Introduction to Digital Design ECE 215
Summer 2005 Test 2

1. What is the difference between a *latch* and a *flip-flop*? (4 points)

2. Give truth tables for the following components (3 points each)
   (a) S-R latch

   (b) D latch

   (c) J-K flip-flop

   (d) T flip-flop

3. Draw the circuit symbol for a tri-state inverter (3 points)

4. Design a D-latch using tri-state inverters and ordinary inverters. (5 points)

5. Design a master-slave D flip-flop using two D-latches. (5 points)

6. Construct a JK flip-flop using a D flip-flop, a 2-to-1 line multiplexer and an inverter. (5 points)

7. Why do mechanical switches "bounce"? Give examples of circuts where the bounce matters and circuit where bounce does not matter. (4 points)

8. What is the difference between serial and parallel transfer of an n-bit word? (4 points)

9. Explain how to convert serial data to parallel. What kind of register is needed? (4 points)

10. The content of a 4-bit shift register is intially 1101. The register is shifted right six times with the serial input being 110101101 (least significant bit first). What is the content of the register after the sixth shift? (4 points)

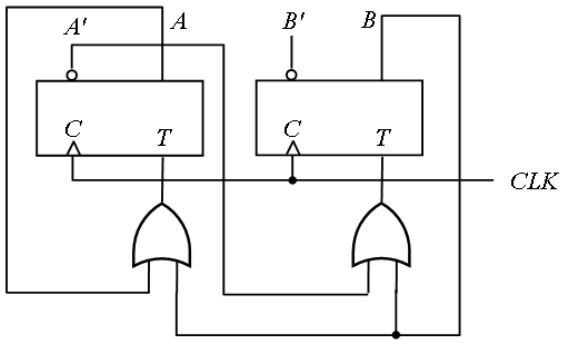
11. Design a four bit shift register with parallel load using D flip-flops (or express the design in Verilog). There are two control signals for the following operations: (10 points)

S_1	S_0	Register operation
0	0	No change
0	1	shift left (new bit enters from right)
1	x	Load parallel data

12. Design a decimal counter using D flip-flops (or express the design in Verilog) (10 points)

13. Explain the differences among a truth table, a state table, a characteristic table, and an excitation table. (5 points)

14. Derive the state table and diagram of the sequential circuit shown below. A and B are the states (and outputs). Describe the operation of the circuit. (15 points)



15. Draw the logic diagram of the sequential circuit (using D flip-flops) described by the following Verilog module: (10 points)

```
module Seqcrt(A,B,C,Q,CLK);
  input A, B, C, CLK;
  output Q;
  reg Q, E;
  always @ (posedge CLK)
  begin
    E <= A & B;
    Q <= E | C;
  end
endmodule
```