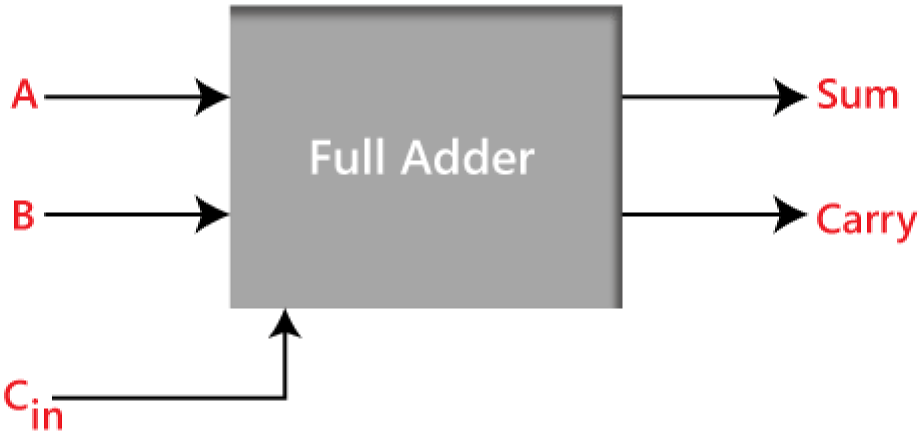
Full Adder

The half adder is used to add only two numbers. To overcome this problem, the full adder was developed. The full adder is used to add three 1-bit binary numbers A, B, and carry C. The full adder has three input states and two output states i.e., sum and carry.

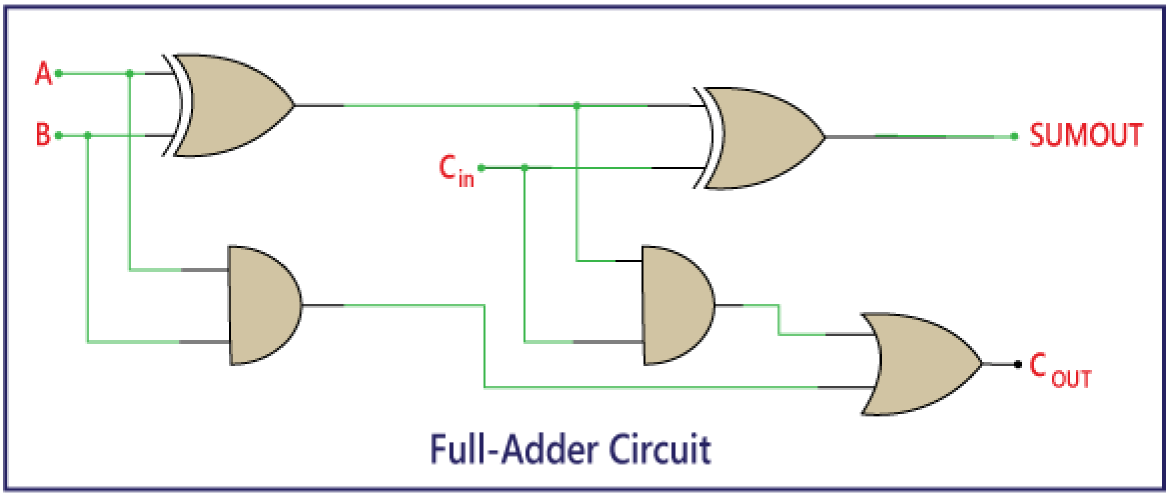
Block diagram



Truth Table

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
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

The full adder logic circuit can be constructed using the **'AND'** and **the 'XOR' gate** with an **OR gate**.



The actual logic circuit of the full adder is shown in the above diagram. The full adder circuit construction can also be represented in a Boolean expression.

Sum:

1. Perform the XOR operation of input A and B.

2. Perform the XOR operation of the outcome with carry. So, the sum is (A XOR B) XOR C_{in} which is also represented as:

$$(A \oplus B) \oplus C_{in}$$

Carry:

1. Perform the 'AND' operation of input A and B.
2. Perform the 'XOR' operation of input A and B.
3. Perform the 'OR' operations of both the outputs that come from the previous two steps. So the 'Carry' can be represented as:

   $A.B + (A \oplus B)$

References:

https://www.javatpoint.com/