Digital Systems : Lab 10

Nimitt

Pratham Sharda

Clock Divider

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/02/2024 03:07:17 PM
// Design Name:
// Module Name: clock_divider
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module clock_divider(
    input wire clk,
    input wire rst,
    output slow_clk
    );
    reg [31:0] counter; // 27-bit counter for dividing the clock
    always @(posedge clk)
               begin
                counter <= counter + 1;
               end
    assign slow_clk = counter[27]

endmodule
```

Moore 10010 Detector

```
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/02/2024 03:07:17 PM
// Design Name:
// Module Name: moore_10010
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
///////////////////////////////////////////////////////////////


module moore_10010 (main_clk, w, reset, z, slow_clk);
    input main_clk, w, reset;
    output z;
    output slow_clk;
    reg y, Y;
    parameter A = 3'b000, B = 3'b001, C = 3'b010, D = 3'b011, E

    wire slow_clk;
    clock_divider inst(.clk(main_clk), .rst(reset),.slow_clk(sl


    always @(w or y)
        begin
            case (y)
            A: if (w == 0)
            begin
            Y = A;
            end
            else
            begin
            Y = B;
            end

            B: if (w == 0)
            begin
            Y = C;
            end
            else
            begin
            Y = B;
            end

            C: if (w == 0)
            begin
            Y = D;
            end
            else
            begin
            Y = B;
            end

            D: if (w == 0)
            begin
            Y = A;
            end
            else
            begin
            Y = E;
            end

            E: if (w == 0)
            begin
            Y = F;
            end
            else
            begin
            Y = B;
            end

            F: if (w == 0)
            begin
            Y = D;
            end
            else
            begin
            Y = B;
            end
            endcase

        end

    always @(negedge slow_clk or negedge reset)
        begin
            if (reset == 0)
```

```
            begin
            y <= A;
            end
            else
            begin
            y <= Y;
            end
        end

    assign z = (y == F);

endmodule
```

```
test bench

module 10010_tb();
    reg main_clk, w, reset;
    wire z;

    initial begin
        main_clk = 0;
        forever #5 main_clk = ~main_clk;
    end

    initial begin
        w = 0; reset = 1;
        #7.5;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        $finish();
    end
endmodule
```

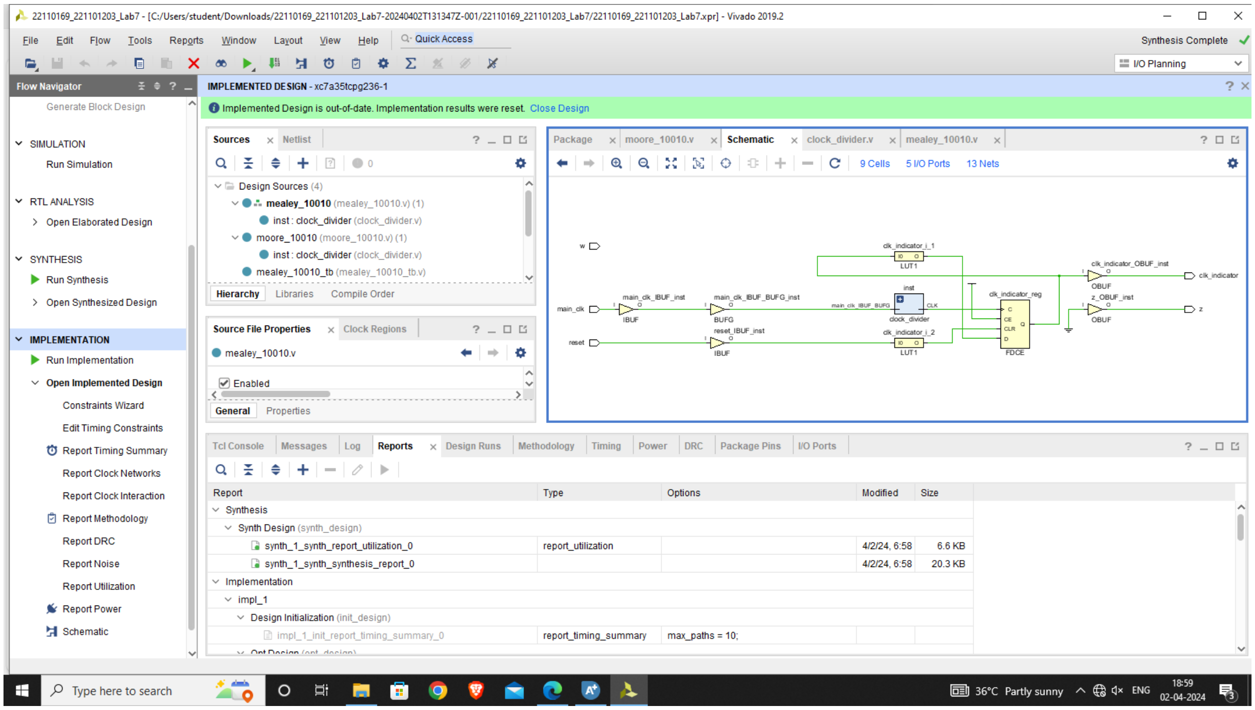
Mealey 10010 Detector

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 04/02/2024 03:07:17 PM
// Design Name:
// Module Name: mealey_10010
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
```

```
///////////////////////////////////////////////////////////////

module mealey_10010 (main_clk, w, reset, z, slow_clk);
    input main_clk, w, reset;
    output z;
    output slow_clk;
    reg z, y, Y;
    parameter A = 3'b000, B = 3'b001, C = 3'b010, D = 3'b011, E

    wire slow_clk;
    clock_divider inst(.clk(main_clk), .rst(reset),.slow_clk(sl

    always @(w or y)
        begin
            case (y)
            A: if (w == 0)
            begin
            Y = A; z = 0;
            end
            else
            begin
            Y = B; z = 0;
            end

            B: if (w == 0)
            begin
            Y = C; z = 0;
            end
            else
            begin
            Y = B; z = 0;
            end

            C: if (w == 0)
            begin
            Y = D; z = 0;
            end
            else
            begin
            Y = B; z = 0;
            end

            D: if (w == 0)
            begin
            Y = A; z = 0;
            end
            else
            begin
            Y = E; z = 0;
            end

            E: if (w == 0)
            begin
            Y = C; z = 1;
            end
            else
            begin
            Y = B; z = 0;
            end
            endcase
        end

    always @(negedge slow_clk or negedge reset)
        begin
            if (reset == 0)
                begin
                    y <= A;
                end
            else
                begin
                    y <= Y;
                end
        end
endmodule
```

```
module 10010_tb();
    reg main_clk, w, reset;
    wire z;

    initial begin
        main_clk = 0;
        forever #5 main_clk = ~main_clk;
    end

    initial begin
        w = 0; reset = 1;
        #7.5;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        w = 0; reset = 0;
        #10;
        w = 1; reset = 0;
        #10;
        $finish();
    end
endmodule
```

FPGA Implementation



22110169_221101203_Lab7 - [C:/Users/student/Downloads/22110169_221101203_Lab7-20240402T131347Z-001/22110169_221101203_Lab7/22110169_221101203_Lab7.xpr] - Vivado 2019.2

IMPLEMENTED DESIGN - xc7a35tcpg236-1

Implemented Design is out-of-date. Implementation results were reset. Close Design

Report	Type	Options	Modified	Size
Synthesis				
Synth Design (synth_design)				
synth_1_synth_report_utilization_0	report_utilization		4/2/24, 6:58	6.6 KB
synth_1_synth_synthesis_report_0			4/2/24, 6:58	20.3 KB
Implementation				
impl_1				
Design Initialization (init_design)				
impl_1_init_report_timing_summary_0	report_timing_summary	max_paths = 10;		

Constraints file

```
set_property IOSTANDARD LVCMOS33 [get_ports clk_indicator]
set_property IOSTANDARD LVCMOS33 [get_ports main_clk]
set_property IOSTANDARD LVCMOS33 [get_ports reset]
set_property IOSTANDARD LVCMOS33 [get_ports w]
set_property IOSTANDARD LVCMOS33 [get_ports z]
set_property PACKAGE_PIN L1 [get_ports clk_indicator]
set_property PACKAGE_PIN P1 [get_ports z]
set_property PACKAGE_PIN W5 [get_ports main_clk]
set_property PACKAGE_PIN R2 [get_ports reset]
set_property PACKAGE_PIN T1 [get_ports w]
```