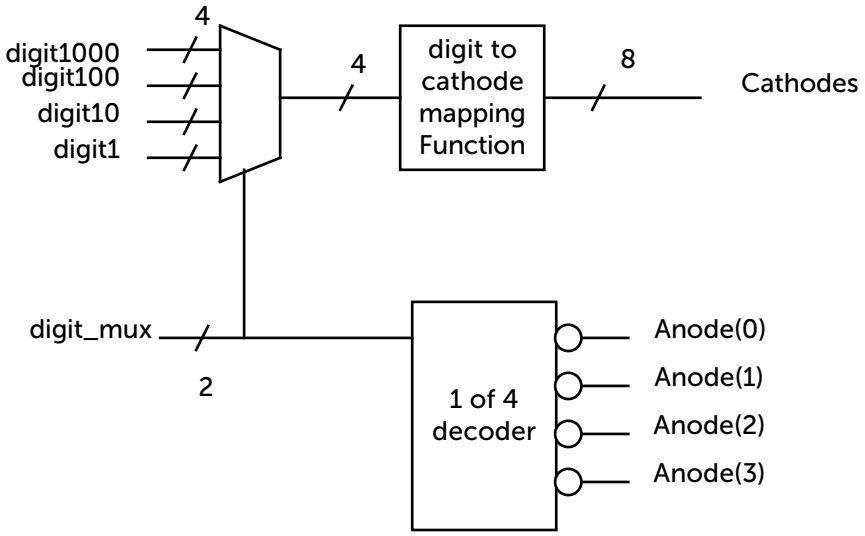
EE 331 Digital Systems with HDL
Oregon Tech, Wilsonville, Winter 2014

Lab Assignment 5, due Week 7, February 18

In this lab you will change the Lab Assignment 4 program so that count value appears on the 7-segment display. The display will also be decimal values.

The block diagram for a generic display driver is:



The two-bit signal, digit_mux, determines which digit is being displayed by selecting the correct digit input signal (digit1000, digit100, digit10, or digit1) with the multiplexer and energizing the correct anode of the 7-segment display array. To generate digit_mux, add a 20 bit counter that runs continuously and use the most significant two bits as signal digit_mux. This value will change roughly every 256,000 clock ticks since there are 18 lower bits in the counter (2^{18}), which is about 2.5 milliseconds. The whole display refreshes about 100 times a second so the flicker will not be noticeable.

The digit to cathode mapping function can be done with a WHEN statement, a CASE statement (in a PROCESS), or by creating a ROM (see the VHDL.pdf file for this approach). The table below shows the values for the segments for each digit value.

	DP	CG	CF	CE	CD	CC	CB	CA
0	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	1
2	1	0	1	0	0	1	0	0

	DP	CG	CF	CE	CD	CC	CB	CA
3	1	0	1	1	0	0	0	0
4	1	0	0	1	1	0	0	1
5	1	0	0	1	0	0	1	0
6	1	0	0	0	0	0	1	0
7	1	1	1	1	1	0	0	0
8	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0
A	1	0	0	0	1	0	0	0
B	1	0	0	0	0	0	1	1
C	1	1	0	0	0	1	1	0
D	1	0	1	0	0	0	0	1
E	1	0	0	0	0	1	1	0
F	1	0	0	0	1	1	1	0

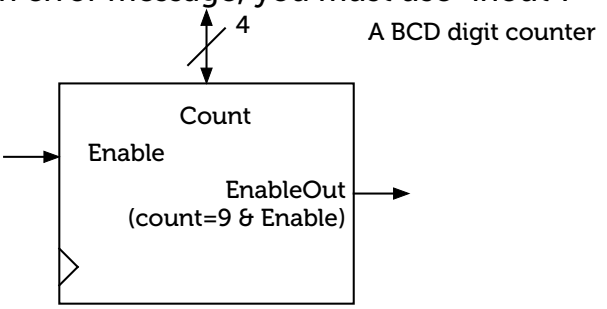
As a first part of this assignment I suggest that you add the 7-segment display and test it out by connecting the 8 bit binary counter to both the row of LEDs and the Digit10 and Digit1 inputs of the 7-segment display circuit. Connect the Digit1000 and Digit100 inputs to a constant 0 value. You should see the 7-segment display count up in hexadecimal.

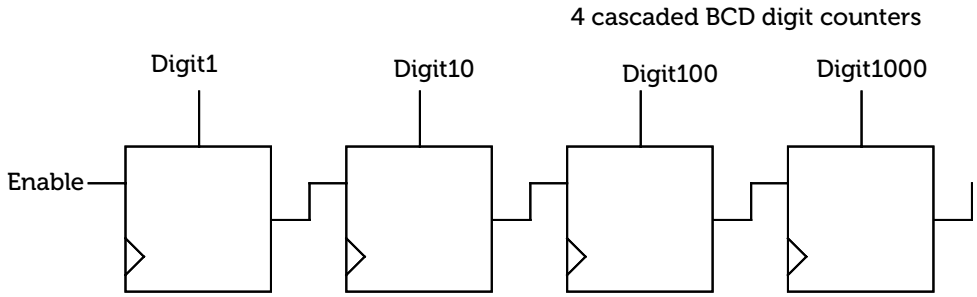
Having a binary counter is quite “user unfriendly” because most people prefer digital value readouts. Replace the 8 bit binary counter with a 4 digit BCD (binary-coded-decimal) counter, and connect to all four digits of the 7-segment display to show its value.

Because of the complexity of the 4-digit BCD counter, put it in a separate module (VHDL component) that will be the counter and have an instance of the counter in your top level module. It needs to have input ports for clock and enable and a std_logic_vector(15 downto 0) output port for the 4 digits of count — 4 bits per digit.

Implement the BCD digit counter as a module you use four times in the 4-digit counter module, once for each digit. The procedure needs two inputs, one for each of clock and enable, an output which is generated from count=9 ANDed with the

enable input that can be used as the enable for the next digit, and a std_logic_vector(3 downto 0) inout for the count. Note that if you use “out” here you will get an error message; you must use “inout”.





To turn in:

1. Report discussing what you did
2. VHDL source (with comments) for all modules incorporated in the document.
3. Design Summary showing successful synthesis.