ECEN2350 Digital Logic – Project 2

Fall, 2020

Due Nov 6th, 2020, by end of the day

110 points (2 checkpoint submissions plus final submission)

Everyone is responsible for turning in their own project. You may work with a partner to complete the project.

This project will require you to create, simulate, verify your design using iVerilog, Quartus, GTKWave, and your DE10-Lite board, and submit a working project and a project report. The goal of Project 2 is to extend your design knowledge to synchronous design techniques, and to create testbenches that simulate this synchronous design.

Upon completion of this project, you will be familiar with synchronous design coding techniques, creating binary and modulo-10 counters, and moving between binary and binary coded decimal (BCD) representations of data. To review BCD is a standard binary format where each digit is represented using 4 bits, but the maximum allowable value of a digit is 9 (or 4’b1001).

There will be two checkpoint submissions required for this project, one due on Friday October 16th, and the second due on Friday, October 23rd. Each checkpoint submission will be worth 15 points. Your final submission will be worth 80 points. You will find the information for each Checkpoint submission in Canvas.

Project Description:

Project2 consists of one design that displays the day of the year and the corresponding month and date.

Detailed Requirements for Project2 (Read carefully, get clarifications early)

1. Create a counter will count between **1** and **99** inclusive (base 10) representing the first 99 days of the year, with the output displayed on HEX5 and HEX4 as decimal digits. The most significant digit must be blanked instead of displaying zero. When the counter value reaches 99 decimal, the counter should roll over to 1 continue counting.

2. The update frequency for the display will be 2 Hz. Use the 10Mhz clock provided by the DE10-Lite board – this clock is named ADC_CLK_10 – and divide down to a 2 Hz display clock. Connect the display clock to LEDR[0] so you can see the

LED blink at the display clock rate.

3. KEY[0] will be used as the RESET signal for your design, pressing KEY[0] will force the day of the year counter back to 1, and the MMDD display back to January 1 (01 01). Every counter you create must include a RESET signal.

4. Using the value of the day of the year, compute and display the corresponding date in MMDD format, where MMDD are decimal digits. HEX2 will contain the month (HEX3 will be blank), and HEX1 and HEX0 will display the day (DD). The most significant digit of DD must be blanked instead of displaying zero.

   For example, if the day of the year is displaying as 15, this corresponds to January 15th (MM = (blank)1, DD = 15). If the day of the year is displaying as 90, this corresponds to March 31st (MM = (blank)3, DD = 31).

5. Increase the display update frequency to 5 Hz whenever KEY[1] is pressed and held.

6. Display the MMDD values based on a leap year when SW[9] is ON. A leap year has 29 days in February.

Testbench Requirements

1. You must create a testbench that demonstrates aspects of the design. At a minimum:
   a. Demonstrate the day of the year counter counting through the entire 1 to 99 range, including roll over.
   b. Show the MMDD outputs as the day of the year counter increments.
   c. Show the effect of reset.
   d. Show the change in display clock frequency based on KEY[1].
   e. Show behavior when leap year is selected.

2. Your should use GTKWave for your simulations, as graphical outputs are generally more informative that text based outputs.

Background Information

1. January contains 31 days.
2. February contains 28 days, except during a leap year, when February contains 29 days.
3. March contains 31 days.

4. Binary coded decimal (BCD) means that the only allowable digits are 0 – 9.

HINTS:

1. You can reuse your seven segment decoder from Project1. You may want to modify the decoder to add blanking capability, or leave your decoder unchanged and use additional logic for display blanking.

2. Start with the day of the year counter. This counter must be coded as a BCD counter. You will find that creating two separate BCD counters (one counter for each digit) will be simpler (this is a recommendation, not a requirement).

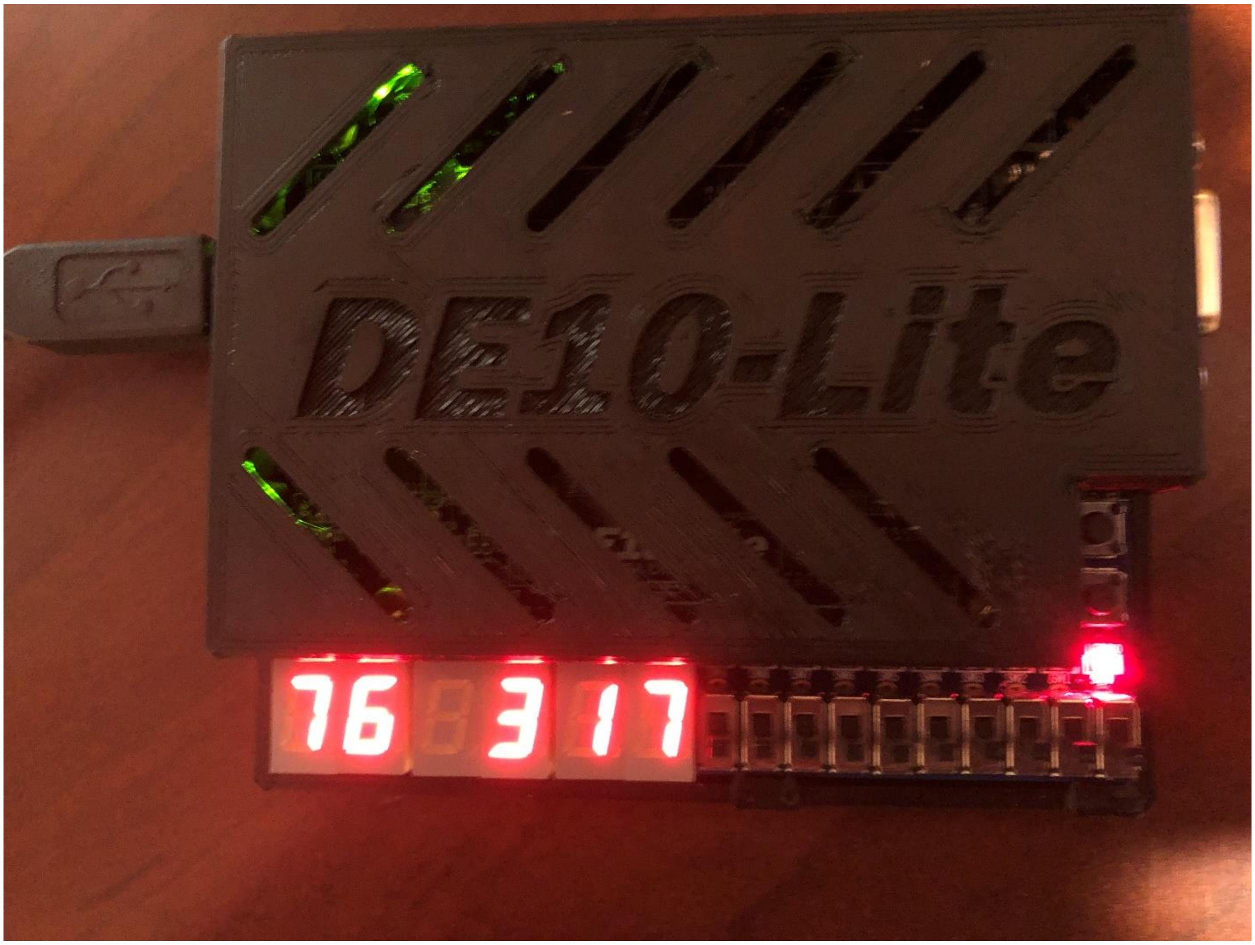
3. To convert from 2 4 bit BCD digits to binary, you can use this (or similar) Verilog code:

```
wire [6:0] binary_value;
assign binary_value = (BCD_value[7:4] * 10) + ({3'b0, BCD_value[3:0]});
```

   If you want an 8 bit value instead of a 7 bit value, make binary_value 8 bits wide, and append 4 0s instead of 3 to BCD_value.

4. Simulating a circuit with a high frequency clock requires many millions of simulation cycles. I will show you a method you can use to simplify you simulations before the second project day.

5. We will discuss the creation of clock dividers in class before the second project day.



76th day of the year, date is March 17th.