Diane Shan

862148900

EE120A Section 23

Jack Huang

Lab 6 – Timer Design – Laser Surgery System

**Overview**

In this lab, we first implemented a special purpose timer and figured out how to get a tick every 10 seconds using the 25MHz internal clock of the FPGA. Then we used that timer to implement a laser surgery system. Then we developed a verilog structural implementation of the system. We successfully implemented the laser surgery system with the timer. Doing the design, synthesis, and implementation of counters is reinforced in this lab along with using an internal clock signal to drive CLK inputs of flip-flops.

**New Concepts**

Special Purpose Timer – specialized type of clock used for measuring specific time intervals and developed for a special purpose

Laser Surgery System – uses laser light to remove diseased tissues or treat bleeding blood vessels

Most Significant Bit(MSB) – the bit in a multiple-bit binary number with the largest value, usually the bit farthest to the left.

**Analysis**

*Procedure*:

1. Implement a special purpose timer
2. Figure out what cnt\_rst should be in order to achieve a tick every 10 seconds using the 25 MHz internal clock of the FPGA
3. Use timer to implement schematics of laser surgery system
   1. Starts in OFF state
   2. If user presses button, then goes to START
   3. Then goes to ON until the timer goes high
4. Implement the structural verilog implementation of the timer and the laser surgery system

**Records**

*Schematics/Diagrams:*

Timer – Laser Surgery System FSM

Diagram

Description automatically generated

Timer – Laser Surgery System

Diagram

Description automatically generated

*Code:*

design.sv

// --------------------------------------

// Comb. Logic

// --------------------------------------

assign cnt\_ini = 32'h0000 ;

assign cnt\_rst = 32'h00001110111001101011001001111111 ; // 10 secs ( 25 MHZ internal clock )

// --------------------------------------

// Comb. Logic - FSM

// --------------------------------------

localparam OFF = 2'b00 ;

localparam START = 2'b01 ;

localparam ON = 2'b10 ;

always @( current\_state ) begin

case (current\_state)

OFF : begin

light = 1'b0 ;

reset = 1'b1;

if (b == 1'b1) begin

next\_state = START ;

end else begin

next\_state = OFF ;

end

end

START : begin

// your code for state transition

light = 1'b0 ;

reset = 1'b0 ;

next\_state = ON ;

end

ON: begin

// your code for state transition

light = 1'b1 ;

reset = 1'b0 ;

if (clk == 1'b1) begin

next\_state = OFF ;

end else begin

next\_state = ON ;

end

end

default: begin

light = 1'b0 ;

reset = 1'b0 ;

next\_state = OFF ;

end

endcase

end

// -------------------------------------- // Timer instantiation // -------------------------------------- timer\_st #( .NBITS(NBITS) ) timerst (

.timer(timer),

.clk(clk),

.reset(reset) ,

.cnt\_ini(cnt\_ini) ,

.cnt\_rst(cnt\_rst)

);

endmodule

**Discussion**

The system works according to the provided specifications. The timer works as it is supposed to and the laser surgery system as well. The timer and laser surgery system work together to form a cohesive system. There were no problems or technical issues encountered that resulted in system redesign or modifications. Everything went smoothly and worked well. Some possible ways to improve the system might be to test the system a little bit because we did not have a testbench. This could help see if there are any problems with the system. Another way to improve the system would be to be able to keep on going in the cycle if you want it to be continuous instead of going to the off state every time the timer goes high.

**Conclusion**

The purpose of this lab was to learn about the design of counters, synthesis, and implementation. We also learned about the usage of internal “clock” signals to drive CLK input of flip-flops. We also got familiar with designing special purpose timers. We were able to design a working special purpose timer that used a counter to change states in a laser surgery system. Now we know more about the uses of counters, special purpose timers, and internal clock signals and we know how to design a laser surgery system with a counter.

**Questions**

Assuming we will set cnt\_init = 0, what should cnt\_rst be in order to achieve a tick every 10 seconds using the 25MHz internal clock (CLK) of the FPGA?

* cnt\_rst should be 00001110111001101011001001111111 to achieve a tick every 10 seconds using the 25 MHz internal clock. This comes from 25,000,000 \* 10 – 1 which in hex is 0EE6B27F.