Assignment # 7

Due Friday March 8

Read Chapter 7 Sections 1, 2, 3, 4.3 and 5.
Read Chapter 8 Sections 1, 2 and 4.
Information on Verilog can be found in the "Bucknell" TR
http://www.eg.bucknell.edu/ cs320/1995-fall/verilog-manual.html

1. Create a Gray code counter that counts with the sequence (000, 001, 011, 010, 110, 111, 101, 100, 000, ...). It should have a *reset* signal that sets it to 100 and an *enable* input that stops the counter (holds the current value) when low. The *reset* signal should work even if the *enable* signal is low.

   Do this in Verilog using a "case" statement. (Hint: Follow good design practices and layout a transition table and/or state diagram before doing the Verilog implementation.) For your simulation, (i.e., build a stimulus using the delay control # operator cf. Bucknell manual Section 2.9.1, or use a test vector) start by resetting the counter (leaving it disabled; it should reset even while disabled), then lower reset but leave it disabled for two cycles. Enable it and let it run for ten (10) cycles. Use either the $display statement (cf. Bucknell manual Section 4.2) or waveforms to test your simulation.

2. Chapter 7 Exercise 7.10 (a), (b) using only D flip-flops and toggle (T) flip-flops and (c). In addition,

   - Draw a schematic in DesignWorks using D flip-flops. Check that your design is correct.
   - Modify your counter so that it is self-starting. Draw a schematic in DesignWorks using D flip-flops. Check that your design is correct.
   - Implement the counter in Verilog using a "case" statement. Turn in your Verilog and simulation results

3. Chapter 8 Exercise 8.6

4. Chapter 8 Exercise 8.12