

Exp-02-Implementation-of-Half-Adder-and-Full-Adder-circuit

' Implementation-of-Half-Adder-and-Full-Adder-circuit

' AIM:

To design a half adder and full adder circuit and verify its truth table in Quartus using Verilog programming.

' Equipments Required:

Hardware – PCs, Cyclone II , USB flasher Software – Quartus prime Theory Adders are digital circuits that carry out addition of numbers.

' Half Adder

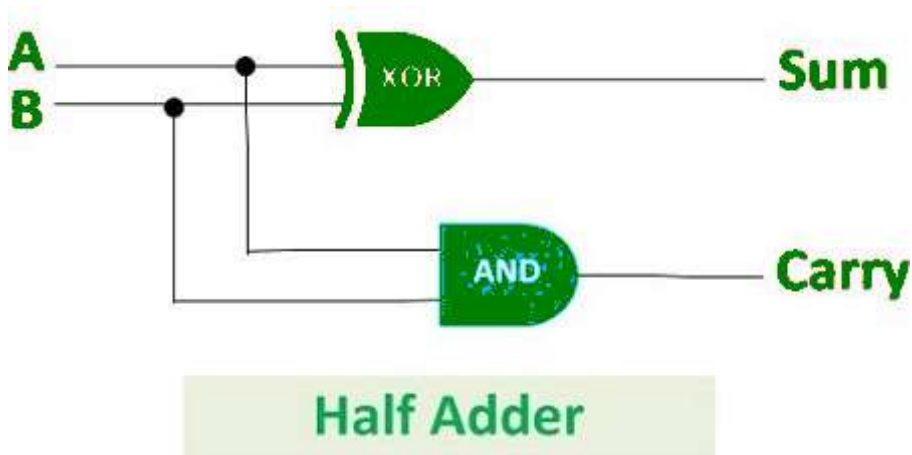
Half adder is a combinational circuit that performs simple addition of two binary numbers. The input variables designate the augend and addend bits; the output variables produce the sum and carry. It is necessary to specify two output variables because the result may consist of two binary digits.

$$\text{Sum} = A'B + AB' = A \oplus B \quad \text{Carry} = AB$$

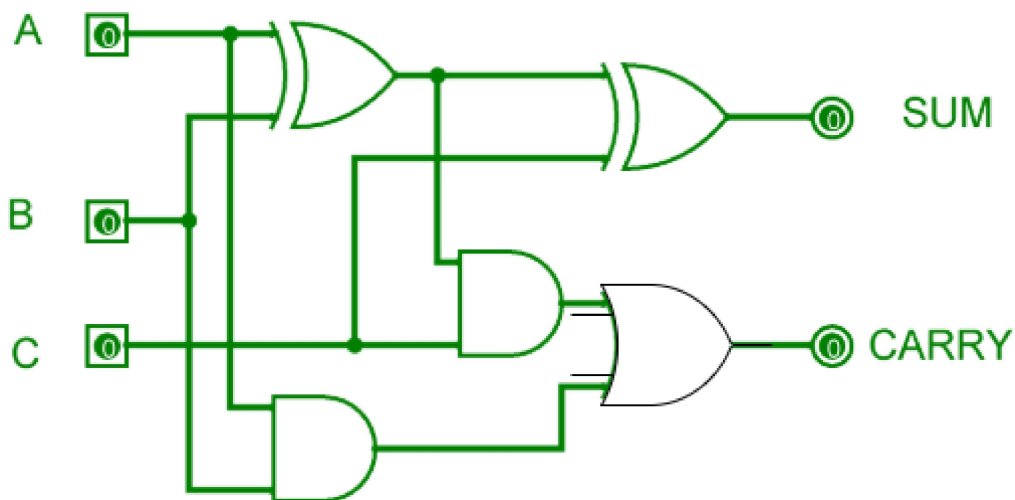
' Full Adder

Full adder is a digital circuit used to calculate the sum of three binary bits. It consists of three inputs and two outputs. Two of the input variables, denoted by A and B, represent the two significant bits to be added. The third input, Cin, represents the carry from the previous lower significant position. Two outputs are necessary because the arithmetic sum of three binary digits ranges in value from 0 to 3, and binary 2 or 3 needs two digits. The two outputs are sum and carry.

$$\text{Sum} = A'B'C_{in} + A'BC_{in}' + ABC_{in} + AB'C_{in}' = A \oplus B \oplus C_{in} \quad \text{Carry} = AB + AC_{in} + BC_{in}$$



› Figure -01 HALF ADDER



› Figure -02 FULL ADDER

› Procedure

Connect the supply (+5V) to the circuit Switch ON the main switch If the output is 1, then the led glows.

Program:

Program to design a half adder and full adder circuit and verify its truth table in quartus using Verilog programming.

Developed by: SHAVEDHA Y

RegisterNumber: 212221230095

HALF ADDER

```
module Adder(a,b,sum,carry);
```

```
input a,b;
```

```
output sum,carry;
```

```
xor(sum,a,b);
```

```
and(carry,a,b);
```

```
endmodule
```

FULL ADDER

```

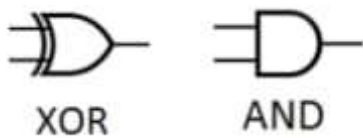
module FullAdder(a,b,c,sum,carry);
input a,b,c;
output sum,carry;
assign sum = ((a^b)^c);
assign carry = ((a&b)|(b&c)|(c&a));
endmodule

```

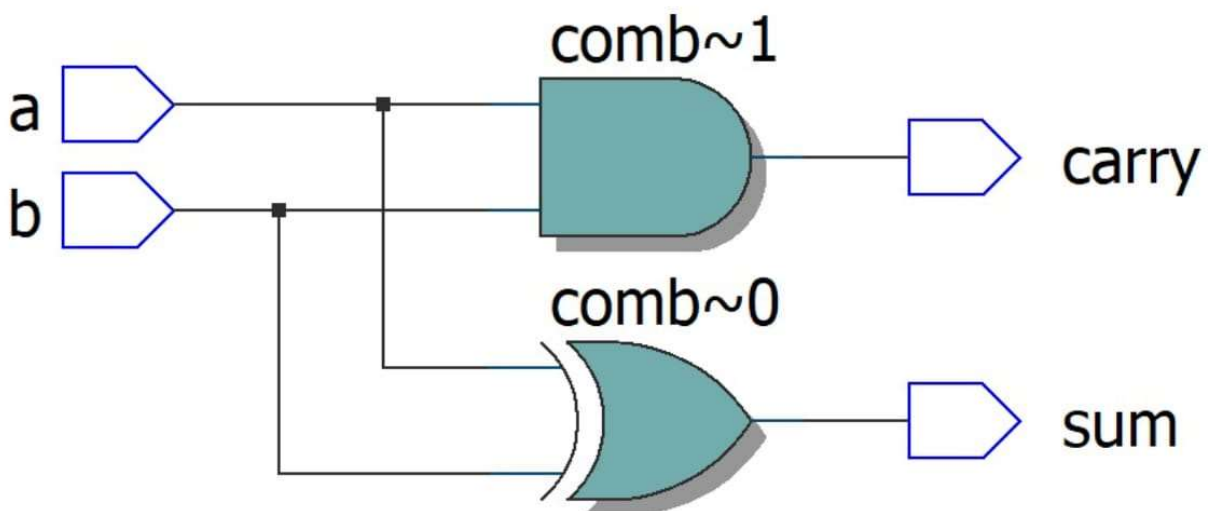
’ OUTPUT:

’ Half Adder :

’ Logic Symbol:



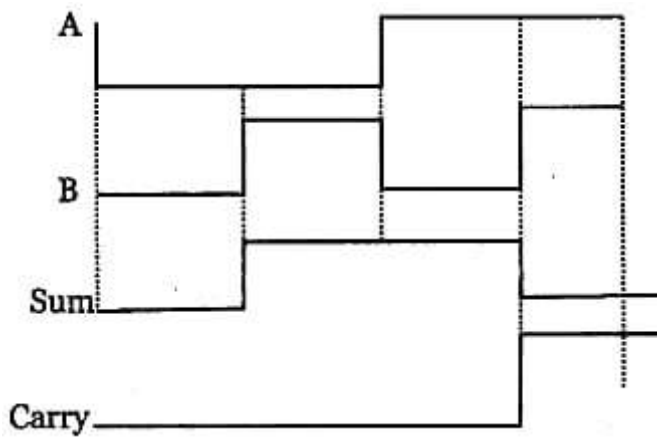
’ RTL Realization:



’ Truthtable:

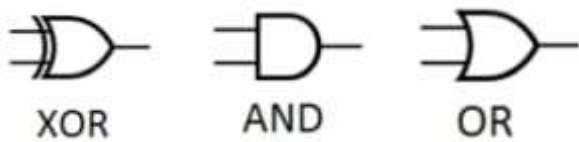
INPUTS		OUTPUTS	
a	b	Sum ($a \oplus b$)	Carry (a.b)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

’ Timing Diagram:

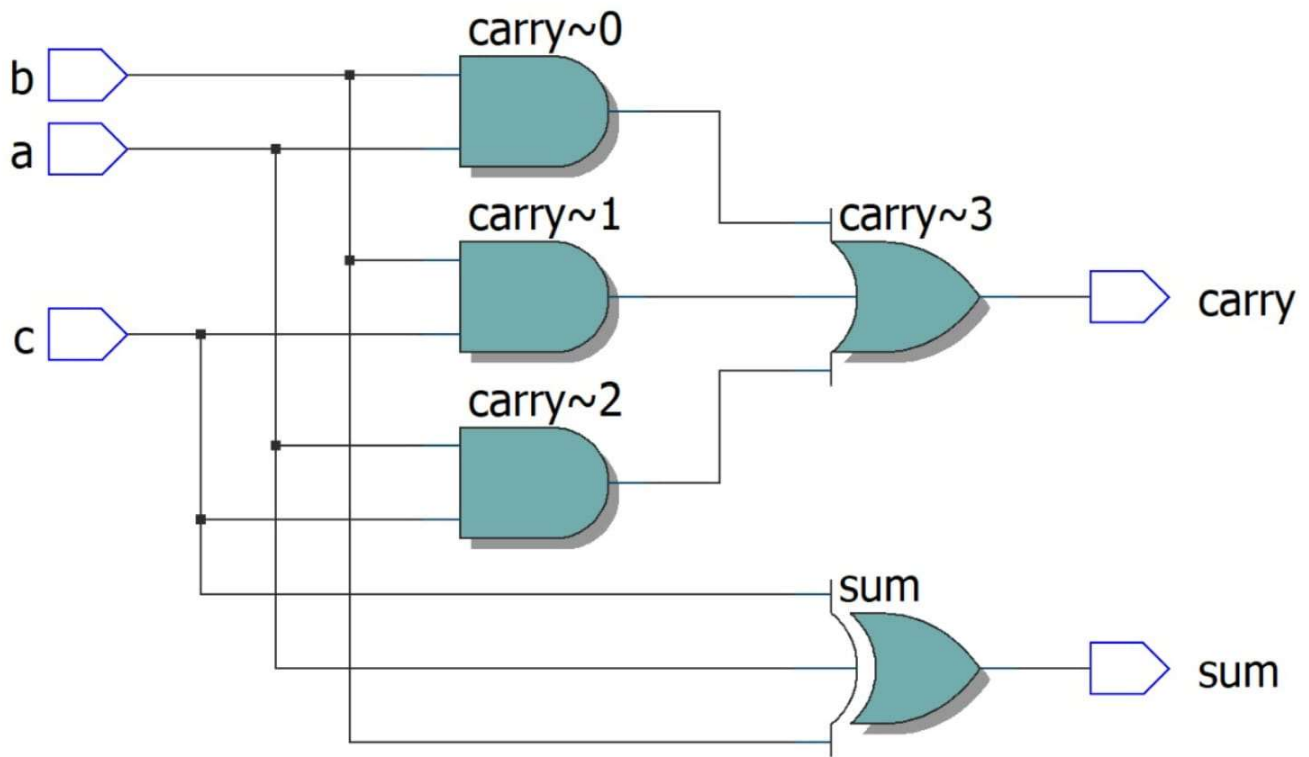


’ Full Adder :

’ Logic Symbol:



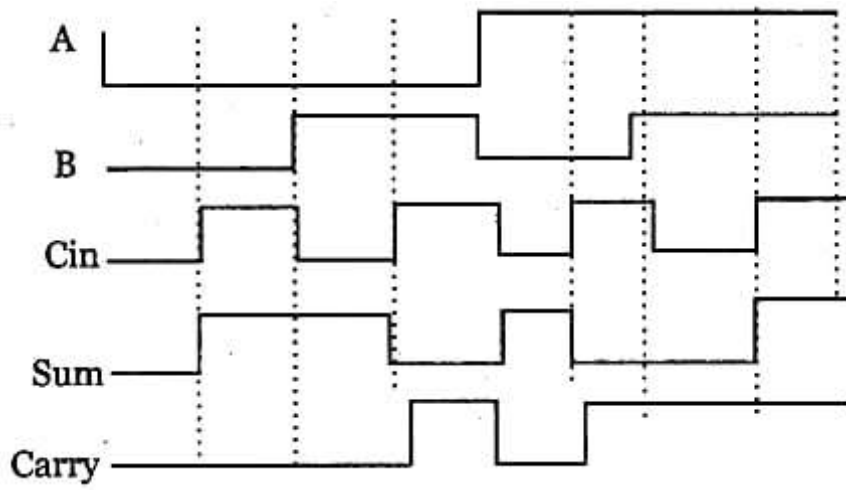
’ RTL Realization:



’ Truthtable:

INPUTS			OUTPUTS	
a	b	c	Sum ($a \oplus b \oplus c$)	Carry ($(a.b) + c(a \oplus b)$)
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

’ Timing Diagram:



’ **Result:**

Thus, a half adder and full adder circuit is designed to verify its truth table in Quartus using Verilog programming.