Digital Systems I

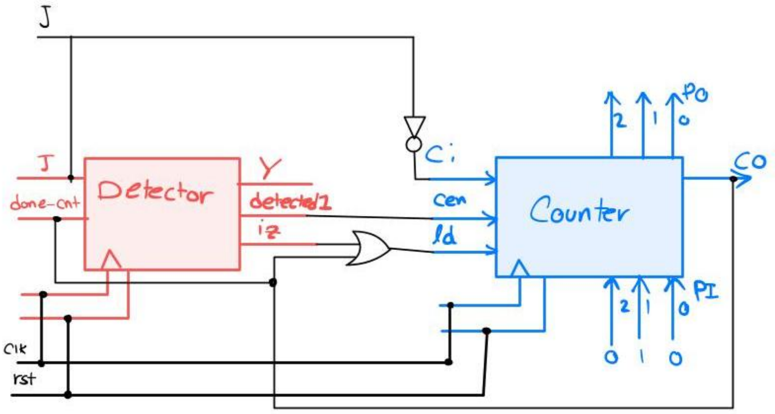
Computer Assignment #4

FSMs and Counters

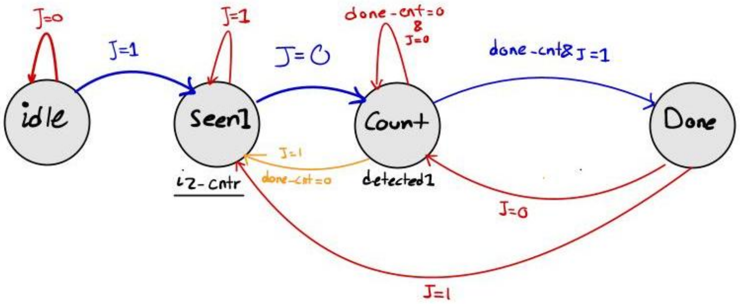
Amirali Dehghani

SID: 810102443

a.



b.



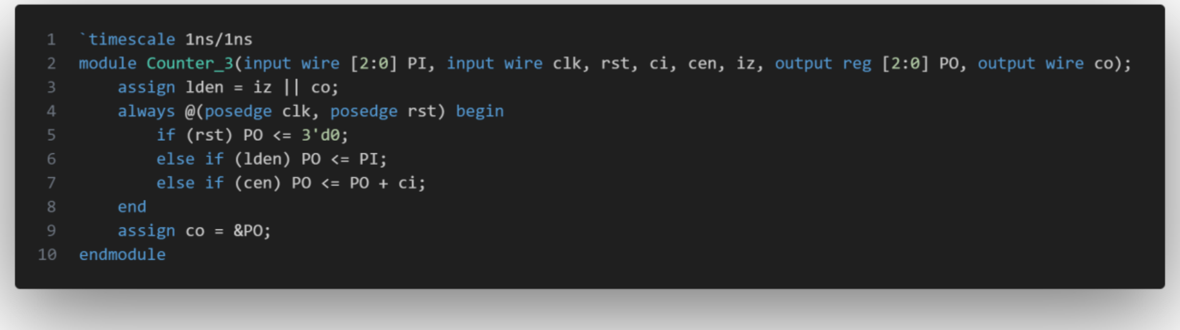
c.

My counter has several inputs and outputs.

- For carry in, because I have to count zeros, I connected ~J to carry in input so it only counts up when J is zero.
- Because we need a Frequency divider (Divide by 6 circuit) I have an parallel input and a load signal. Parallel input is 8 – 6 = 2 -> 010 and the load signal issues when izC from detector is 1 or Carry out is 1(that means counting is finished and we have to initialize back counter to the normal one.)
- I also have count enable signal on my counter and it is connected to 1Detected signal out from FSM so it can count only when 1 input get's detected.
- PO is optional but I kept it because it helped me in debugging the code.

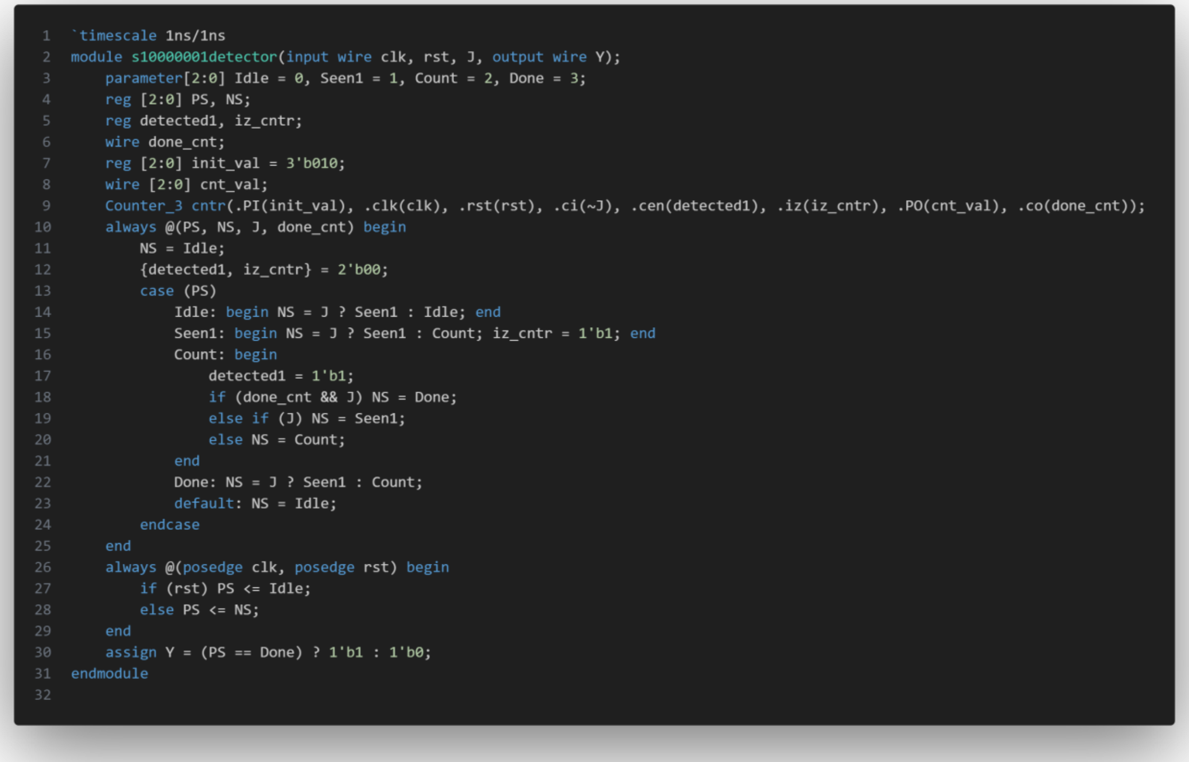
d.

This is the code for my 3-bit counter.



```verilog
`timescale 1ns/1ns
module Counter_3(input wire [2:0] PI, input wire clk, rst, ci, cen, iz, output reg [2:0] PO, output wire co);
    assign lden = iz || co;
    always @(posedge clk, posedge rst) begin
        if (rst) PO <= 3'd0;
        else if (lden) PO <= PI;
        else if (cen) PO <= PO + ci;
    end
    assign co = &PO;
endmodule
```

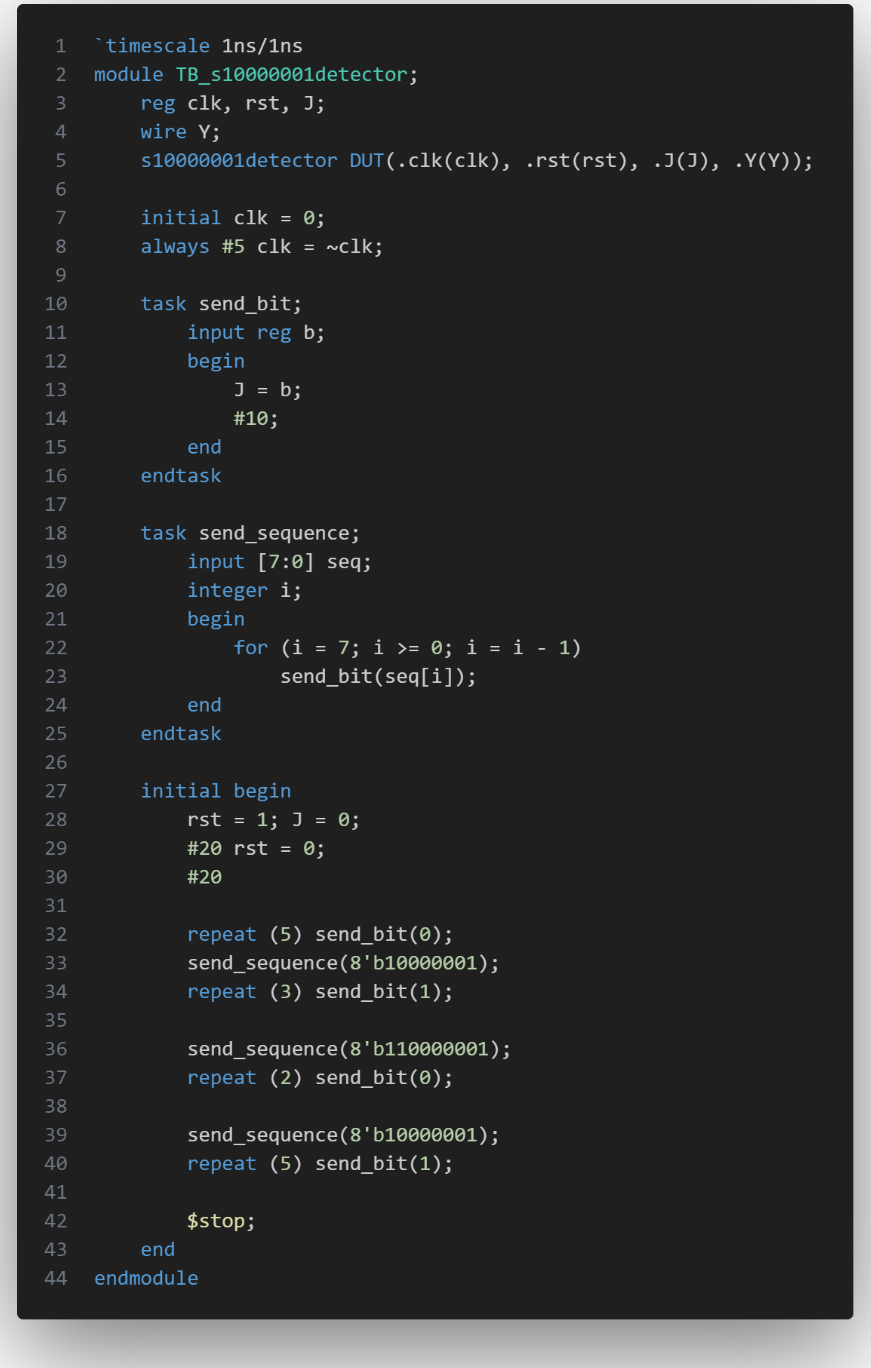
And also this is for the full sequence detector.



```verilog
`timescale 1ns/1ns
module s10000001detector(input wire clk, rst, J, output wire Y);
    parameter[2:0] Idle = 0, Seen1 = 1, Count = 2, Done = 3;
    reg [2:0] PS, NS;
    reg detected1, iz_cntr;
    wire done_cnt;
    reg [2:0] init_val = 3'b010;
    wire [2:0] cnt_val;
    Counter_3 cntr(.PI(init_val), .clk(clk), .rst(rst), .ci(~J), .cen(detected1), .iz(iz_cntr), .PO(cnt_val), .co(done_cnt));
    always @(PS, NS, J, done_cnt) begin
        NS = Idle;
        {detected1, iz_cntr} = 2'b00;
        case (PS)
            Idle: begin NS = J ? Seen1 : Idle; end
            Seen1: begin NS = J ? Seen1 : Count; iz_cntr = 1'b1; end
            Count: begin
                detected1 = 1'b1;
                if (done_cnt && J) NS = Done;
                else if (J) NS = Seen1;
                else NS = Count;
            end
            Done: NS = J ? Seen1 : Count;
            default: NS = Idle;
        endcase
    end
    always @(posedge clk, posedge rst) begin
        if (rst) PS <= Idle;
        else PS <= NS;
    end
    assign Y = (PS == Done) ? 1'b1 : 1'b0;
endmodule
```

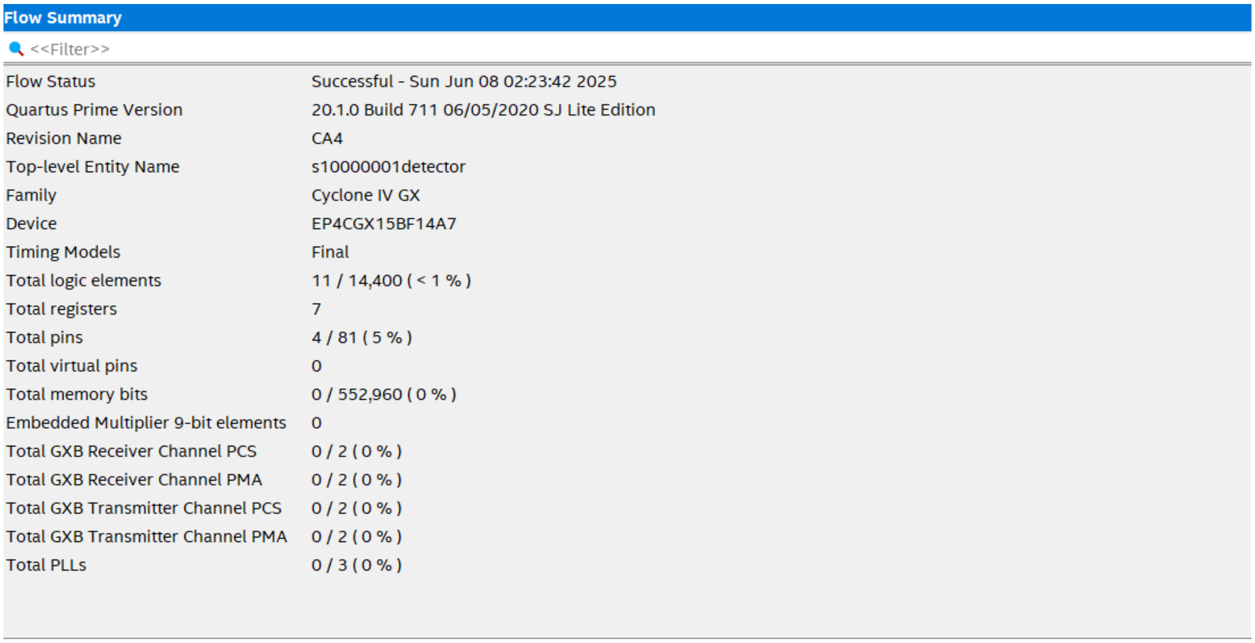
e.

Code



```
`timescale 1ns/1ns
module TB_s10000001detector;
    reg clk, rst, J;
    wire Y;
    s10000001detector DUT(.clk(clk), .rst(rst), .J(J), .Y(Y));

    initial clk = 0;
    always #5 clk = ~clk;

    task send_bit;
        input reg b;
        begin
            J = b;
            #10;
        end
    endtask

    task send_sequence;
        input [7:0] seq;
        integer i;
        begin
            for (i = 7; i >= 0; i = i - 1)
                send_bit(seq[i]);
        end
    endtask

    initial begin
        rst = 1; J = 0;
        #20 rst = 0;
        #20

        repeat (5) send_bit(0);
        send_sequence(8'b10000001);
        repeat (3) send_bit(1);

        send_sequence(8'b110000001);
        repeat (2) send_bit(0);

        send_sequence(8'b10000001);
        repeat (5) send_bit(1);

        $stop;
    end
endmodule
```

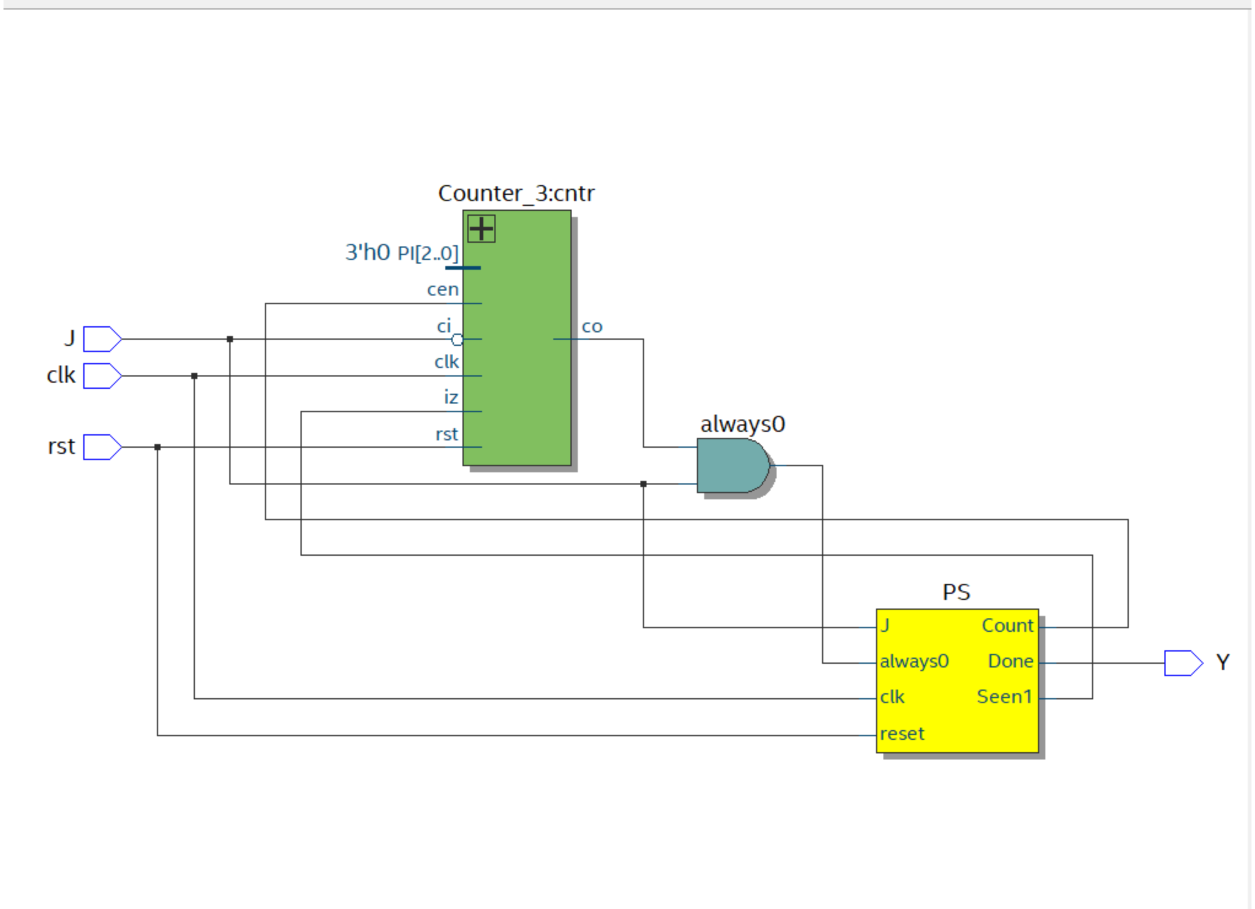
f.

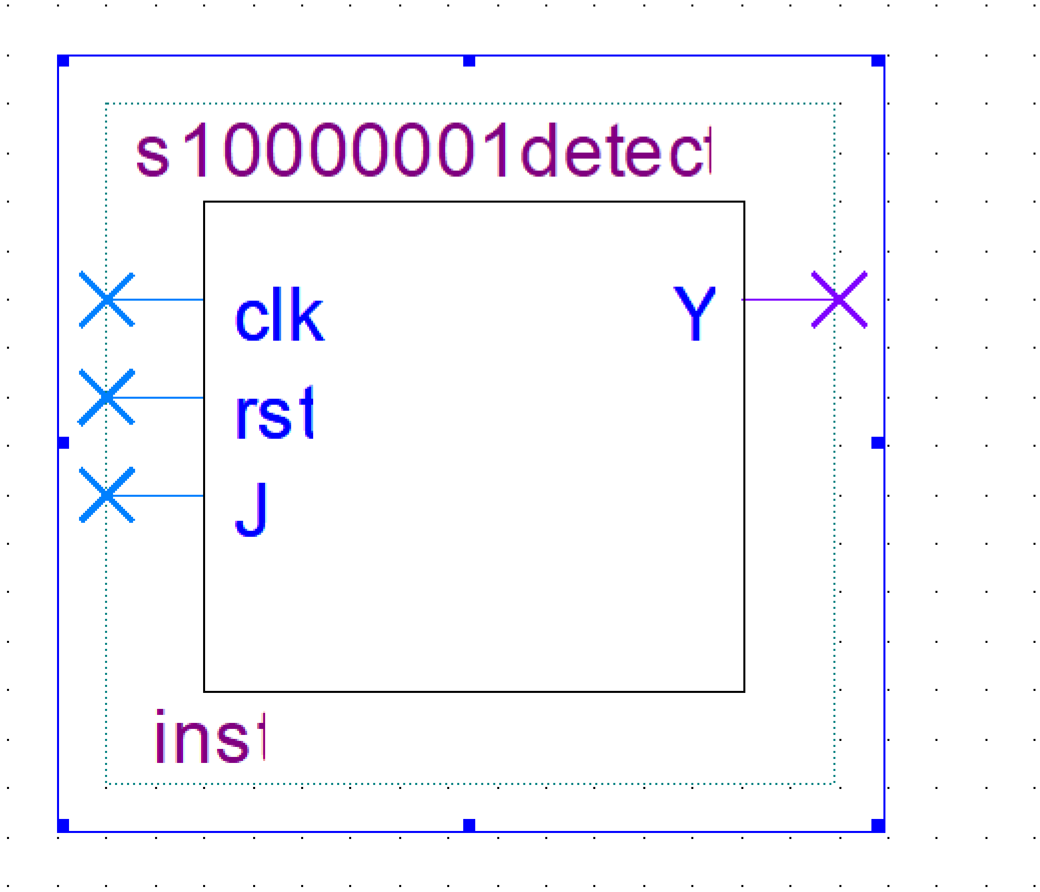


Flow Summary

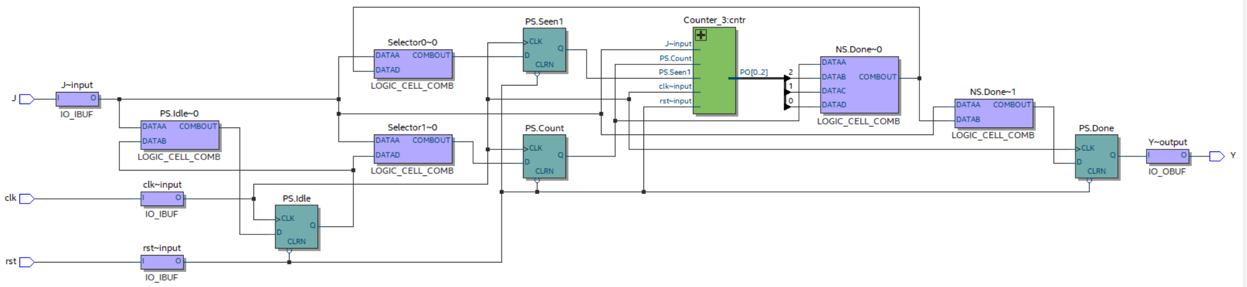
<<Filter>>

Flow Status	Successful - Sun Jun 08 02:23:42 2025
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	CA4
Top-level Entity Name	s10000001detector
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	11 / 14,400 (< 1 %)
Total registers	7
Total pins	4 / 81 (5 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)

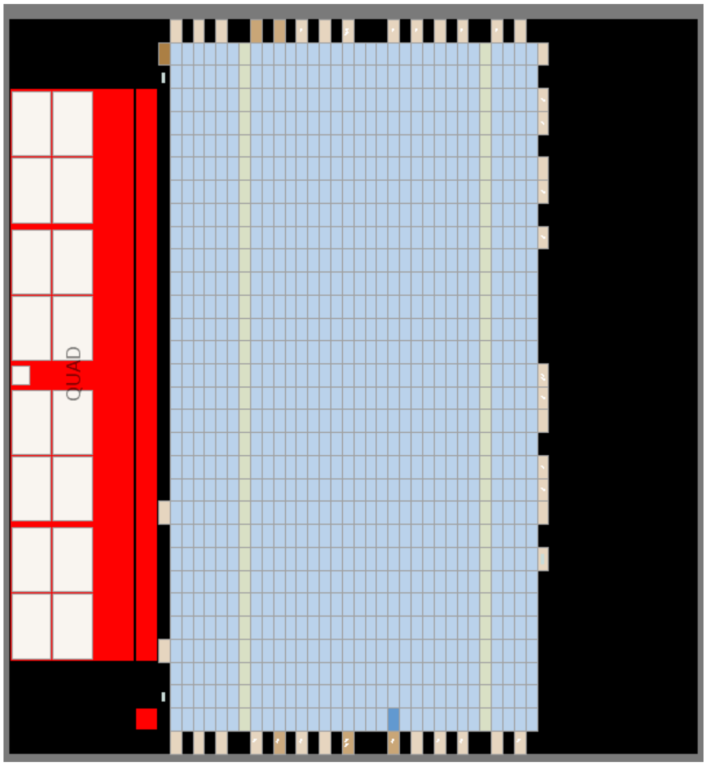


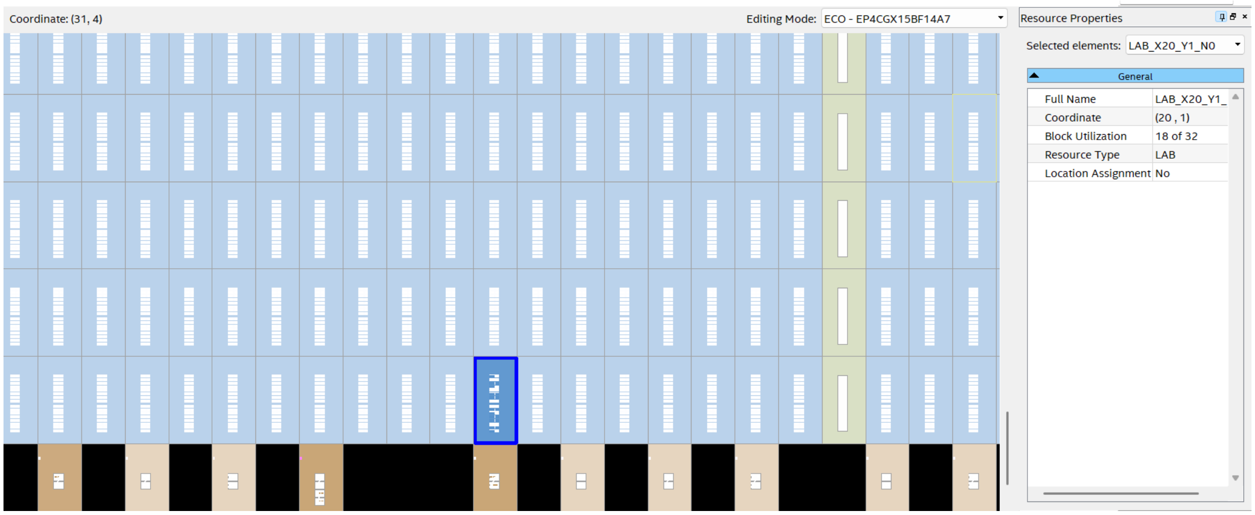


s1000001detect
clk
rst
J
Y
inst



J
J~input
IO_IBUF
PS.Idle~0
DATAA
DATAB
COMBOUT
LOGIC_CELL_COMB
clk
clk~input
IO_IBUF
rst
rst~input
IO_IBUF
PS.Idle
CLK
D
Q
CLRN
Selector0~0
DATAA
DATAD
COMBOUT
LOGIC_CELL_COMB
Selector1~0
DATAA
DATAD
COMBOUT
LOGIC_CELL_COMB
PS.Seen1
CLK
D
Q
CLRN
PS.Count
CLK
D
Q
CLRN
Counter_3:cntr
J~input
PS.Count
PS.Seen1
clk~input
rst~input
PO[0..2]
2
1
0
NS.Done~0
DATAA
DATAB
DATAC
DATAD
COMBOUT
LOGIC_CELL_COMB
NS.Done~1
DATAA
DATAB
COMBOUT
LOGIC_CELL_COMB
PS.Done
CLK
D
Q
CLRN
Y~output
IO_OBUF
Y




Coordinate: (31, 4)

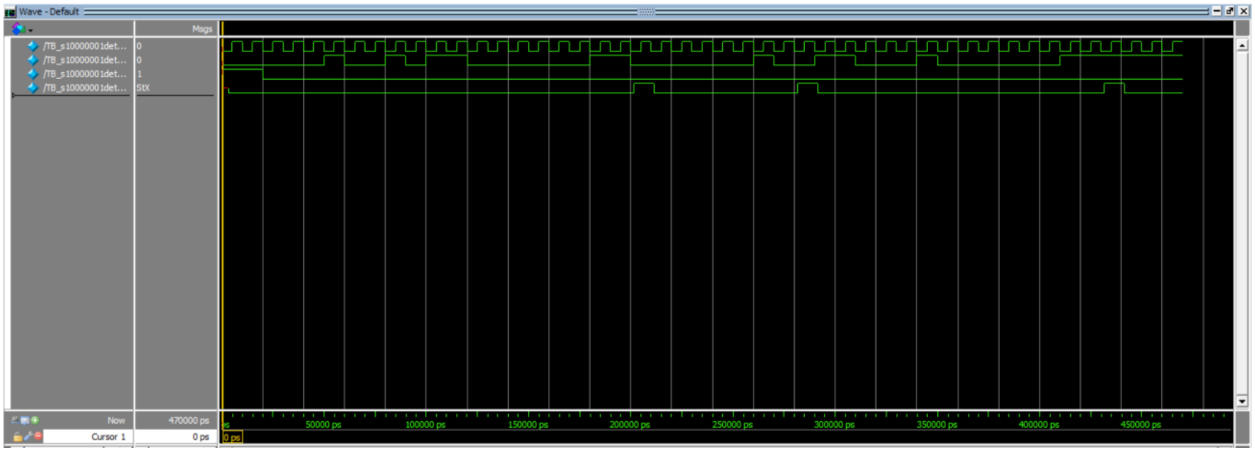
Editing Mode: ECO - EP4CGX15BF14A7

Resource Properties

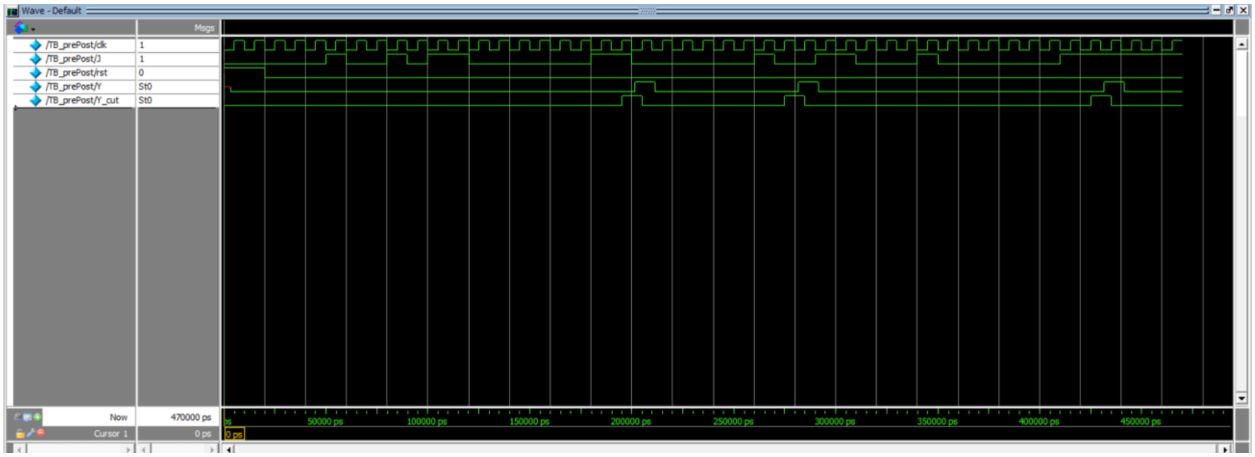
Selected elements: LAB_X20_Y1_N0

General

Full Name	LAB_X20_Y1_
Coordinate	(20 , 1)
Block Utilization	18 of 32
Resource Type	LAB
Location Assignment	No



Testing both:


Wave - Default
Msgs
/TB_prePost/clk 1
/TB_prePost/J 1
/TB_prePost/rst 0
/TB_prePost/Y St0
/TB_prePost/Y_cut St0
Now 470000 ps
Cursor 1 0 ps
ps
50000 ps
100000 ps
150000 ps
200000 ps
250000 ps
300000 ps
350000 ps
400000 ps
450000 ps
0 ps