Digital System Design

Submission Date: 16th October, 2017

1. **Write RTL VERILOG Code to implement the logic shown in figure 1 representing a full adder where A, B are 1 bit input whereas Cin 1 bit carry. Write another module for implementing a 4-bit adder, use "module for figure-1" to implement the four bit adder shown in figure 2. Write a stimulus to exhaustively test the 4-bit adder.**

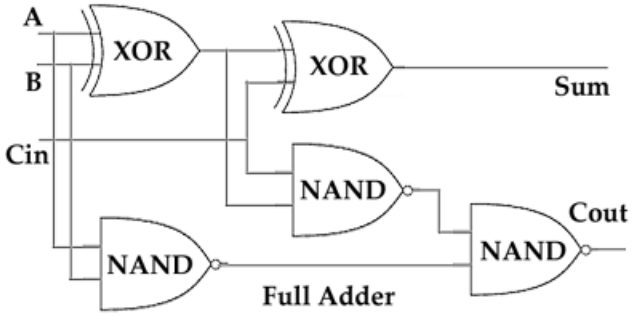


Figure 1: Logic representing Full-Adder

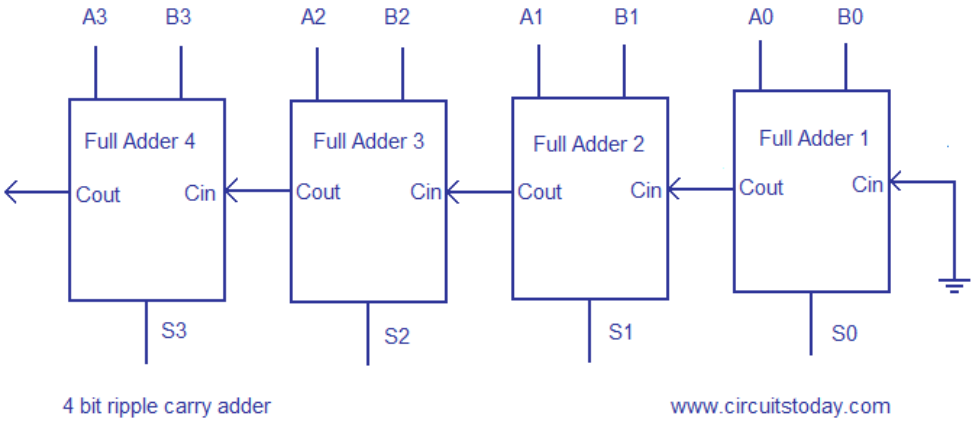


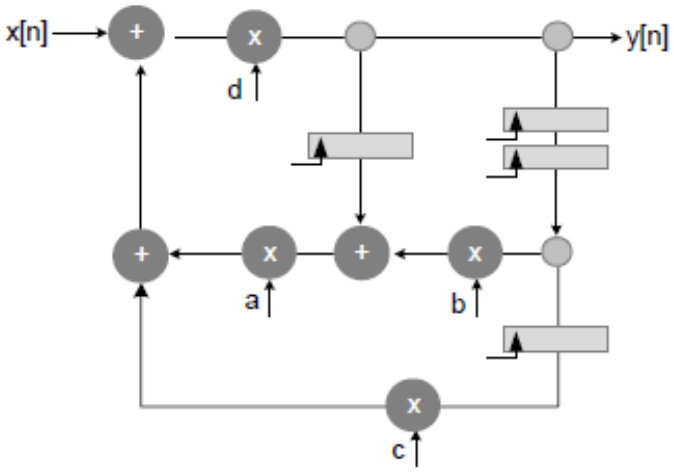
Figure 2: 4-bit Carry Ripple Adder

2. **Draw RTL diagram corresponding to the given difference equation also write its RTL VERILOG code. Write the stimulus for your design for verification. Assume 8 bit input and 16 bit output lines.**

$$y[n] = a * x[n-1] + b * x[n-3] + c * y[n-1] - d * y[n-2]$$

Assume a, b and c as any constant.

3. **Write RTL VERILOG code to implement the design shown below. Also design a stimulus to verify your RTL code of the design. Assume a = 0.5, b = 0.75, c= 0.25 and d = 0.125, do not use multiplication in the design.**



4. **Draw RTL diagram representing the following VERILOG code. Clearly specify all the details (signals, bus width, directions etc) in the RTL diagram.**

```
module Question_4(a, b, sel, clk, rst_n, c);
input [6:0] a, b;
input clk, rst_n;
input [2:0] sel;
output reg [6:0] c;
reg [6:0] x1, x2, x3, t_x1, t_x2, t_x3;
always @ (posedge clk or negedge rst_n)
begin
    if (!rst_n)
    begin
            x1<= 0; x2<=0; x3<=0;
    end
    else
    begin
            x1<= t_x1; x2<= t_x2; x3<= t_x3;
    end
end
```

```
always@(*)
begin
    if (sel == 0)
    begin
            t_x1 <= x2;
            t_x2 <= x1;
            t_x3 <= 1;
    end
    else if (sel == 1)
    begin
            t_x1 <= x3;
            t_x2 <= x2;
            t_x3 <= x2;
    end
    else
    begin
            t_x1 <= 0;
            t_x2 <= x3;
            t_x3 <= 0;
    end
end

always@(*)
begin
    case(|(a&b))
    0: c = x1<<2;
    1: c = x2<<2;
end
endmodule
```