Georgia Institute of Technology
School of Computer Science
CS 3220: Fall 2017
Project 1: Timer [100 Points]

Due: Friday, 20^{th} October, 2017

Project description. In this project you will design and synthesize an egg timer for your FPGA development board. The timer will count down from a user-defined value to zero. You will use the following components of the board:

1. SW[7:0] (switches).
2. KEY[2:0] (keys).
3. LEDR[9:0] (red lights).
4. HEX3, HEX2, HEX1, HEX0 (four 7-segment displays).
5. CLOCK_50, the clock you will use for your design (50MHz).

These are the user interface of your design:

1. KEY[0] is the *reset* button of the timer.
2. KEY[1] is used to *set* the initial value of the timer.
3. KEY[2] is used for *stop/start*.
4. HEX3 and HEX2 displays the minutes, whereas HEX1 and HEX0 will display the seconds as the timer counts down.
5. Flashing LEDR[9:0] indicates the countdown has reached zero.
6. SW[3:0] sets the least significant BCD digit, and SW[7:4] will set the most significant BCD digit of the timer. Refer to step 2.

You need to design the timer such that it provides the following functionalities:

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 should be on or flashing in this state.

Step 2: Set Seconds

HEX1 displays the tens place while HEX0 displays the units place of the seconds. You should use switches (SW) to set the seconds. Refer to user interface bullet 6.

While you are setting the switches, the 7 segment displays — HEX1 and HEX0 — should change reactively, depending on the current input. When you push the *set* button (KEY[1]), the seconds are set and the timer goes to the next step.

Step 3: Set Minutes

HEX3 displays the tens place while HEX2 displays the units place of the minutes. You should use switches (SW) to set the minutes. Refer to user interface bullet 6.

While you are setting the switches, the 7 segment displays — HEX3 and HEX2 — should change reactively, depending on the current input. When you push the *set* button (KEY[1]), the initial timer value is set.

Step 4: Run Timer

When the *start/stop* button (KEY[2]) is pressed, the time you set in the seven segment displays should start decreasing until it reaches to 0000. You should always be able to stop the timer by pushing the *start/stop* button.

Step 5: 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 flashing period of the red lights should be exactly **one** second. That is, the LEDR[9:0] LEDs will be on for **0.5** seconds and then off for **0.5** seconds. The flashing should continue until you push the *reset* button.

Notes

1. This project is a **team** project so you mush work together with your teammate.
2. Your design must include a separate module for the controller of the egg timer. You must name this file **TimerController.v**
3. You must submit all the files and reports through T-Square.
4. Please use Piazza for any questions that you might have.

What to include in your submission

There are two deliverables:

1. Block diagrams of the design. You must submit two block diagrams in one PDF file: (1) describing the entire design and (2) showing the details of the state machine of the controller of the egg timer. This file must be in PDF format. Hand-drawn diagrams are *not* accepted. You must adhere to this naming format: **Timer-[Student's Full Name].pdf (*e.g.* Timer-AmirYazdanbakhsh.pdf)**.

2. Quartus project that includes Verilog files. You must submit a zip file containing the *entire* Quartus project directory containing your Verilog code. The project directory name and the zip file name should adhere with the following format:

   - Directory: **Timer-[Student's Full Name] (*e.g.* Timer-AmirYazdanbakhsh)**
   - Zip File: **Timer-[Student's Full Name].zip (*e.g.* Timer-AmirYazdanbakhsh)**

Note 1. You get zero points if you do not follow the naming format.

Note 2. Your Quartus project has to be synthesized on FPGA without any other modifications.

Note 3. Please use Quartus II Web Edition 13.0sp1 from Altera. Here's a quick installation guide: `https://www.cc.gatech.edu/~hadi/teaching/cs3220/doc/quick-start/quick-start-de0.pdf`.

Note 4. The Verilog code for BCD to 7-segment conversion is uploaded to T-Sqaure (**dec2_7seg.v**). Feel free to use the code in your design.

Good Luck!