CS303 // PRE-LAB #4 Report

Berna Yıldıran 26431

Circuit Flow

The module "Mod7":

Inputs:

- `clk`: a clock input signal
- `reset`: a reset input signal
- `enable`: an enable input signal
- `configure`: a 3-bit configuration input signal
- `serial_in`: a 32-bit serial input signal

Output:

- `out`: a 3-bit mod 7 result output signal
- `cnt_out`: an output signal indicating when the count value has reached the configuration value
- `led`: an output signal that toggles when `cnt_out` is 1

Internal signals:

- `x`: a 32-bit register used to store a value that will be modified in the code
- `cnt`: a 5-bit register used to store a count value

Step by Step Circuit:

1. The module has an `always` block that is triggered on the rising edge of either the `clk` or `reset` signals. This means that the block of code inside the `always` block will execute every time either of these signals makes a transition from 0 to 1.
2. The `always` block begins by checking the value of the `reset` signal. If `reset` is 1, the code initializes the values of `x`, `cnt`, `out`, `cnt_out`, and `led` to 0.
3. If `reset` is 0, the code increments the count value stored in `cnt`, and checks if it is equal to the `configure` value. If it is, the code resets the count value to 0 and sets `cnt_out` to 1.
4. The code then checks the value of `cnt_out`. If it is 1, the code toggles the value of `led` and sets `cnt_out` back to 0.
5. Finally, the code concatenates the lower 30 bits of `x` with the `serial_in` value and stores the result in `x`. It then performs a modulo 7 operation on `x` and stores the result in `out`.

Verilog: Design Sources

The Verilog source code of the “Mod7” circuit:

```
`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////
// Company: Sabanci University PRE-LAB 4
// Engineer: Berna Yildiran
//////////////////////////////////////////////////////////////////////////////////

module Mod7(
    input clk,
    input reset,
    input enable,

    // 3-bit configuration input
    input [2:0] configure,

    // 32-bit serial input
    input [31:0] serial_in,

    // 3-bit mod 7 result output
    output reg [2:0] out,
    output reg cnt_out,
    output reg led
    );

// 32-bit register used to store a value that will modified in the code
reg [31:0] x;

// 5-bit register used to store a count value
reg [4:0] cnt;

// Always block triggered on the rising edge of the clock or reset signals
always @(posedge clk or posedge reset)
    begin
        // Check the value of the reset signal
        // If reset is 1, initialize values to 0
        if (reset) begin
          x <= 0;
          cnt <= 0;
          out <= 0;
          cnt_out <= 0;
          led <= 0;
        end

        // If reset is 0, execute the following code
        else begin
          // Increment the count value
          cnt <= cnt + 1;

          // Check if the count value is equal to the configuration value
          // If so, reset the count value to 0 and set cnt_out to 1
          if (cnt == configure) begin
            cnt <= 0;
            cnt_out <= 1;
          end

          // If cnt_out is 1, toggle the value of led
          if (cnt_out) begin
            cnt_out <= 0;
            led <= ~led;
          end
```

```
          // Concatenate the lower 30 bits of x with the serial input value and store the result in x
          // Perform a mod 7 operation on x and store the result in out
          x <= {x[30:0], serial_in};
          out <= x % 7;
        end
    end
endmodule
```

Verilog: Simulation Sources

The simulation code of the circuit is written as the following:

```
`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////
// Company: Sabanci University
// Engineer: Berna Yildiran
//////////////////////////////////////////////////////////////////////////////////

module sim_Mod7;
  reg clk;
  reg reset;
  reg enable;
  reg [2:0] configure;
  reg [31:0] serial_in;
  wire [2:0] out;
  wire cnt_out;
  wire led;

  // Instantiate the div7 module
  Mod7 UUT (
    .clk(clk),
    .reset(reset),
    .enable(enable),
    .configure(configure),
    .serial_in(serial_in),
    .out(out),
    .cnt_out(cnt_out),
    .led(led)
  );

  // Initialize inputs
  initial begin
    clk = 0;
    reset = 0;
    enable = 1;
    configure = 0;
    serial_in = 0;
  end

  // Generate clock signal
  always #5 clk = ~clk;

  // Test case 1: Reset
  initial begin
    #100;
    reset = 1;
    #100;
    reset = 0;
  end
```

```
  initial begin
      // Test case 2: Mod 7 of a 4-bit number
      #200;
      serial_in = 4'b0010; //2 --> 1
      #100;
      serial_in = 4'b0101; //5 --> 2
      #100;
      serial_in = 4'b1010; //10 --> 3
      #100;
      serial_in = 4'b1110; //14 --> 0

      //Test case 3: Mod 7 of a 5-bit number
      #100;
      serial_in = 5'b01011; //11 --> 4
      #100;
      serial_in = 5'b10011; //19 --> 5
      #100;
      serial_in = 5'b00011; //20 --> 6
      #100;
      serial_in = 5'b11000; //24 --> 0
      #100;
      serial_in = 5'b11110; //30 --> 2

      //Test case 4: Mod 7 of a 6-bit number
      #100;
      serial_in = 6'b011111; //31 --> 3
      #100;
      serial_in = 6'b100010; //34 --> 6
      #100;
      serial_in = 6'b101010; //42 --> 0
      #100;
      serial_in = 6'b101110; //46 --> 4
      #100;
      serial_in = 6'b111010; //58 --> 2
      #100;
      serial_in = 6'b111111; //63 --> 0

  end
endmodule
```

Verilog: Constraints

The constraints file for exporting circuit to the FPGA board are the following:

```
# I changed "CLK100MHZ" to "clk"
# ------------------------------------------------------------------------------------------
# set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }];
# create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];
# ------------------------------------------------------------------------------------------

set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk }];
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}];

set_property -dict { PACKAGE_PIN H6  IOSTANDARD LVCMOS33 } [get_ports { serial_in }];
set_property -dict { PACKAGE_PIN P18  IOSTANDARD LVCMOS33 } [get_ports { reset }];

set_property -dict { PACKAGE_PIN M13  IOSTANDARD LVCMOS33 } [get_ports { configure[2] }];
set_property -dict { PACKAGE_PIN L16  IOSTANDARD LVCMOS33 } [get_ports { configure[1] }];
set_property -dict { PACKAGE_PIN J15  IOSTANDARD LVCMOS33 } [get_ports { configure[0] }];
```

```
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { out[0] }];
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { out[1] }];
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { out[2] }];
set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { led }];

set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { enable }];
```

Verilog Simulation Results

Waveform of the “Mod 7” circuit based on the simulation code:

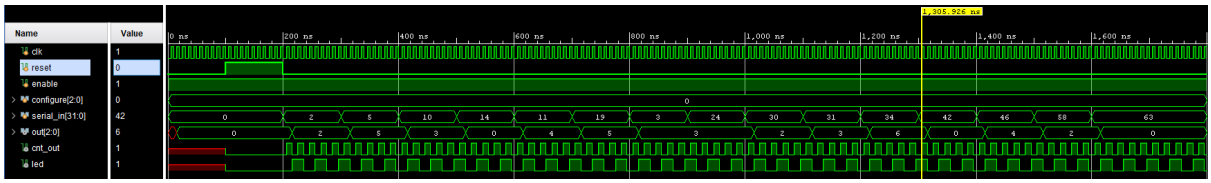

Figure 1: Complete Waveform

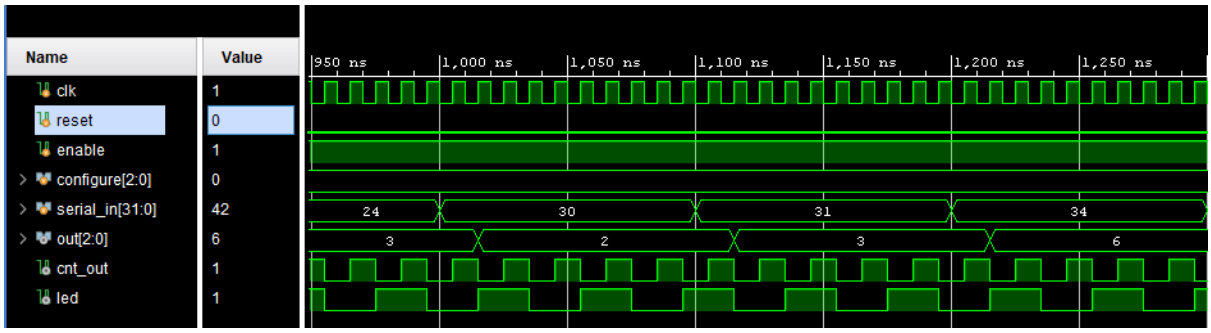

Figure 2: Zoomed in Waveform

Verilog: Synthesis

Synthesis results of “Mod 7” circuit.

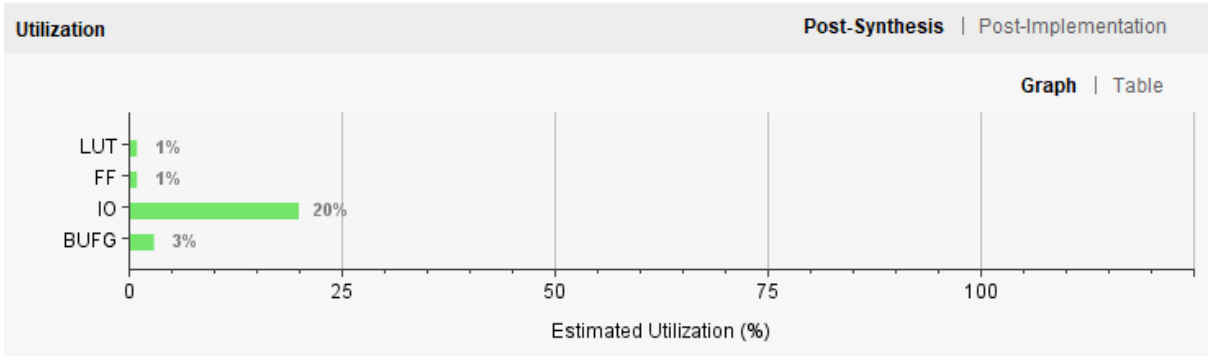

Figure 3: Synthesis Graph

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	120	63400	0.19
FF	42	126800	0.03
IO	42	210	20.00
BUFG	1	32	3.13

Figure 4: Synthesis Table

Verilog: Implementation

Implementation results of “Mod 7” circuit.

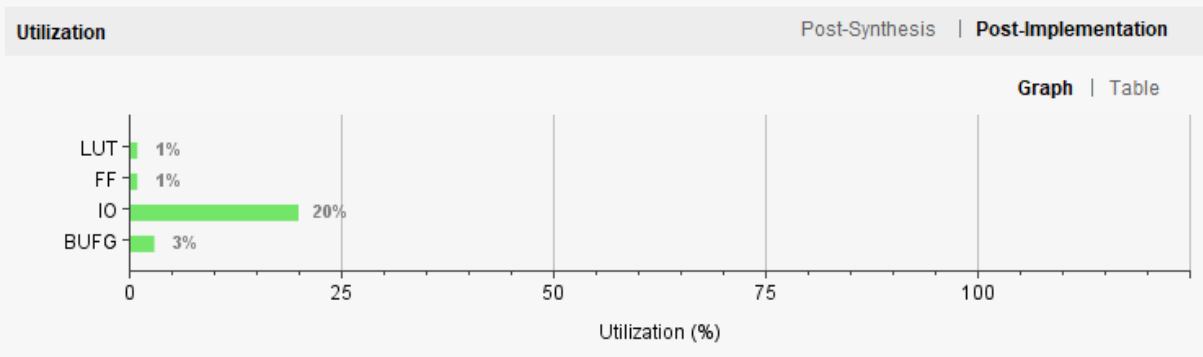

Figure 5: Implementation Graph

Utilization — Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	120	63400	0.19
FF	42	126800	0.03
IO	42	210	20.00
BUFG	1	32	3.13

Figure 6: Implementation Table

Verilog: Project Summary

Project summary of the “Mod 7” circuit.

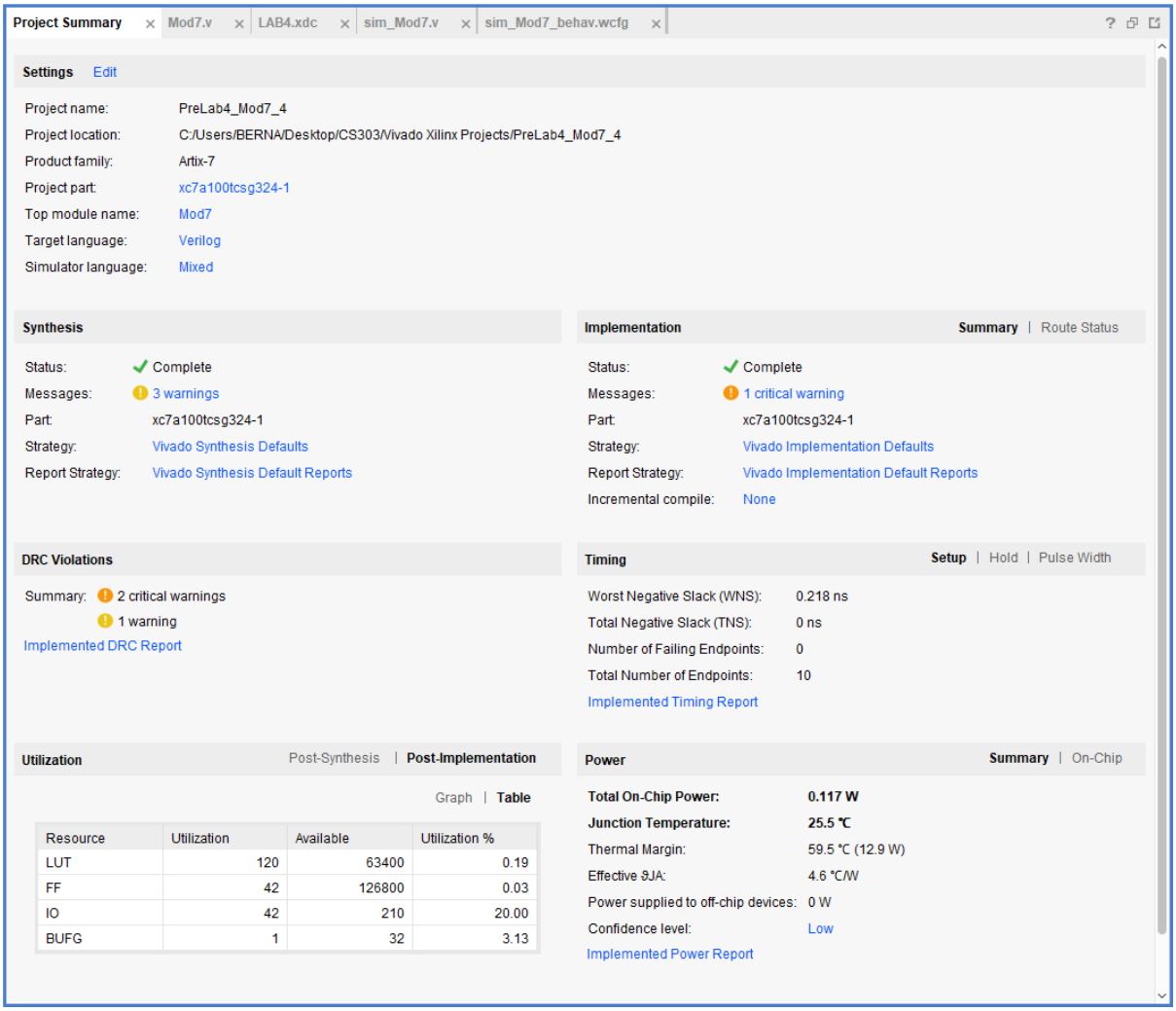

Figure 7: Project Summary