Nishalini-Ramesh Update README.md 6ec8a29 · now

82 lines (39 loc) · 2.57 KB

Preview Code Blame Raw

NAME : NISHALINI R

REG NO: 212224040222

EXP4 FULL ADDER SUBTRACTOR

AIM:

To design a Full Adder and Full Subtractor circuit and verify its truth table in Quartus using Verilog programming.

Equipments Required:

Hardware – PCs, Cyclone II , USB flasher

Software – Quartus prime

Full Adder and Full Subtractor

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.

Sum =A'B'Cin + A'BCin' + ABCin + AB'Cin' = A $\oplus$ B $\oplus$ Cin

Carry = AB + ACin + BCin

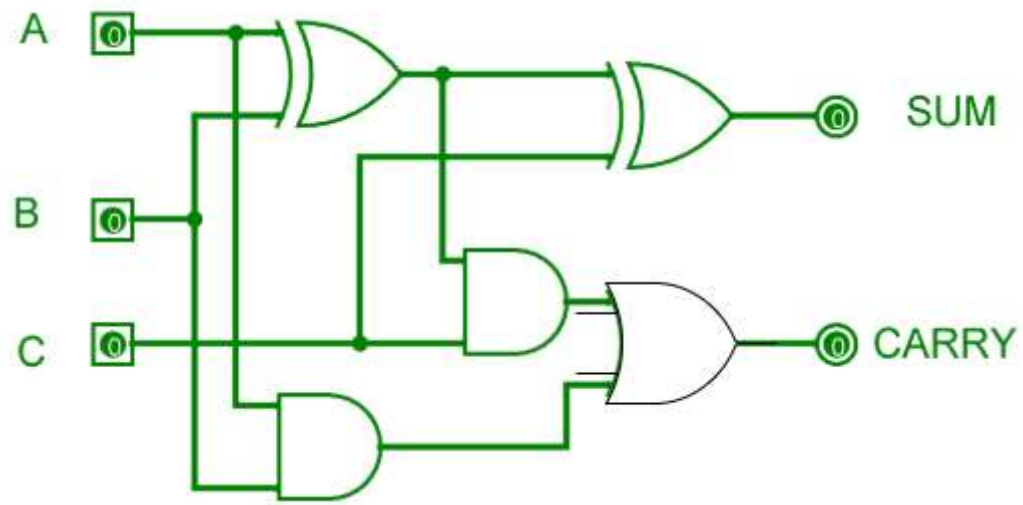
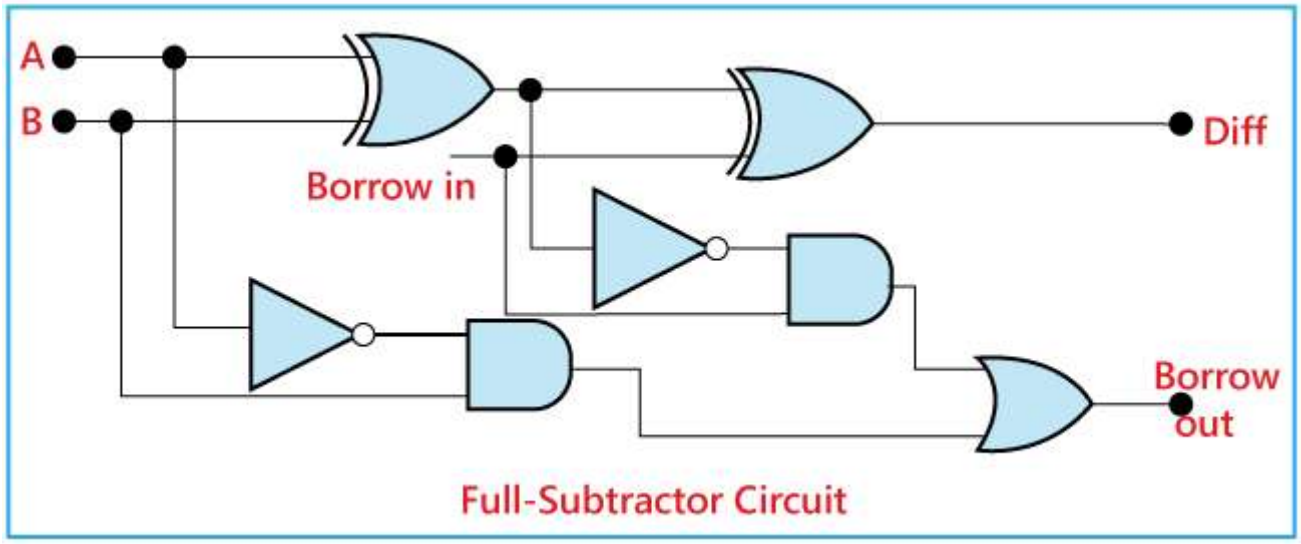


Figure -1 FULL ADDER

Full Subtractor

A full subtractor is a combinational circuit that performs subtraction involving three bits, namely minuend, subtrahend, and borrow-in . It accepts three inputs: minuend, subtrahend and a borrow bit and it produces two outputs: difference and borrow.



Diff = A ⊕ B ⊕ Bin

Borrow out = A'Bin + A'B + BBin

Truthtable

Procedure

1. Type the program in Quartus software.
2. Compile and run the program.
3. Generate the RTL schematic and save the logic diagram.
4. Create nodes for inputs and outputs to generate the timing diagram.
5. For different input combinations generate the timing diagram.

Write the detailed procedure here

Program:

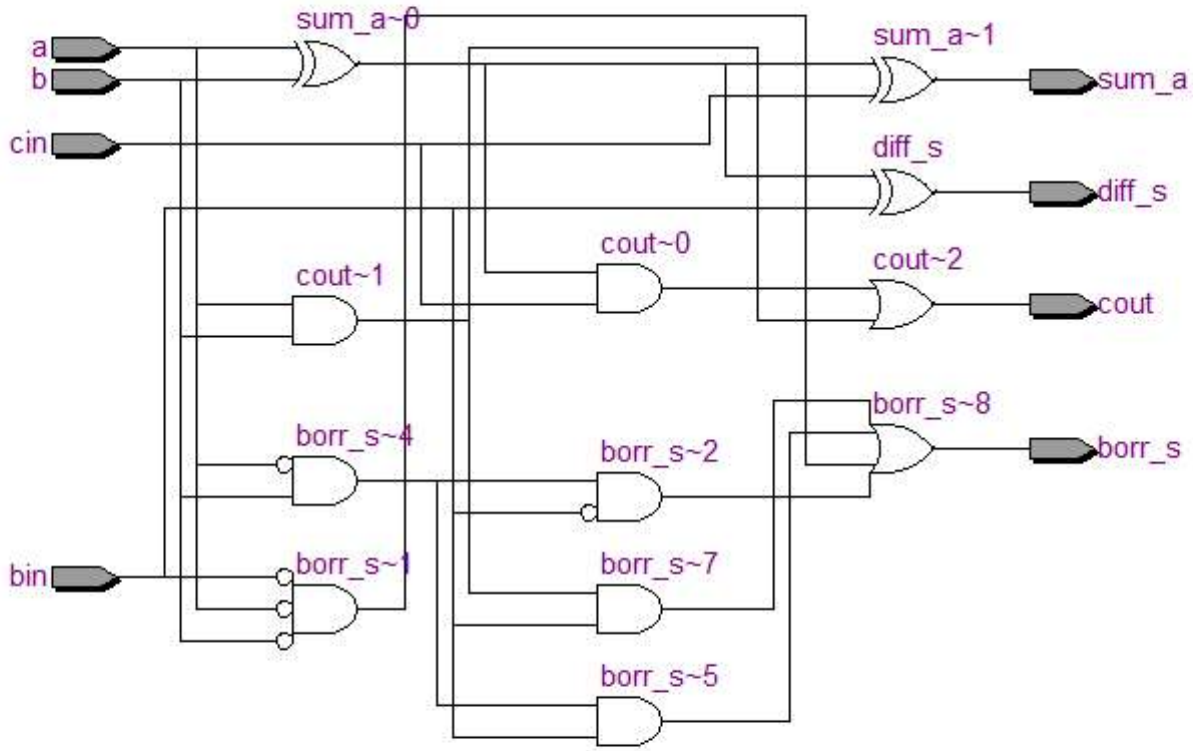
```
1  module exp3(a,b,sum_a,carry_a,diff_s,borrow_s);
2  input a,b;
3  output sum_a,carry_a,diff_s,borrow_s;
4  assign sum_a=a^b;
5  assign carry_a=a&b;
6  assign diff_s=a^b;
7  assign borrow_s=(~a&b);
8  endmodule
```

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

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

RTL


a
b
cin
bin
sum_a~0
cout~1
borr_s~4
borr_s~1
cout~0
borr_s~2
borr_s~7
borr_s~5
sum_a~1
diff_s
cout~2
borr_s~8
sum_a
diff_s
cout
borr_s

Waveform

```
module jkexp7(q,qbar,j,k,clock);
input j,k,clock;
output reg q;
output qbar;
always @ (posedge(clock))
begin
    q <=(j&(~q)) & ((~k)&q);
end
assign qbar=(~q);
endmodule
```

RESULT

Thus the Full Adder and Full Subtractor circuits are designed and the truth tables is verified using Quartus software.