Name: **Section:**

Homework Assignment: submit via gradescope

Neatly draw the Circuit Diagram for your cascade counter.

Upload the following to your 383 Github Repo:

- Your VHDL code for your cascade counters (do your comments include the truth table?)
- Your VHDL code for your testbench and simulation waveform plot. The testbench exercising the cascade pair, must:
  - Hold the least significant counter at 4 for one clock cycle (using crtl='0').
  - Roll over the least significant counter once.
  - Show clk, reset, Q1, Q0, (least significant) roll signal, and the crtl input to the most significant counter.
  - Remove all junk signals.
  - Fit waveform on one page.

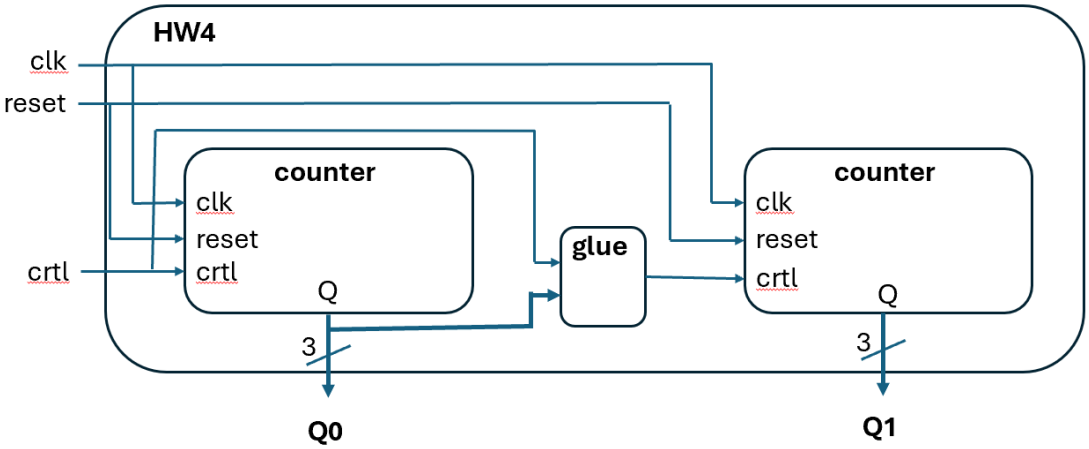
Details.

1. Instantiate a pair of cascaded counters, similar to those in Figure 13.2 of the text. The pair of counters operate in a coordinated fashion, with one counter representing a least significant value, and the other a most significant value. When the least significant counter is going to roll over, the most significant counter will count up by one. At no other time will the most significant counter increment.

You should have one counter that operates according to the following truth table.

clk	reset	crtl	Q+
0,1,falling	x	x	Q
rising	0	x	0
rising	1	0	Q
rising	1	1	Q+1 mod 5

You should instantiate this counter twice and add some glue logic between the two devices so that the most significant counter counts up only when the less significant counter is going to roll over. Include the truth table for the glue logic as an explicit comment in your VHDL code. The high-level architecture for this assignment is given in the block diagram below. Please note that the counter digits do NOT generate a "roll" signal, rather create “glue” logic as a combinational logic statement along side the two counter instances. In your circuit diagram, replace the glue logic with the actual logic used (like a mux and an OR gate?)



The top level entity description should look like the following. When crtl = '1', the counters are enabled to count up as a cascade pair, and when crtl = '0', the counter should hold their value.

```
entity hw4 is
    port(   clk, reset: in std_logic;
            crtl: in std_logic;
            Q1, Q0: out unsigned(2 downto 0));
end hw4;
```

Testbench Tip

In order to test your cascade counters, you will need to apply a complex test sequence to the control signal. The following VHDL code in your testbench will help achieve this. This is CSA version of the process structure given in section 2.2.4 of the textbook.

```
crtl <= '1', '0' after 15us, '1' after 16us, '0' after 17us, '1' after 18us;
```

Check out the testbench linked at the top of lecture 4 for more details.

Documentation Statement: For all assignments in this course, you may work with any faculty members or students **currently** enrolled in ECE383 unless otherwise indicated. We expect all graded work, to include software programs, wired circuits, lab notebooks, and written reports, to be your own work. If they aren't, you've copied and will receive **no academic credit** even if the copying is documented. Further, copying without attribution is dishonorable and will be dealt with as a suspected honor code violation. As in all courses, cadets must document any assistance received in the execution of graded work. If you receive no assistance on an assignment, the use of the **Documentation: None** statement is mandatory. If no documentation statement exists, the assignment will be returned for correction and the work will be considered at least one day late.