CPEN 230L: Introduction to Digital Logic Laboratory

Lab # 10: Switch Debouncing and Counters

Purpose

- Learn more about switch debouncing, synchronous counters, modulo-N counters, cascaded counters, and the DE2-115 board 50 MHz clock.
- Learn more about Verilog: procedural assignment requirements (reg), "parameter" and parameter overriding, concatenate and replicate, and the timescale compiler directive.

Part 1: Switch Debouncing

Pre-Lab

- **Background:** Most mechanical switches bounce when changing state, resulting in many rapid transitions between 0 and 1 before finally settling. It is necessary to clean up these noisy transitions prior to using the switch for an edge transition such as a clock input. **Switch debounce circuits** accomplish this. An S-R Latch can be used to debounce a single-pole double-throw switch as shown in Figure 7.51(b) on page 484 of your textbook. The DE2-115 board provides single-pole single-throw switches (Figure 7.51(a)). We will use a counter-based circuit to require about 10 ms of switch stability before letting an output make a clean single-edge state transition.
- Read provided file **debouncer.v**. Understand why output clean_o is type reg. Understand its use of a Verilog **parameter** (pp.710-713 textbook) to allow overriding the default delay of 10.5 ms. Understand Verilog **concatenate and replicate** -- "{cnt_bits{1'b1}}" means "concatenate cnt_bits copies of the single bit 1". In this code, cnt_bits defaults to 19, so the result is 19'h7FFFF. If cnt_bits were overridden in higher level code to be 20, the result would be the constant 20'hFFFFF.
- Read provided file **debouncer_tb.v**. Understand its use of the Verilog compiler directive **timescale** to make 1 simulation tick be 1 ms. Understand how it simulates the debounce time of 2^19−1 cycles of a 50 MHz clock (10.5 ms) in 2^3−1 = 7 simulation clock cycles. Understand how it overrides debouncer's default cnt_bits value of 19 with a value of 3.
    - Why don't we want to leave cnt_bits at 19 during simulation?
- Look at file **debouncer_wave.jpg**, the results of debouncer_tb.v.
    - What is being demonstrated in the time between the two markers?
    - What is being demonstrated from the B marker to 60 ms?
- Create a file cmdfile.txt (see Lab.7) and use it to compile the verilog files T_FlipFlop.v, debouncer.v and debouncer_top.v with Icarus:

```
iverilog –c cmdfile.txt
```

During Lab

- Use Quartus and the DE2-115 board to build, debug and verify your debouncer circuit.
- When convinced your circuit is working properly, demonstrate it to your instructor

Part 2: Mod-10 counter with 1-second update

Pre-Lab

- Read provided files **cascadeCounter.v** and **cascadeCounter_tb.v**
- Look carefully at the table and waveforms generated by the simulation (**cascadeCounter_table.jpg** and **cascadeCounter_wave.jpg**): make sure to fully understand how cascade counters work and how can be reused.
- Create directory **Lab10\timerModN** with other directories and files under it, to **simulate and synthesize** a synchronous mod-10 cascade counter with 1-second update rate and asynchronous Reset. When synthesized on the DE2-115 board, 7-segment display HEX0 will count 0 to 9, 1 second per value, and then repeat. Pressing KEY0 will reset the count to 0. Releasing KEY0 will allow the count to start counting up again from 0.
    - Module **timerModN** will include three Verilog parameters to allow higher-level code to set the number of bits in stage 0, the terminal count of stage 0, and the terminal count of stage 1. Default values in this module need to provide the functionality described so far (1-second update rate, mod-10).

- Module **timerModN_top** will connect module timerModN to the DE2-115 board KEY[0], KEY[1] and HEX0 input/output devices.
- Module **timerModN_tb** tests timerModN (not timerModN_top). It overrides the timerMod10 parameter that sets the stage 0 terminal count, giving it a value of 4. GTKWave should indicate on its waveform display that 5 cycles of the timerModN clock correspond to 1 second.
    - By setting the stage 0 terminal count to 4, it will count mod-5 during simulation, instead of a much larger default value that would be impractical to use during simulation. Making 5 clock cycles correspond to 1 second should make the simulation waveform display appear to increment the stage 1 mod-10 counter every 1 second.

During Lab

- When your counter is working as desired on the DE2-115 board, demonstrate it to your instructor.
- Modify module timerModN_top to make the counter count mod-7 with a 0.5 second update.
- When the modified counter is working as desired on the DE2-115 board, demonstrate it to your instructor.

Pre-Lab Deliverables

The penalty for not handing in the prelab deliverables is 50% of the lab grade.

Part 1

- cmdfile.txt

Part 2

- timerModN.v, timerModN_tb.v, timerModN_top.v
- a snapshot of the simulation waveforms