Georgia Institute of Technology

School of Computer Science

CS3220: Fall, 2014

Project 1: Timer

Version 1.0

Due: Wednesday, September 24th, 2014 at 03:00pm

Project description:

In this project, you will design an egg timer and synthesize it on your FPGA board. The timer will be counting down from a preset value to zero.

You will use several components on the board: (1) SW[7:0] (switches), (2) KEY[2:0] (keys), (3) LEDR[9:0] (red lights), and (4) HEX3, HEX2, HEX1, HEX0 (four 7-segment display). The following describes the detailed functionality of the timer:

KEY[0] is the reset button of the timer.
KEY[1] is used to set the time on the timer.
KEY[2] is for stop/start.

HEX3, HEX2 will show the minutes and HEX1, HEX0 will show the seconds as the timer counts down.

Flashing LEDR[9:0] show that count down has reached zero.

SW[3:0] will be setting the least significant BCD digit and SW[7:4] will be setting the most significant BCD digit of the timer seconds and minutes in two consecutive steps.

You need to design the timer such that it provides the following functionally.

Step 1: Reset
Whenever the *reset* button (KEY[0]) is pushed, all seven segment displays must display, "0000". This is an initial state of the timer and whenever you push the *reset* button, the timer should go back to this state. No LEDs will be on or flashing in this state.

Step 2: Set Seconds
HEX1 and HEX0 represent seconds of our timer. HEX1 shows tens place while HEX0 shows unit's place. You can use switches (SW) to set the seconds. SW[7:4] is used to set tens place while SW[3:0] is used to set unit's place. While you are setting the switches, the 7 segment displays, HEX1 and HEX0, should be changed constantly given your inputs. When you push the *set* button (KEY[1]), the seconds are set and the timer goes to the next step, in which we

will set the minutes.

Step 3: Set Minutes
This step operates similar to Step 2 except that it is for setting minutes that will be displaying at HEX3 and HEX2. In the same way you used the switches in Step 2, you should be able to change numbers displaying at HEX3 and HEX2 before you push the *set* button (KEY[1]). In this step, you will be using the same switches as Step 2.

Step4: Run Timer
Whenever you push the *start/stop* button (KEY[2]), the time that you set in the seven segment displays will be decreased until it reaches to 0000. You should always be able to stop the timer by pushing the *start/stop* button.

Step5: Turn Lights
Right after the timer counts down to "0000", *all red lights* (LEDR[9:0]) should be flashing until you push the *reset* button. The period of flashing the red lights should be 1 second. That is, the LEDR[9:0] will be one for 0.5 second and will be off for 0.5 second. The flashing will continue until you push the rest button.

*** You can use the code we provide below for BCD to 7-segment conversion. The clock we use in this project is CLOCK_50.

Verilog code for BCD to 7-segment conversion:

```
module dec2_7seg(
	input [3:0] num,
	output [6:0] display
);
	assign display =
		num == 0 ? ~7'b0111111 :
		num == 1 ? ~7'b0000110 :
		num == 2 ? ~7'b1011011 :
		num == 3 ? ~7'b1001111 :
		num == 4 ? ~7'b1100110 :
		num == 5 ? ~7'b1101101 :
		num == 6 ? ~7'b1111101 :
		num == 7 ? ~7'b0000111 :
		num == 8 ? ~7'b1111111 :
		num == 9 ? ~7'b1100111 :
		7'bxxxxxxx;   // Output is a don't care if illegal input
endmodule
```

Rules:

This project is a team project so you should work together with your teammate.

What to hand in via T-Square:

There are two things you have to submit.

(1) Block diagrams of the design (PDF file)

You should submit block diagrams describing the entire design. The easiest way to do this would be to take a picture of a hand-drawn diagram. The file should be transformed into the pdf format before submission. You should also include a separate diagram in the same pdf file showing the state machine of the controller for your egg timer.

You should follow the following specific naming format:

Timer[Student's Full Name].pdf (e.g. TimerJongsePark.pdf).

(2) Quartus Project that includes Verilog files (ZIP file)

You should also submit zip file containing the entire Quartus project directory that includes your Verilog codes. Your design should include a separate module for the controller of the egg timer. You should name this file TimerController.v. The project directory name and the zip file name should comply with the following naming formats:

Directory: Timer[Student's Full Name] (e.g. TimerJongsePark)
Zip file: Timer[Student's Full Name].zip (e.g. TimerJongsePark.zip)

Finally, you should submit your files through t-square.