1) The content of a 4-bit shift register is initialized to 1111. The register is shifted to the right with the following input sequence: 101001101. The leftmost bit is shifted in first. What is the content of the register after each shift?

2) What is the difference between serial and parallel transfer? Explain how to convert serial data to parallel and parallel data to serial. What type of register is needed?

3) Draw the logic diagram of a 4 bit register with mode selection inputs $S_1$ and $S_0$. When $S_1S_0=00$, there is to be no change in the register contents. When $S_1S_0=01$, the value of the register is to be changed to zero. When $S_1S_0=10$ the register is to load a new value. When $S_1S_0=11$, the value is to be set to 15 ($1111_2$). All register changes should be synchronous to clock (i.e. don’t use asynchronous reset/clear). Your logic diagram should use only edge-triggered D flip flops and simple gates (AND, OR, NOT). Do not gate the clock signal.

4) Figure 5-5 in Mano and Kime is a design for a serial accumulator (they call it an adder, but is really an accumulator since it adds an input to the previous result). This design has a serious flaw; It gates the clock! Redesign the circuit so that it does not gate the clock. Implement your design in VHDL and simulate adding the following numbers: 1, 2, 3, 4. You might find it easiest to build a 4-bit register, and a full adder and then use these in your final design. Also be sure to reset your registers before you start entering the data. (NOTE: You might want to add a parallel output to Register A to make the result more obvious. Also think about how many times you need to clock this circuit to add four, 4-bit numbers.) Turn in your VHDL code and simulation output clearly showing the final result in Register A.

5) Again looking at Figure 5-5 in Mano and Kime, what is the maximum frequency you could clock this circuit? Assume that the clock is not gated and that the registers are built as you did in Problem 4. Further assume that the flip-flops have a propagation delay (clock-to-output) of $[2,5]$ ns ([min, max]), a setup time of $[1,2]$ ns and a hold time of $[0,1]$ ns; each logic gate has a delay of $[2,3]$ ns (AND/OR/XOR are all the same) and that you only have 2-input gates; the registers have one gate (AND gate) at the input to the flip-flops, as well (prove this to yourself!).

6) Design a counter that counts 1,2,3,5,8,13 and then repeats. Show the equations for each flip-flop (next state equations). Implement your design in VHDL. You may implement this with your equations or in a behavioral way. Be sure to show a simulation that shows the count sequence and that is goes back and starts over.
7) Consider the synchronous ripple carry counter on page 5-10 of the notes. Assume that the D flip flops have a setup time requirement of 3 ns and that the gates in the circuit all have a propagation delay of 2 ns, max.

   a. If the enable input changes from 0 to 1 at time 0, what is the earliest time that the clock can change from 0 to 1 without violating setup time requirements, assuming the initial value of the counter is 1?

   b. What if the initial value of the counter is 3?

   c. What if it is 7?

   d. If we extend the counter to 16 bits, how much time must there be between a change to the enable input and the next rising clock edge if we are to avoid setup time violations?

8) A 4 bit twisted ring counter is a sequential circuit that produces the following sequence of output values: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001 and then repeats. Design a circuit for a 4 bit twisted ring counter that uses four D flip flops. Draw a state transition diagram, a state table and a schematic for your circuit. Also include a synchronous reset input that will force the state to 0000 when reset is low. Use as few gates in addition to the four flip-flops as possible.