EN2911X: Reconfigurable Computing
Brown University
Division of Engineering
EN2911X Reconfigurable Computing
Prof. Sherief Reda
HW/LAB 03 (150 points). Due Date: Friday, Oct 3\textsuperscript{rd} and Oct 10\textsuperscript{th}.

- Tutorial 2 is similar to Tutorial 1 except it uses Verilog instead of schematics. Please follow it to make sure you know how to use Verilog with Quartus II.
- For all problems, report (1) the details of the logic and routing resources utilized by your circuit; (2) a screen capture of the RTL circuit view (tools $\rightarrow$ netlist viewer $\rightarrow$ RTL viewer); (3) a screen capture of the post-mapped circuit view (tools $\rightarrow$ netlist viewer $\rightarrow$ technology map); (5) a screen capture for the floorplan after fitting. Make sure to email your project’s archived file (.qar) to Marco Donato (marco_donato@brown.edu).

1. [25 points – due date Oct 3\textsuperscript{rd} – \textbf{team work not allowed}] Design a circuit that uses three push buttons to display three possible messages on the 7-segment display. If one button is pushed, the 7-segment displays should display the message “HI”. If the second button is pushed, the 7-segment displays should display “LO”, and if the third button is pushed, the 7-segment displays should display “=”. The message should stay displayed as long as the button is pushed.

2. [35 points – due date Oct 3\textsuperscript{rd} – \textbf{team work not allowed}] Design a circuit that enables the following two-player game. Essentially the first player will make up an arbitrary number (say between 0 and 31) but not reveal it to the second player, and the second player will try to guess the number correctly. The second player should input a number (in binary) using the first 5 switches on the DE2 board and the number should be displayed accordingly on the 7-segment display. The first player then takes turn by pressing one of three buttons that display either “HI”, “LO”, or “=” on the 7-segment displays to indicate whether the guessed number is higher, lower or equal to the made-up number. If the second player guessed the number correctly, then the game ends; otherwise, the game continues. Of course the second player’s (essentially the TA) goal is to finish with the fewest number of guesses.

3. [40 points – due date Oct 3\textsuperscript{rd} – \textbf{team work not allowed}] Design a light blinker circuit that uses Morse code to transmit messages visually using one red LED light. A dash in Morse code should correspond to a blink that lasts for a period of 3 seconds, while a dot should correspond to a blink that lasts one second. Pause 0.5 seconds between letters and 3 seconds between words. Your code should be able to any set of characters on the LED. For lab demonstration, please use your design to output the statement “I LOVE YOU” on the LED using Morse code. (Hint: You will have to build a timer that uses the available 50 Mhz clock signal to create the required period.)

4. [50 points – due date Oct 10\textsuperscript{th} – \textbf{team work allowed (2/team)}] You are asked to create a reconfigurable circuit for a game! The game in suggestion here is inspired by the
BrainTuner application for iPhone. In this game the DE2 board will prompt you with simple arithmetic calculations and their results, and you are required to indicate whether the result is correct or not. You will use the pushbuttons to indicate whether the result of an operation is right or wrong. If you are correct, the DE2 board will light a green light. If you are wrong, the board will light a red light and penalize you 5 seconds. At all time, the board will also display a 2-digit timer in seconds as the questions are being prompted. Your final score is your time.

The game should be challenging by posing questions that trick you in answering them incorrectly especially that you are under time stress to finish quickly. For example, notice the “trickery” of the following random questions, where an incorrect answer is not “far” from the correct answer:

- $4 \times 8 = 36$
- $3 - 1 = 6$
- $2 - 5 = 3$
- $5 \times 8 = 40$
- $10 - 6 = -4$
- $7 \times 5 = 32$
- $2 \times 2 = 2$

To manage the 7-segment displays, the figure shows the recommended layout. You will only use signal digit operands and three operations only (+, -, *). The result can occupy with the negative sign at most two 7-SEG. At the end of the game, only the final time will be displayed. At any time (whether during or at the end), you can press a reset button to start playing all over again with different sets of random or pseudo-random questions. 5 points will be allocated to games that work smoothly in an entertaining manner.

**Helpful Tips:**

- Make sure to the import the pin assignment from the DE2 pin assignment file (csv) that is available at the class web site. The names of various input/output signals are given in the figure below.
- To access the floorplan, choose Tools → Chip Planner
- To access the resource utilization, choose Processing → Compilation Report
- The Quartus tool has an extensive library of pre-designed megafunctions that implement many functions (e.g., decoders, counters, registers, comparators, ROMs, etc). You should use these megafunctions whenever possible. You can access the library by clicking the “symbol tool” in the tool bar.