** and **sum**.



Sum(A, B,C) = ∑m (1, 2, 4, 7)
Cout(A, B, C) = ∑m (3,5,6,7)

For S:

$A \backslash BC_{in}$	$\overline{B}\overline{C}_{in}$	$\overline{B}C_{in}$	BC_{in}	$B\overline{C}_{in}$
$\overline{A}$		1		1
A	1		1	

A	B	Cin	Sum	Cout
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

$$\overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + ABC$$

$$\overline{A}(\overline{B}C + B\overline{C}) + A(\overline{B}\,\overline{C} + BC)$$

$$\overline{A}(B \oplus C) + A(B \ominus C)$$

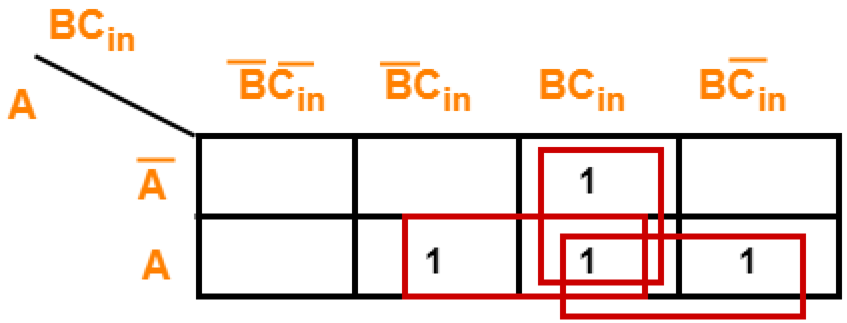
$$Let\ B \oplus C = D$$

$$\overline{A}D + A\overline{D}$$

$$A \oplus D$$

$$A \oplus B \oplus C$$

For Cout



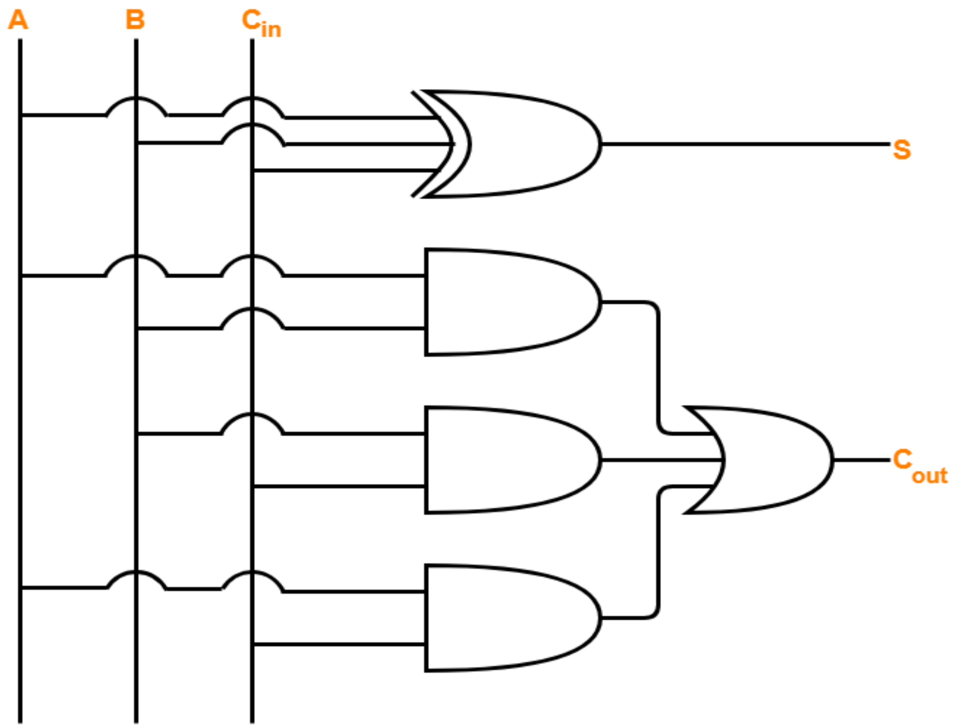
$A \backslash BC_{in}$	$\overline{B}\overline{C}_{in}$	$\overline{B}C_{in}$	BC_{in}	$B\overline{C}_{in}$
$\overline{A}$			1	
A		1	1	1

$$\overline{A}BC + A\overline{B}C + ABC + AB\overline{C}$$

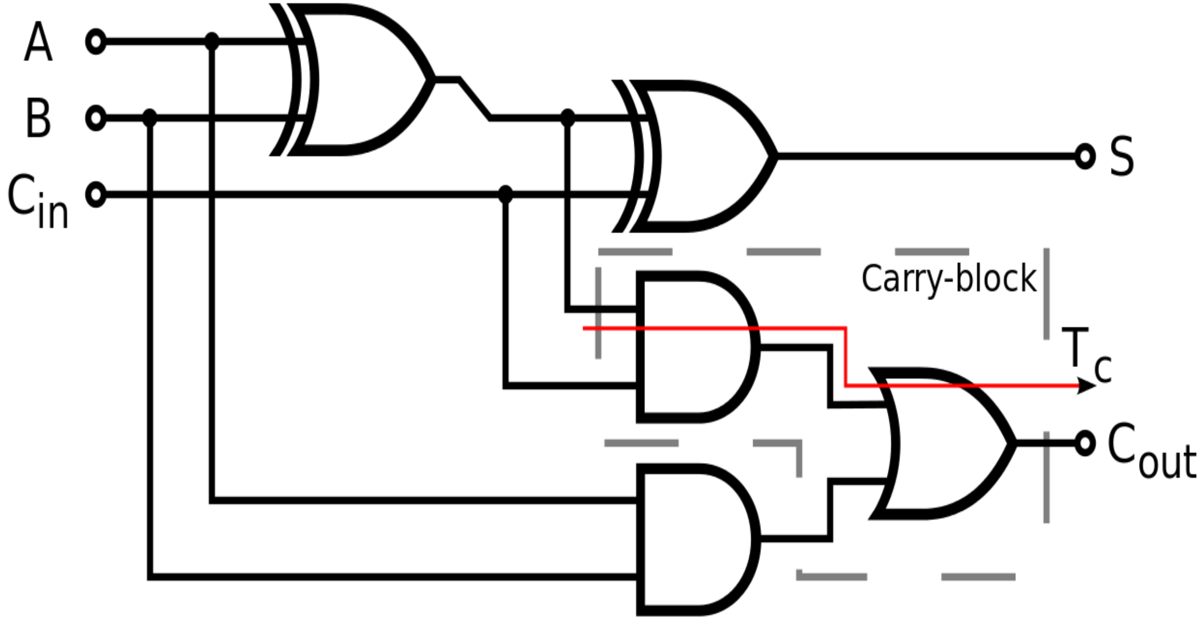
$$C(\overline{A}B + A\overline{B}) + AB(C + \overline{C})$$

$$(A \oplus B)C + AB$$

Cout=AB+BC+AC



Full Adder Logic Diagram



Full Adder using Half Adder

FA

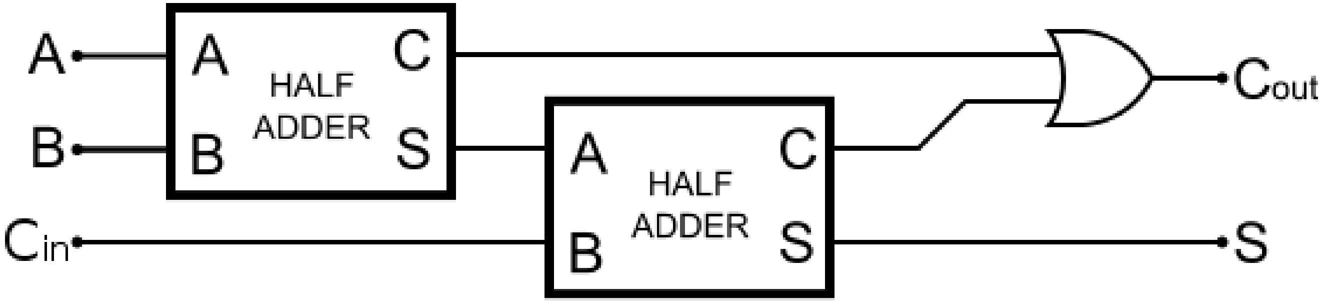
Sum= A⨁B⨁C

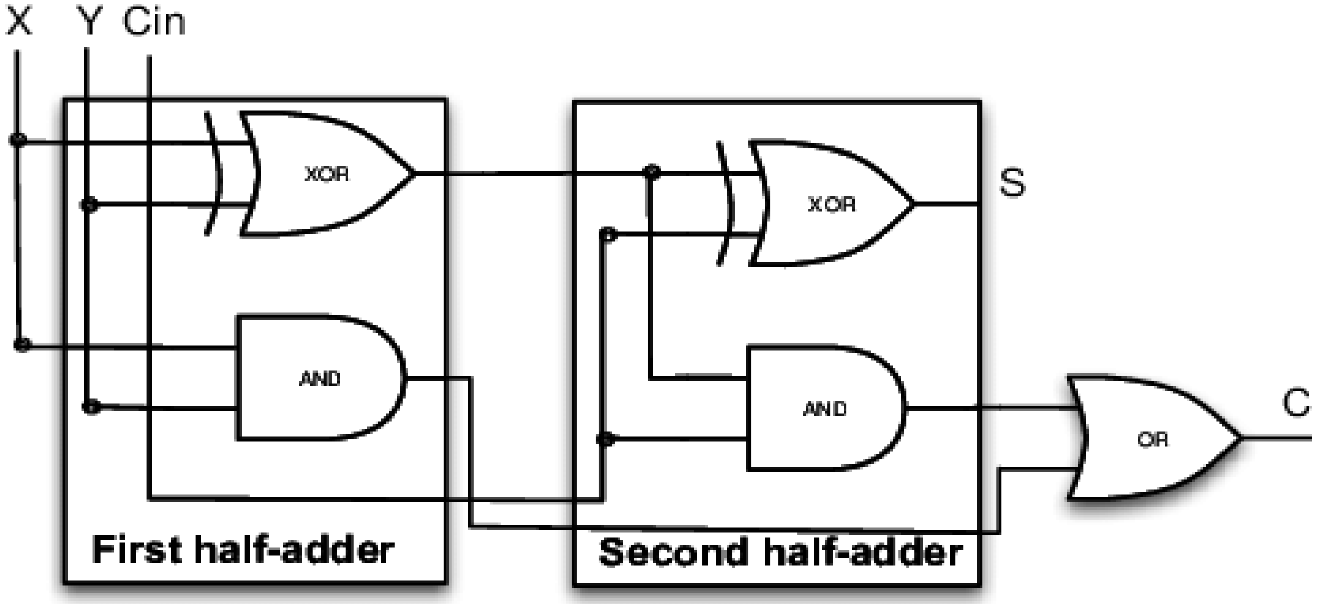
Carry=AB+(A⨁B)C

HA

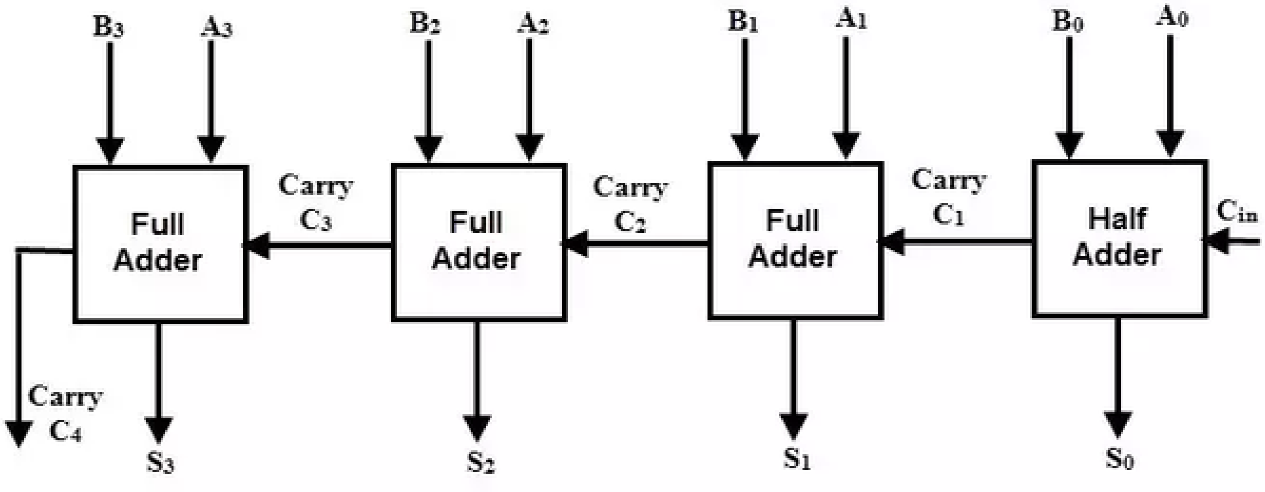
Sum= A⨁B

Carry=AB



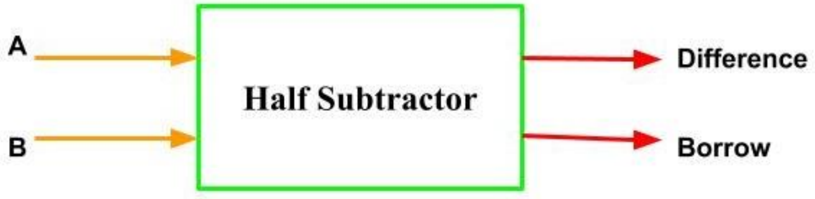


4-bit Ripple Carry Adder



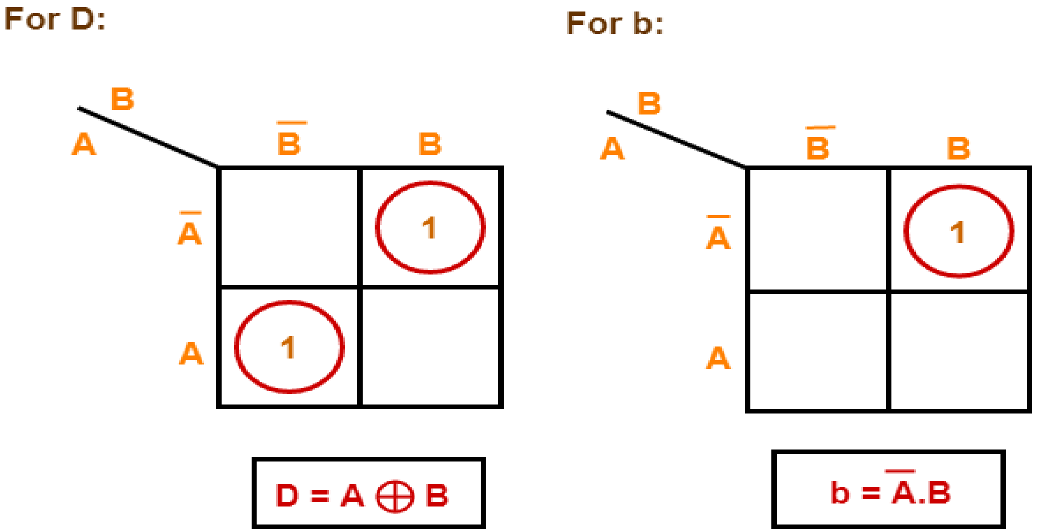
Half Subtractor

Combinational circuit perform binary Subtraction
Accepts 2 input and Two output Difference and Borrow

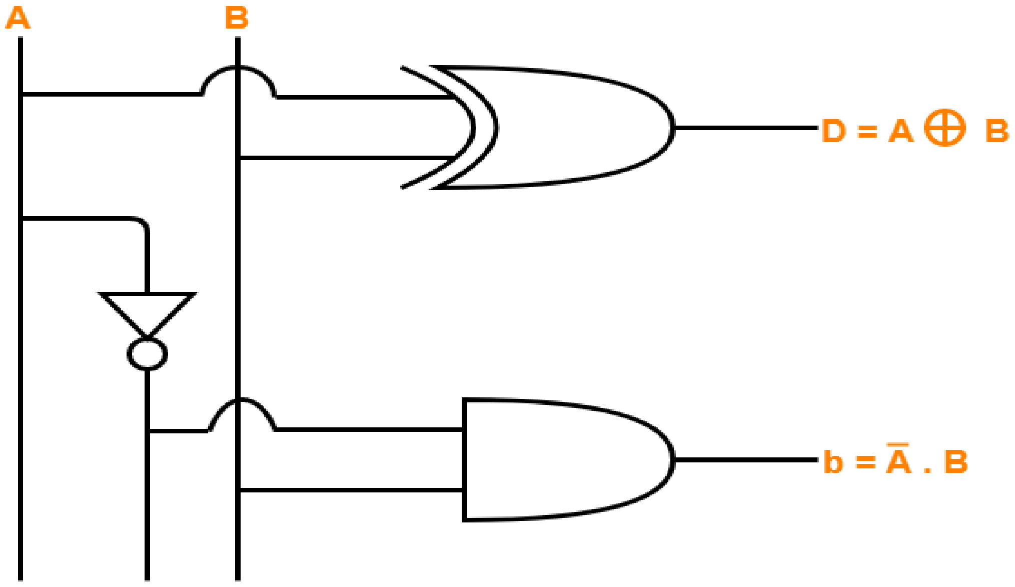


Inputs		Outputs	
A	B	D (Difference)	b (Borrow)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D(A, B) = \sum m\ (1, 2)$$
$$Br(A, B) = \sum m\ (1)$$



K Maps



Half Subtractor Logic Diagram

Full Subtractor

Performs subtraction of 3 bits

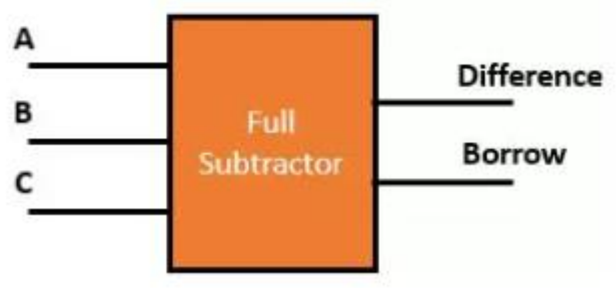
This circuit **has three inputs and two outputs**.

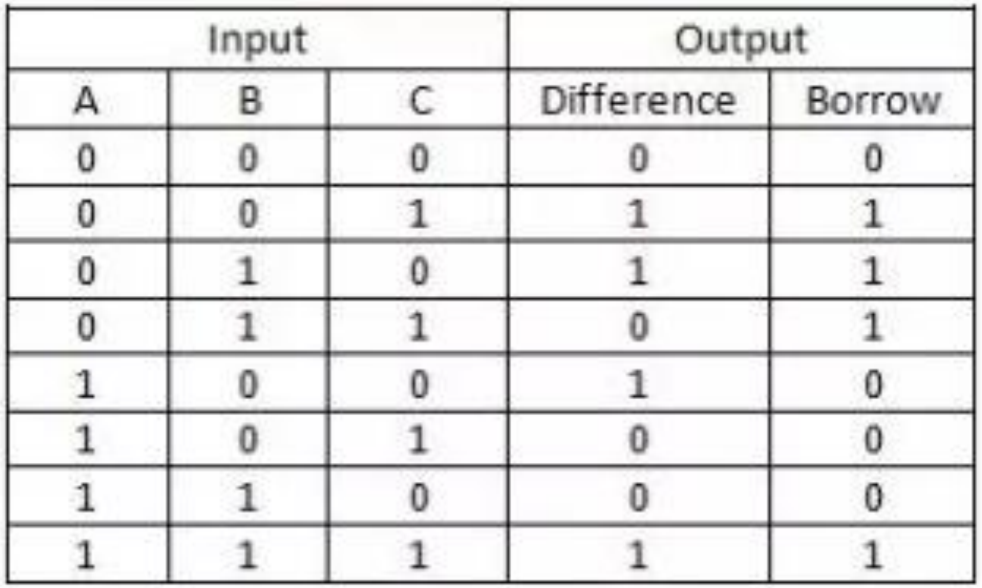
The three inputs A, B and C, denote the minuend, subtrahend, and previous borrow, respectively.

The two outputs, D and Bout

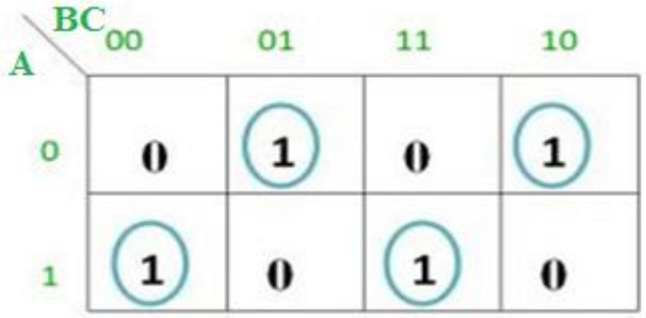
Sum(A, B,C) = ∑m (1, 2, 4, 7)

Bout(A, B,C) = ∑m (1,2,3,7)



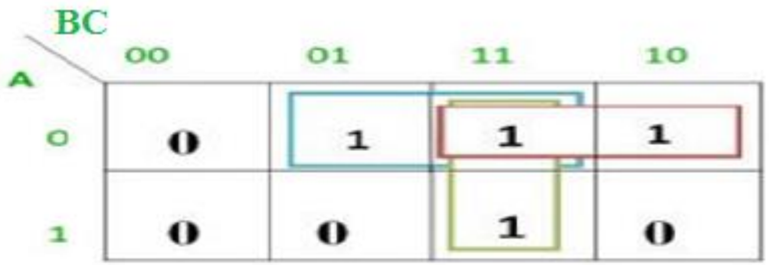


Input			Output	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$
\begin{aligned}
Difference &= \overline{A}\ \overline{B}\,C + \overline{A}\,B\,\overline{C} + A\,\overline{B}\ \overline{C} + ABC \\
&= C\,(\overline{A}\ \overline{B} + AB) + \overline{C}\ (\overline{A}\,B + A\,\overline{B}\,) \\
&= C\ (A\odot B) + \overline{C}\ (A\oplus B) \\
&= C\ (\overline{A\oplus B}\,) + \overline{C}\ (A\oplus B) \\
&= C \oplus (A\oplus B)
\end{aligned}
$$

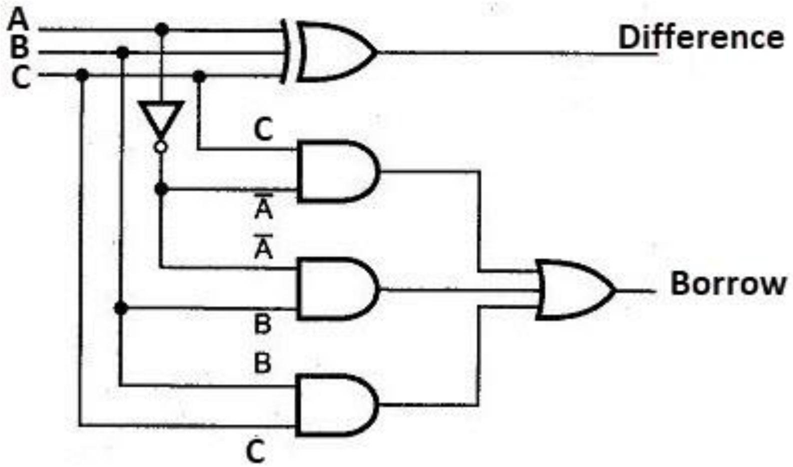


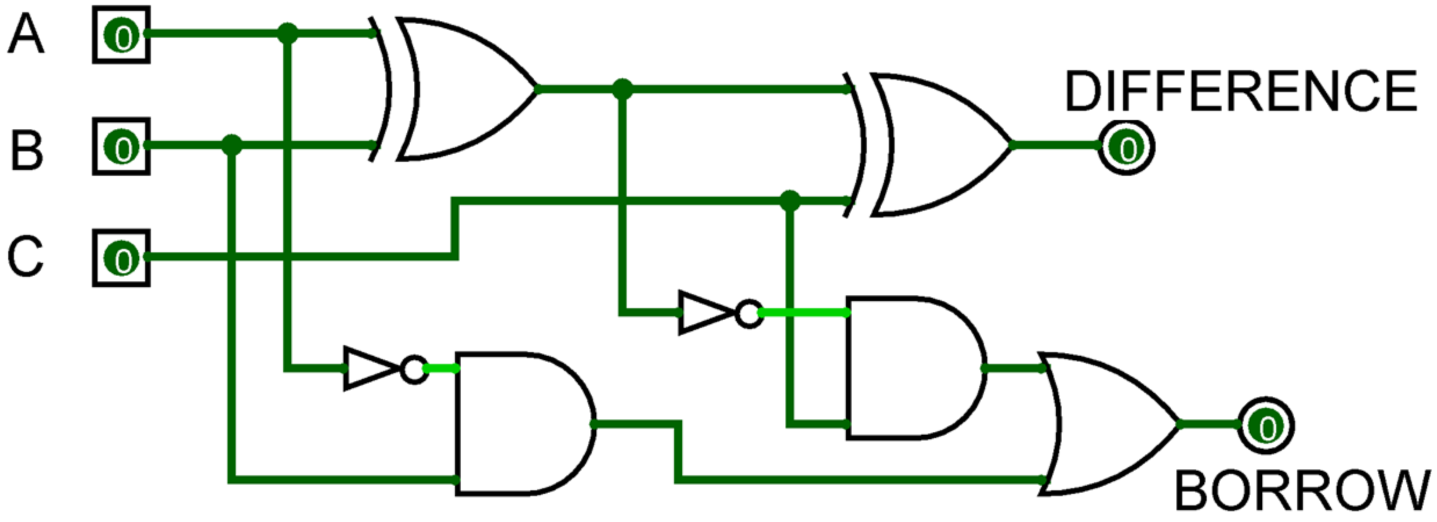
A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

```
Bout = A’B’C + A’BC’ + A’BC + ABC
     = C(AB + A’B’) + A’B(C + C’)
     = C( A XNOR B) + A’B
     = C (A XOR B)’ + A’B
```

$$Borrow = \overline{A}\ \overline{B}\ C + \overline{A}\ B\ \overline{C} + \overline{A}\ BC + ABC$$

$$= \overline{A}\ B + \overline{A}\ C + BC$$





Full Subtractor using Half Subtractor

FS

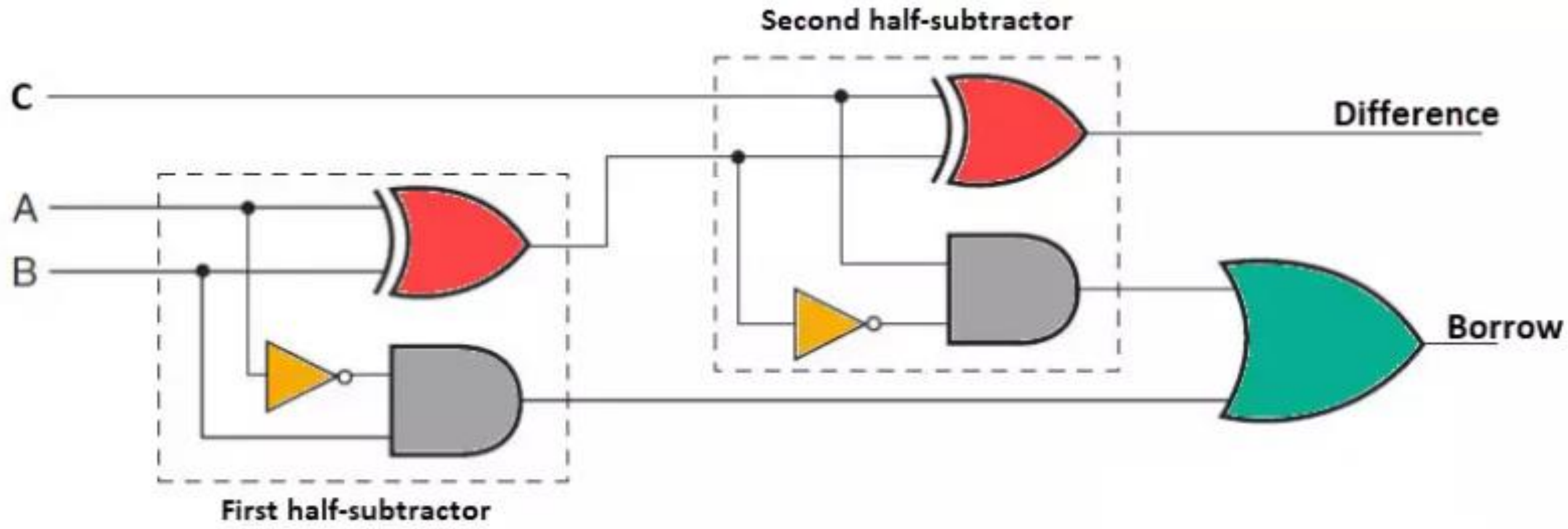
Sum= A⊕B⊕C

Carry=A'B+(A⊕B)'C

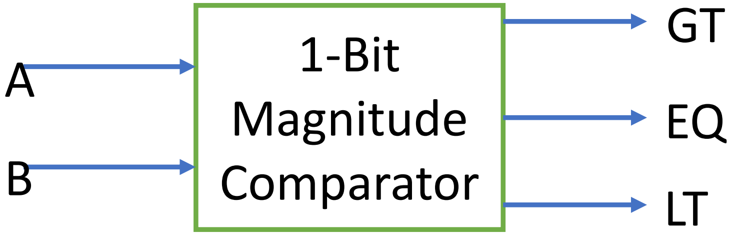
HS

Sum= A⊕B

Carry=A'B



1-BIT Magnitude Comparator



A

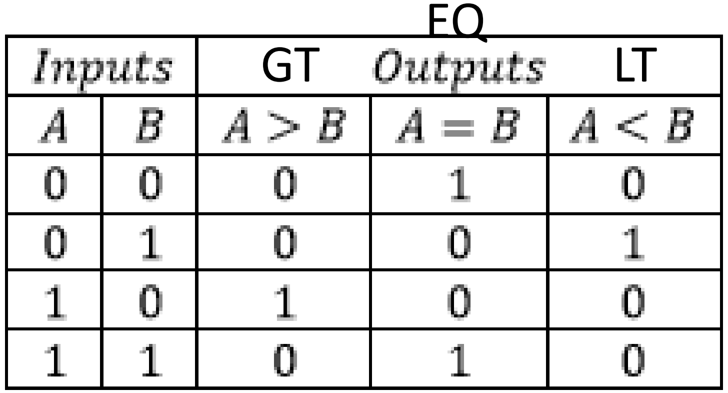
B

1-Bit Magnitude Comparator

GT

EQ

LT

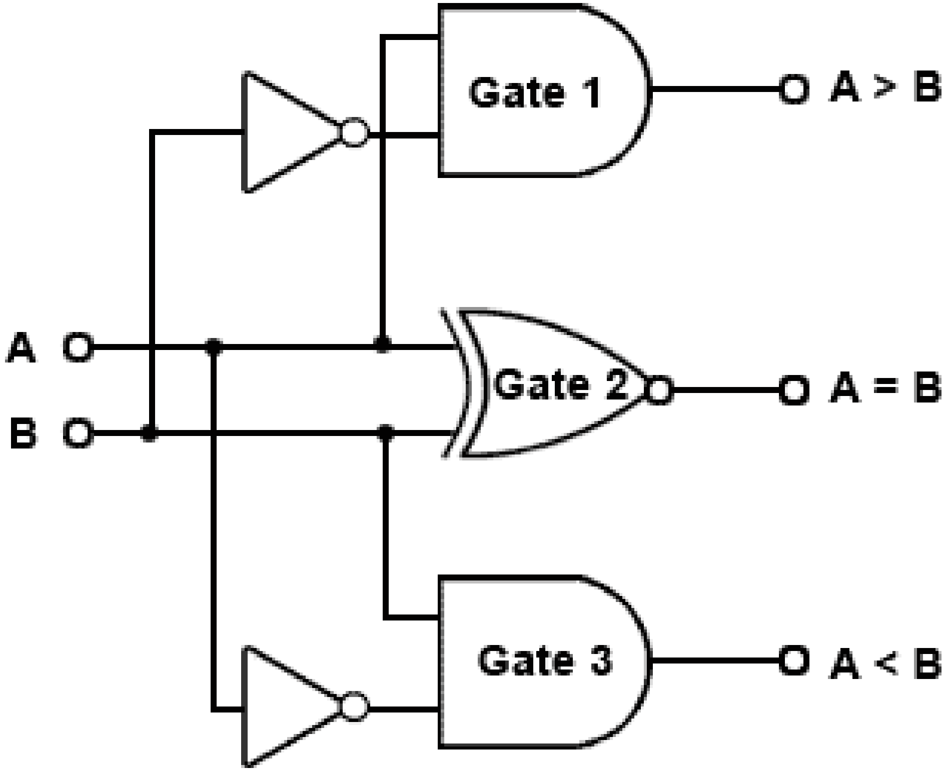


Inputs		GT	EQ *Outputs*	LT
A	*B*	*A* > *B*	*A* = *B*	*A* < *B*
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

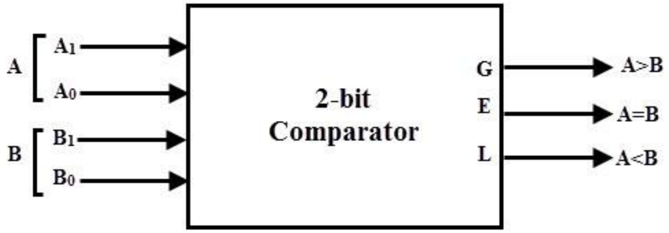
GT=AB’

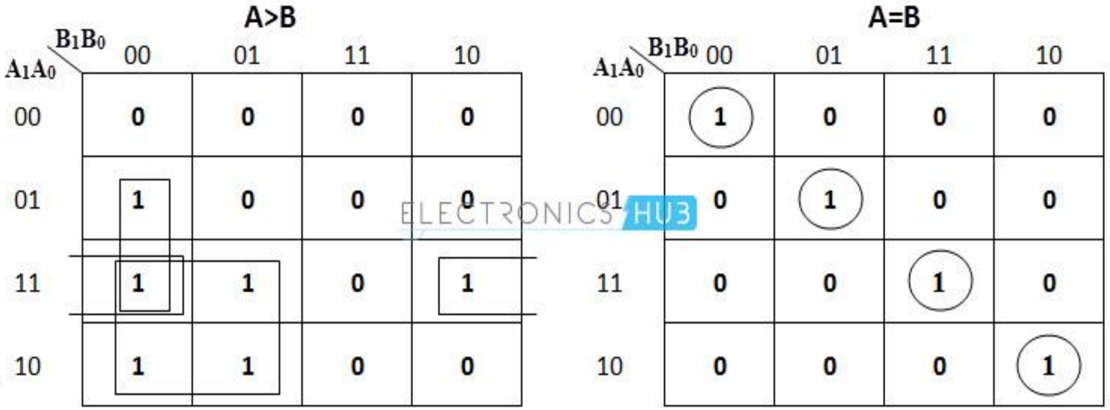
EQ=A’B’+AB

LT=A’B



2-BIT Magnitude Comparator





A>B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

A=B

A_1A_0 \ B_1B_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

$$A>B: G = A0\,\overline{B1}\,\overline{B0} + A1\,\overline{B1} + A1\,A0\,\overline{B0}$$

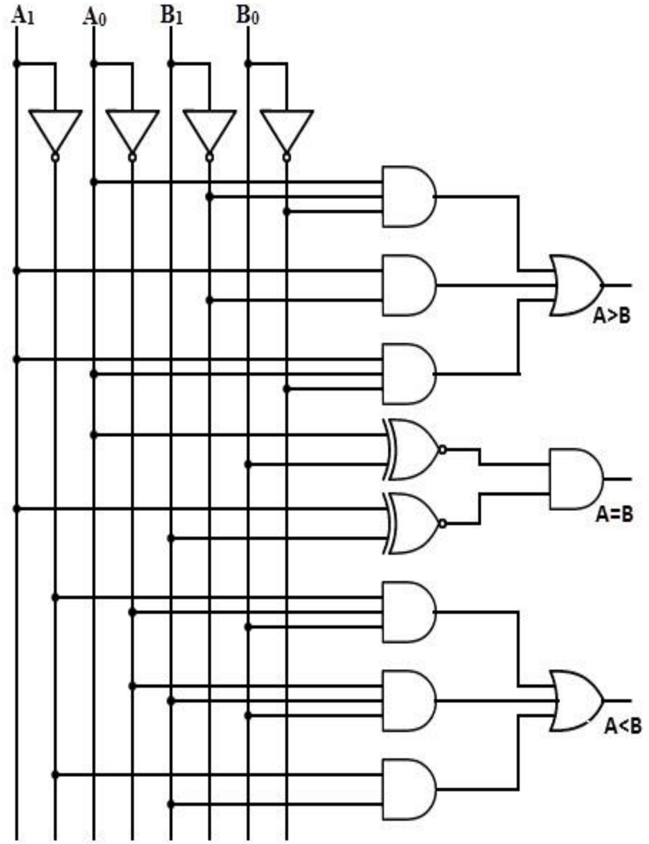
$$A = B: E = \overline{A1}\,\overline{A0}\,\overline{B1}\,\overline{B0} + \overline{A1}\,A0\,\overline{B1}\,B0 + A1\,A0\,B1\,B0 + A1\,\overline{A0}\,B1\,\overline{B0}$$

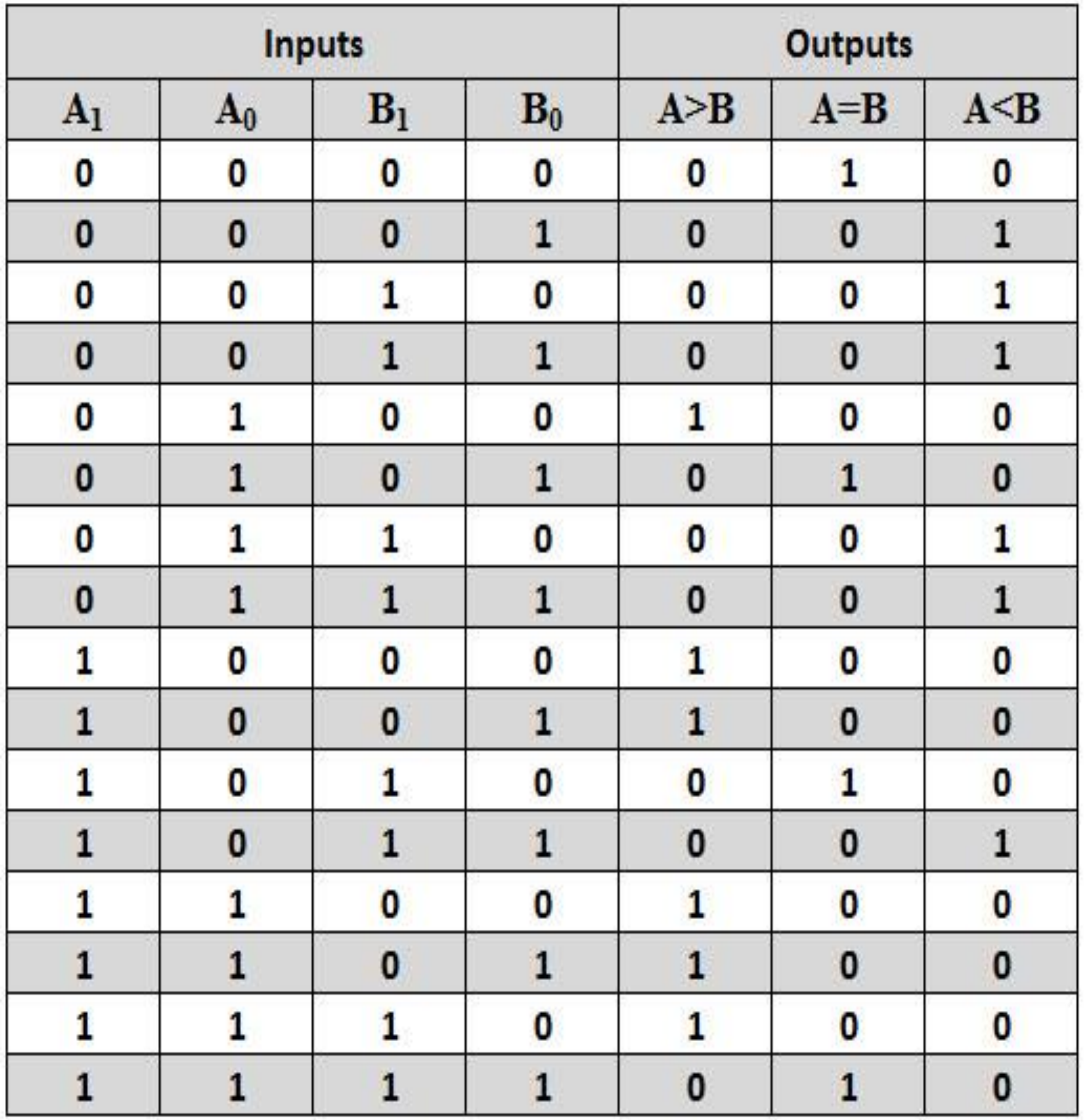
$$= \overline{A1}\,\overline{B1}\,(\overline{A0}\,\overline{B0} + A0\,B0) + A1\,B1\,(A0\,B0 + \overline{A0}\,\overline{B0})$$

$$= (A0\,B0 + \overline{A0}\,\overline{B0})\,(A1\,B1 + \overline{A1}\,\overline{B1})$$

$$= (A0 \text{ Ex-NOR } B0)\,(A1 \text{ Ex-NOR } B1)$$

$$A < B: L = \overline{A1}\,B1 + \overline{A0}\,B1\,B0 + \overline{A1}\,\overline{A0}\,B0$$





Inputs				Outputs		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Parity Circuit

- Error detection techniques for the data transmission.

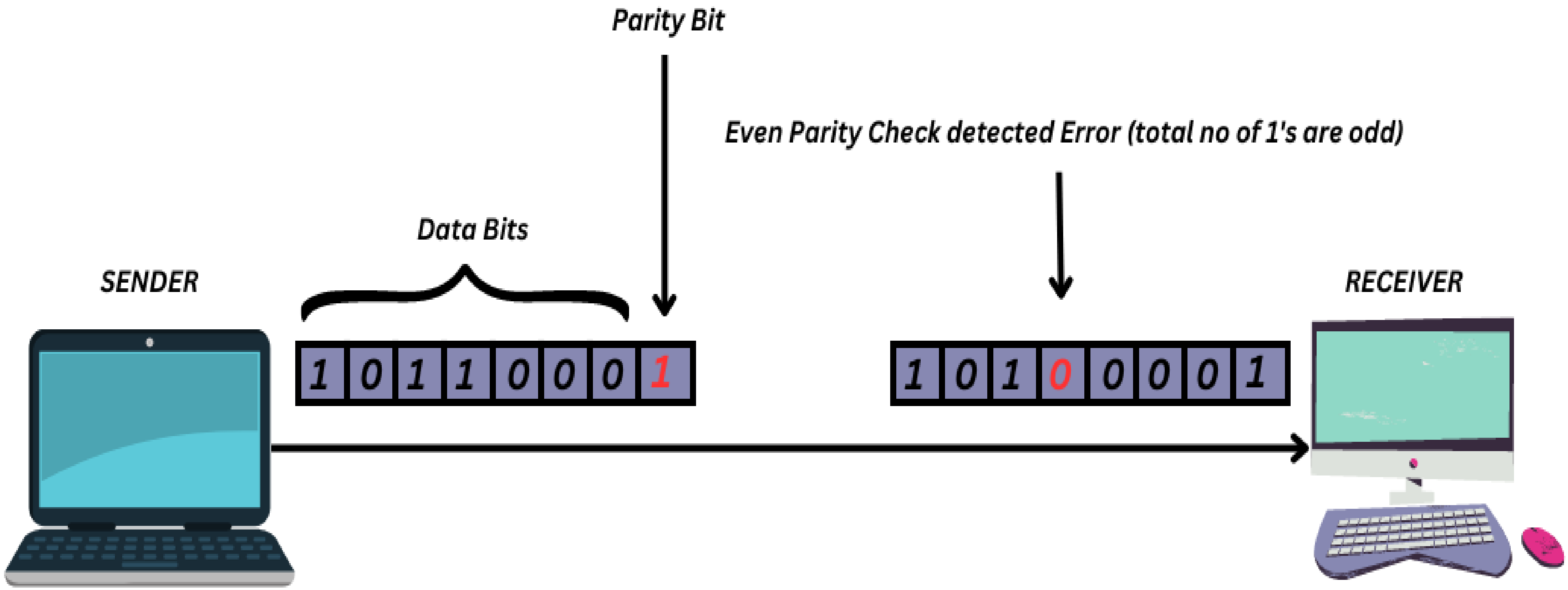
- In digital systems, when binary data is transmitted and processed, data may be alter 0s (of data bits) to 1s and 1s to 0s due to noise.

- Parity Bit is added to word containing data in order to make number of 1s either even or odd.

  The message containing the data bits along with parity bit is transmitted.

- At the receiving end, number of 1s in the message is counted and if it doesn’t match with the transmitted one, it means there is an error in the data.

  Thus, Parity Bit is used to detect errors, during the transmission of binary data.



Parity Bit

Even Parity Check detected Error (total no of 1's are odd)

Data Bits

SENDER

RECEIVER

1 0 1 1 0 0 0 1

1 0 1 0 0 0 0 1

Parity Generator Circuit

Even Parity Generator

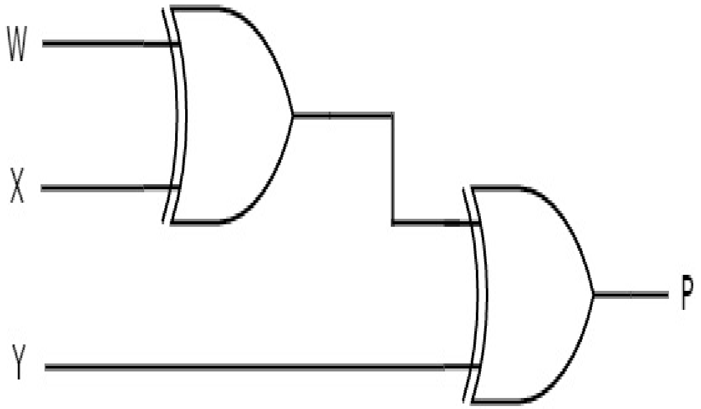
If odd number of ones present in the input, then even parity bit, P should be ‘1’ so that the resultant word contains even number of ones.

Binary Input WXY	Even Parity bit P
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

$$P = W'X'Y + W'XY' + WX'Y' + WXY$$

$$\Rightarrow P = W'(X'Y + XY') + W(X'Y' + XY)$$

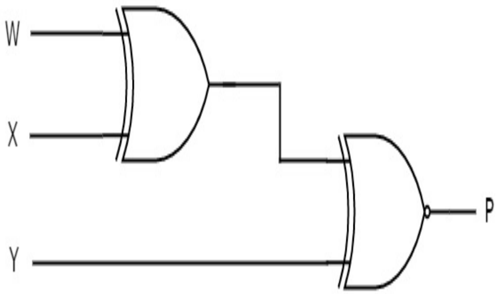
$$\Rightarrow P = W'(X \oplus Y) + W(X \oplus Y)' = W \oplus X \oplus Y$$



Odd Parity Generator

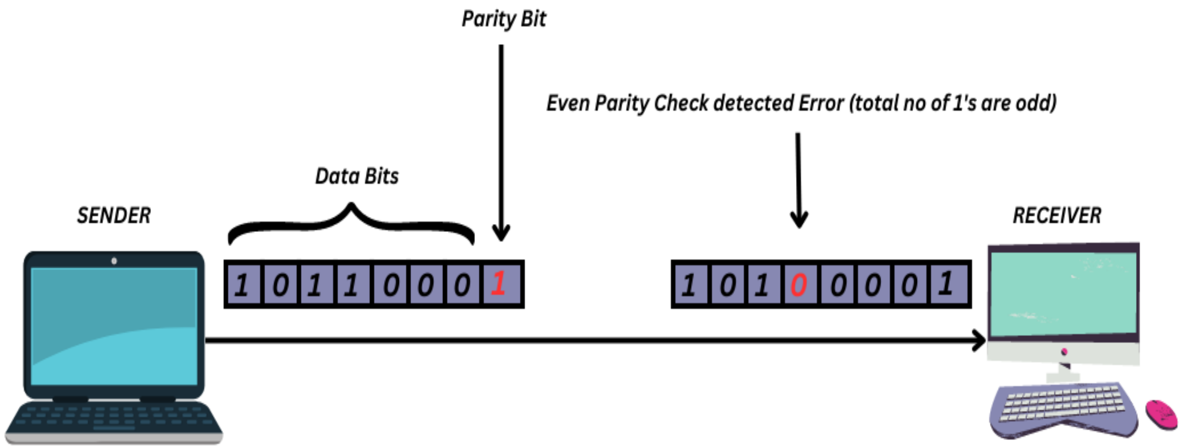
If even number of ones present in the input, then odd parity bit, P should be ‘1’ so that the resultant word contains odd number of ones

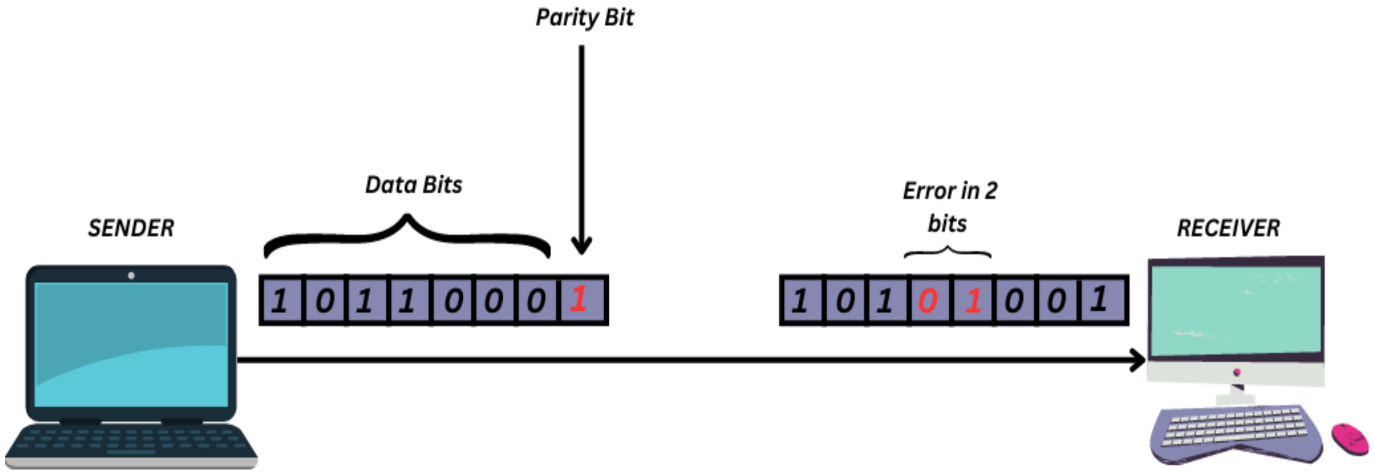
Binary Input WXY	Odd Parity bit P
000	1
001	0
010	0
011	1
100	0
101	1
110	1
111	0



Parity Checker

- The parity checker won't be able to detect if there are errors in more than '1' bit and the correct of data is also not possible.





Parity Checker Circuit

Even parity checker checks error in the transmitted data, which contains message bits along with even parity.

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

$$PEC = \overline{A}\,\overline{B}\,(\overline{C}\,P + C\,\overline{P}) + \overline{A}\,B(\overline{C}\,\overline{P} + CP) + AB(\overline{C}\,P + C\,\overline{P}) + A\overline{B}\,(\overline{C}\,\overline{P} + CP)$$

$$= \overline{A}\,\overline{B}\,(C \oplus P) + \overline{A}\,B(\overline{C \oplus P}) + AB(C \oplus P) + A\overline{B}\,(\overline{C \oplus P})$$

$$= (\overline{A}\,\overline{B} + AB)(C \oplus P) + (\overline{A}\,B + A\overline{B})(\overline{C \oplus P})$$

$$= (A \oplus B) \oplus (C \oplus P)$$