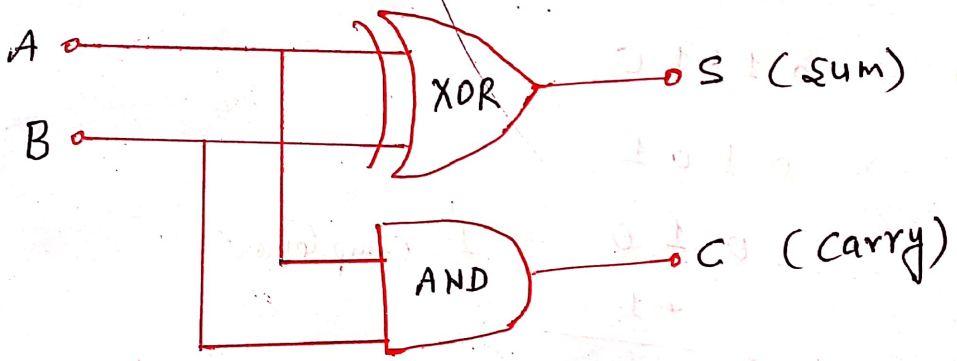
Half Adder and Full Adder

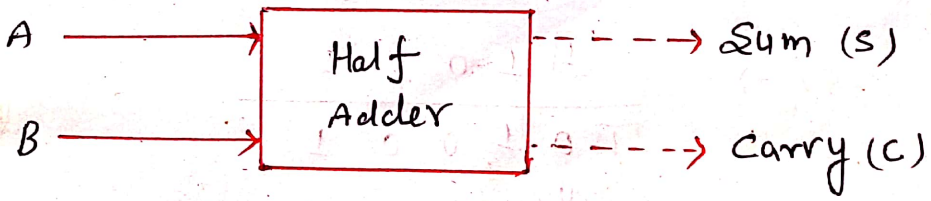
* An adder is a digital circuit that performs addition of numbers.

Half Adder :->

* The half adder adds two binary digits called as augend and addend and produces two outputs as. Sum and carry, XOR gate is applied to both inputs to produce Sum and AND gate is applied to both inputs to produce carry.



$$\text{Sum} = A \oplus B, \qquad \text{Carry} = A \cdot B$$

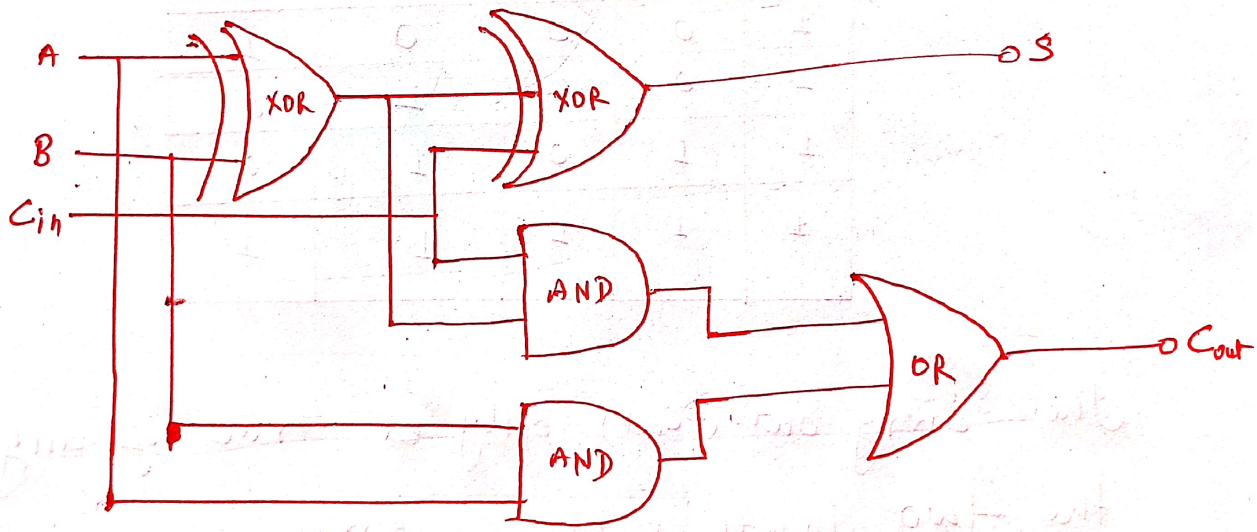


Truth Table:

INPUTS		OUTPUTS	
A	B	SUM(S)	CARRY(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

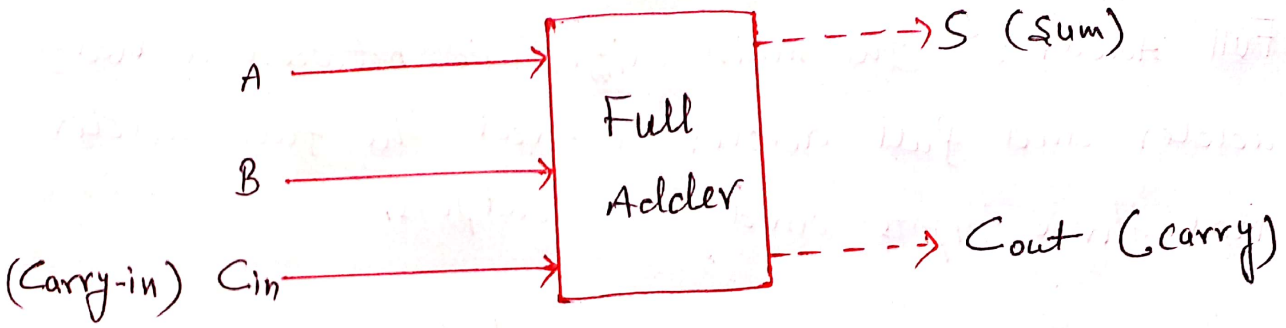
Full Adder :=> The main difference between a half adder and full adder is that the full adder has three inputs and two outputs.

The first two inputs are A and B and the third input is an input carry designated as Cin. When a full adder logic is designed we will able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.



$$\text{Sum} = (A \oplus B) \oplus C_{in}$$

$$\text{Carry} = A \cdot B + C_{in} \cdot (A \oplus B)$$



Truth Table:

INPUTS			OUTPUTS	
A	B	Cin	Cout	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

The carry-out (Cout) only be true if any of the two inputs out of the three are high. From the above truth table the full adder logic can be implemented.