ECEN 2350 Lab 2 Report
Adam Chehadi - 11/6/19

Problem Statement

The purpose of Lab2 is to display the day of the year, and its corresponding month and date. A BCD (Binary Coded Decimal) counter displays the day of the year between 1 and 99 inclusive. In addition to this, the day of the year is used to display its corresponding BCD representing the date in MMDD format. The counter updates at 1Hz, and in addition to updating the displays at the clock frequency, LEDR[1] is connected to the clock. The value of KEY[0] is latched, its value is displayed on LEDR[0], and used to activate a low RESET signal for the design.

The purpose of the extra credit portion of Lab2 is to create an additional clock updating at 5Hz, displayed on LEDR[1], and controlled by KEY[1]. The default clock frequency is 1Hz. In addition to this, SW[9] is used to modify the MMDD BCD behavior to count based on a leap year.

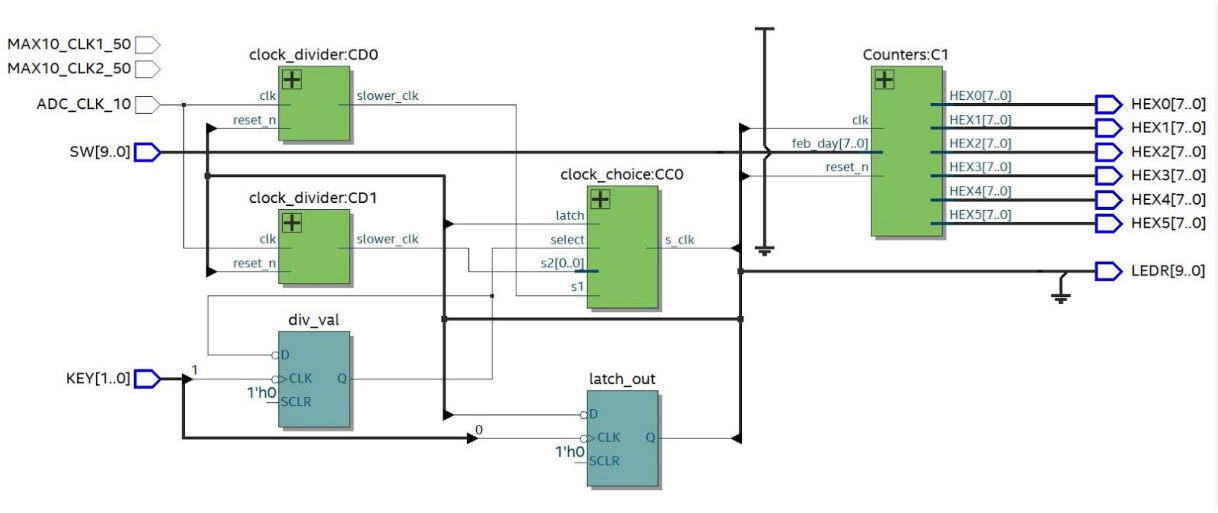
Theory of Operation

The updating clock frequency is created by dividing the ADC_CLK_10 clock provided by the DE10-Lite board. A `clock_divider.v` module is created and passed in the desired frequency parameter, the system ADC_CLK_10, the reset latch from KEY[0], and a slow clock wire to output the slower clock frequency. The `clock_divider.v` module takes those values and flips the value of the slow clock every time it's one less than the desired frequency parameter. `clock_divider.v` is instantiated twice in `Lab2.v` for 1Hz and 5Hz. A `clock_choice.v` module is created to switch between the 1Hz and 5Hz clock frequency based on the latched value of KEY[1]. `clock_choice.v` utilized a multiplexer case statement to switch the clock frequencies.

A `Counters.v` module is utilized to form and display a 1-99 BCD on HEX5-4. KEY[0] is the reset key, and its latched value is passed into a `Counters.v` instantiation along with the output of the clock divider, and HEX displays. In addition to this, a feb_day register is passed into `Counters.v`. When SW[9] is on, the BCD operates in a leap year, and otherwise the BCD operates as if it's a normal year. When SW[9] is on, feb_day <= 8'd29, and when SW[9] is off, feb_day <= 8'd28. `Counters.v` contains three 8-bit registers that hold the value of the BCD tens place, BCD ones place, and the total_count 1-99 value of the BCD. If the reset switch is toggled at any time, the tens place is set to blank, and the ones place and total_count are set to 8'd1. The ones place cycles from 1-9, the tens place increments when the ones place == 8'd9, and the total_count increments every time the tens/ones place changes. When the tens place and ones place both == 9, all three registers are reset. The tens and ones BCD place are passed into SevenSeg and displayed on HEX5 and HEX4 respectively. The total_count and feb_day are passed into a `month_day.v` module.

A `month_day.v` module is utilized to interpret the value of total_count and display the proper month and date value. The total number of days for January: 31, February: 31 + feb_day, and March: 31 + feb_day + 31, and April: else condition. The corresponding month values are 1, 2, 3, and 4 respectively. `SevenSeg.v` is instantiated with a blank value passed to HEX3, and the month value passed to HEX2. An 8-bit day_full register is created in month_day and hold the date value. In January: day_full <= total_count, February: day_full <= total_count - 8'd31, March: day_full <= total_count - (8'd31 + feb_day), April: day_full <= total_count - (8'd31 + feb_day + 8'd31). day_full is passed into `day_display.v`. A `day_display.v` module is utilized to interpret the day_full value and properly display its tens and ones place value.

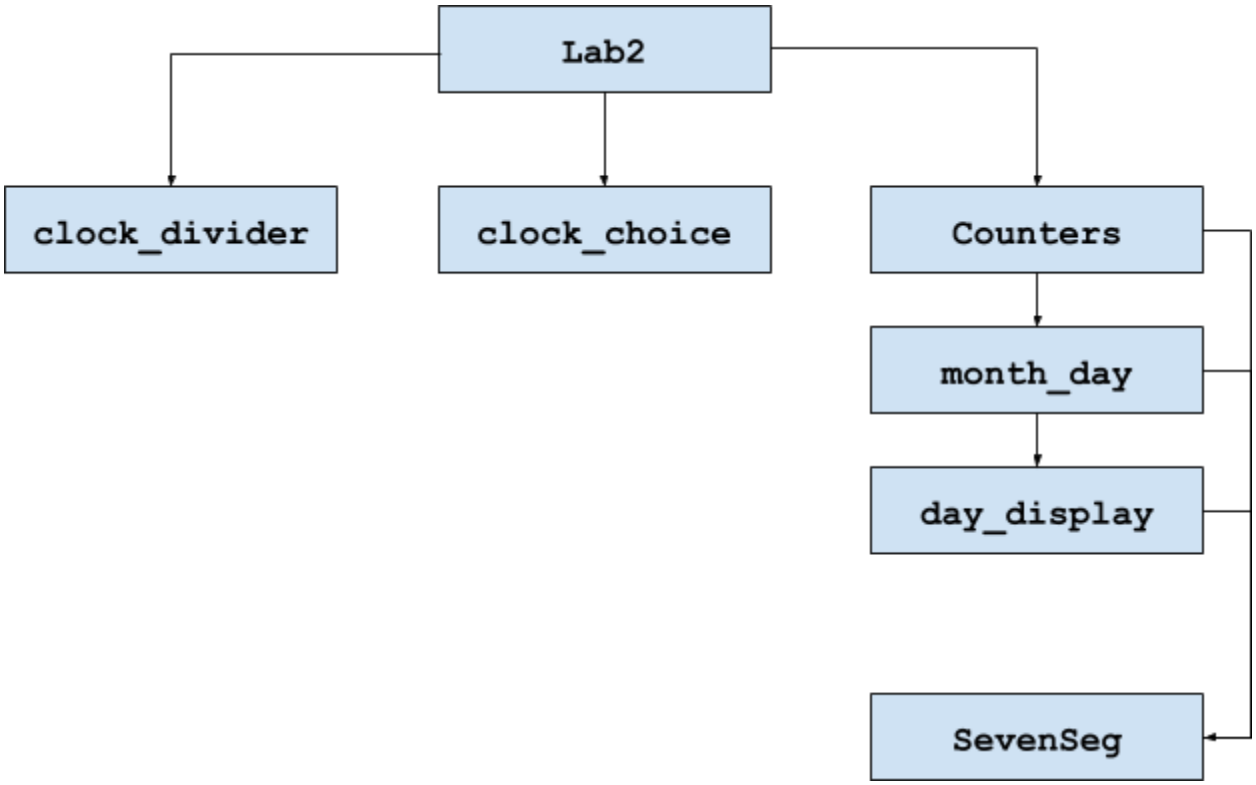
Block Diagram of Your Design



Block diagram of the modules, along with inputs and outputs produced by Quartus (Tools > Netlist Viewers > RTL Viewer).

Hierarchy of Source Files

`Lab2.v` is at the top of the hierarchy. `Lab2.v` instantiates `clock_divider.v`, `clock_choice.v`, and `Counters.v`. `Counters.v` instantiates `month_day.v`, and `month_day.v` instantiates `day_display.v`. `month_day.v` and `day_display.v` instantiate `SevenSeg.v`.



Flow chart of module hierarchy and instantiation, from the top file down.

Description of Testbench Operation and Output

The testbenches are created to test and demonstrate the functionality of the design. A boilerplate testbench was created to use across multiple testbenches. This base testbench is comprised of a clock initialized to zero, and a reset initialized to zero and changes to one 10ps later. The value of the clock is flipped every 5 ps.

The testbench for the clock divider (`TB_clock_divider.v`) passes a small parameter into the clock divider in order to display a slower output in GTKWave.

The testbench for the BCD (`TB_Counters.v`) tests the values of total_count, counter_tens_p, counter_ones_p, and the day of the month HEX displays. A testbench similar to `TB_clock_divider.v` is created that instantiates the `Counters.v` module. The values of total_count, counter_tens_p, counter_ones_p, and the reset can be viewed in GTKWave. The counters initially contain random values, and are reset to their appropriate values. The HEX displays update on the clock cycles in GTKWave, but their actual values in HEX are not easily discernible as the physical number they are actually displaying. The HEX displays are passed an

8-bit binary value, and each bit drives one of its segments. Since the HEX displays are updating at their proper frequency, it is clear that the month and date are updating at the correct frequency as well.

The behavior of the different selected clock frequencies (`TB_clock_choice.v`), and leap year (`TB_leap_year.v`) was also tested. For these testbenches, the behavior of the counters and slower clock frequency were already tested in the prior testbenches, so only the values that changed were tested. In order to test the clock choice, KEY[1] is toggled on the positive edge of the clock, which correctly flips the divider value (div_val). The value of the leap year is tested by toggling the value of SW[9] on the positive edge of the clock. The number of days in February (feb_day) is seen in GTKWave to switch between 8'd28 when SW[9] == 0, and 8'd29 when SW[9] == 1.

Summary of Project Success, What Works and What Doesn't

The project fulfills all of the project requirements, and also implements the extra credit functionality. The design is able to properly utilize a BCD to display the days of the year on HEX4-5, and its corresponding month on HEX3-2 and date on HEX1-0. SW[9] can be toggled to modify the behavior to count based on a leap year, and the MSB is always blanked instead of displaying zero. The default update clock frequency is 1Hz, is connected to LEDR[1], and a 5Hz frequency can be toggled with KEY[1]. KEY[0] is toggled, latched, connected to LEDR[0], and is used as an active low RESET signal for the design.