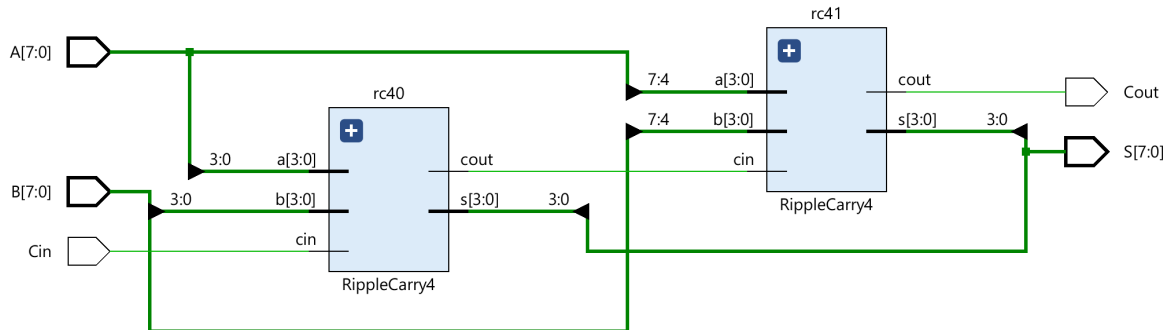


CSULB
CECS225

Create an 8-bit adder and an 8-bit subtracter

PROCEDURE: First import the halfadder, fulladder, RippleCarry4 module source files from the previous lab. *Be sure to use the exact same file names from the previous lab.*

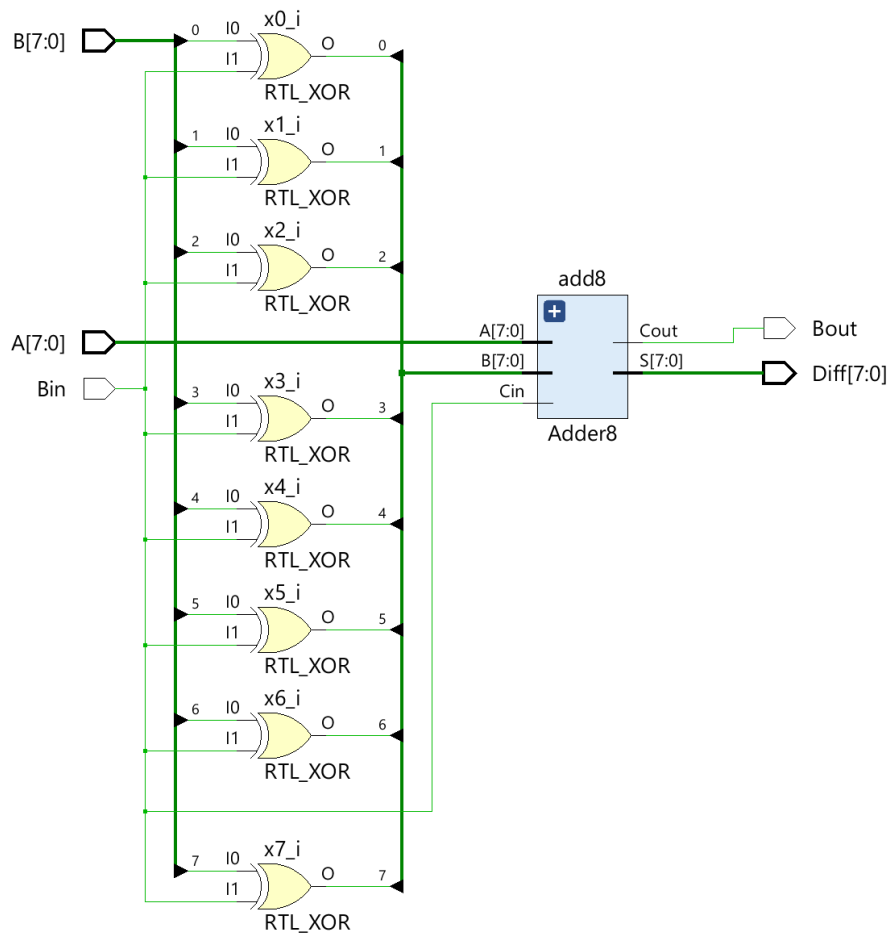
Next create an 8-bit Ripple carry adder using 2 of the 4-bit ripple carry adders according to the block diagram below:



Name your file and use the module skeleton below to get started:

```
module Adder8(
    input Cin,
    input [7:0] A,
    input [7:0] B,
    output [7:0] S,
    output Cout
);
    wire cout3;
    RippleCarry4
        rc40(Cin, A[3:0], B[3:0], S[3:0], cout3),
        rc41(
            cout3, A[7:4], B[7:4], S[7:4], Cout
        );
endmodule
```

Then create an **8-bit ripple subtracter** according to the block diagram below:



```

module Subtractor8(
    input Bin,
    input [7:0] A,
    input [7:0] B,
    output [7:0] Diff,
    output Bout
);
    wire [7:0] e;
    xor
        x0(e[0], A[0], B[0]),
        x1(e[1], A[1], B[1]),
        x2(e[2], A[2], B[2]),
        x3(e[3], A[3], B[3]),
        x4(e[4], A[4], B[4]),
        x5(e[5], A[5], B[5]),
        x6(e[6], A[6], B[6]),
        x7(e[7], A[7], B[7]);
    Adder8
        add8(Bin, e, A, B, Diff, Bout);
endmodule

```

Now let us test our design using the following testbench:

```

module TestBench();
    reg Add0Sub1;
    reg [7:0] a, b;
    wire [7:0] sum, difference;
    wire carry, borrow;

    integer i;
    Adder8 //instantiate 8-bit Adder
    uutA(Add0Sub1, a, b, sum, carry);

    Subtractor8 //instantiate 8-bit Subtractor
    uutS(Add0Sub1, a, b, difference, borrow);

    initial
    begin
        $display("\n\n");
        for(i = 0; i < 10; i = i + 1)
        begin
            {a, b, Add0Sub1} = $random;
            test_case;
        end//for
        $finish;
    end//initial

    task test_case;
    #1
    if(Add0Sub1 == 1'b0)
    begin
        $display("\nAdder Test");
        $display("a + b = %b + %b = %b, carry = %b", a, b, sum, carry);
    end//if
    else
    begin
        $display("\nSubtractor Test");
        $display("a - b = %b - %b = %b, Borrow = %b", a, b, difference, borrow);
    end//else
    endtask
endmodule

```

If everything works correctly then the following console output will be produced:

Adder Test

$a + b = 10011010 + 10010010 = 00101100$, carry = 1

Subtractor Test

$a - b = 10101111 - 01000000 = 01101111$, Borrow = 1

Subtractor Test

$a - b = 01101011 - 00000100 = 01100111$, Borrow = 1

Subtractor Test

$a - b = 00101011 - 00110001 = 11111010$, Borrow = 0

Subtractor Test

$a - b = 10111101 - 10000110 = 00110111$, Borrow = 1

Subtractor Test

$a - b = 11001100 - 11000110 = 00000110$, Borrow = 1

Subtractor Test

$a - b = 01000010 - 00110010 = 00010000$, Borrow = 1

Adder Test

$a + b = 10101001 + 00001001 = 10110010$, carry = 0

Subtractor Test

$a - b = 11110001 - 10000000 = 01110001$, Borrow = 1

Subtractor Test

$a - b = 11100110 - 10000110 = 01100000$, Borrow = 1

Study your output. Is it correct? Explain

You may use %d -instead or besides %b (Then add another variable for the output)- to output the values in decimal

Copy past your code, a screenshot of your timeline and output with a link to a video uploaded to YouTube explaining your code and your output. Run the simulation when capturing your screen.