

ECE241F - Digital Systems - Lab 4

Sequential Logic and Hierarchical Design

Fall 1999

A. Carusone, P. Lehn, J. Rose

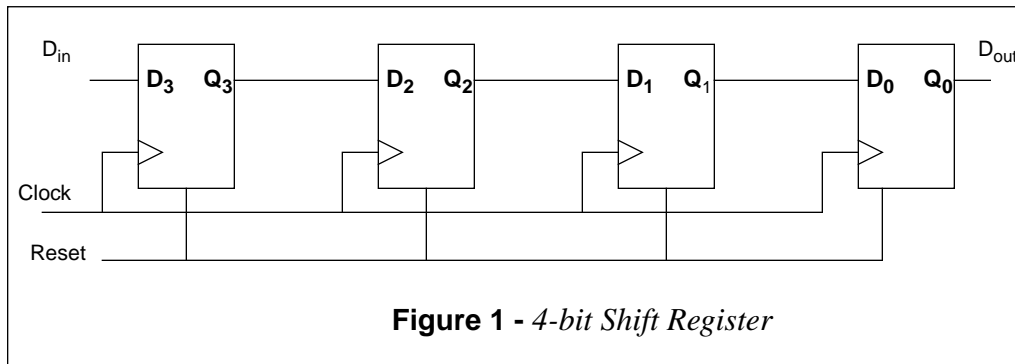
1.0 Purpose

The purpose of this lab is to learn about the basic operation of sequential logic (circuits with memory), by building them up from basic gates. You will also build a shift register from flip flops, and learn how to invoke flip-flops from the VHDL language. As a side-effect of building bigger things from smaller things, we illustrate the concept and use of hierarchy in design.

2.0 Background

In class we have covered the basic circuit for RS latches, D latches, clocked D latches and master-slave edge triggered D flip-flops, so you should review your notes on these.

A *shift register* is typically used to send and receive data, one bit at a time, under the control of a clock signal. Figure 1 below illustrates a 4-bit shift register built out of positive-edge triggered D-type flip flops. On each positive clock edge, the value of the signal D_i is copied onto the value of Q_i . Thus it takes four clock cycles for the original value of D_{in} to appear as D_{out} . The entire contents of the shift register can be initialized by resetting each flip-flop with a reset signal.



3.0 Preparation

Design and simulate (using timing simulation) the following circuits, using the graphic editor of maxplus2 (DO NOT USE VHDL, except for part 5). You should design all of these circuits with gates from the “prim” library. Make a separate directory for each of the circuits you design. Parts 1, 2, and 3 concern the basic operation of circuits that have memory, and so we have you design these from basic gates. Part 3 also demonstrates the use of hierarchical design in maxplus2.

1. A set-reset latch, using cross-coupled 2-input NOR gates.

2. A level-sensitive clocked D latch, based on the set-reset latch you designed above.
3. Turn the D-latch into a symbol (using **File | Create Default Symbol** method as described in Tutorial 1) and use it to design a positive edge-triggered master-slave D flip-flop. By doing this, you are creating hierarchy, which is the essential method of all large scale design.
4. Create a four-bit shift register (as illustrated above in Figure 1) from D flip-flops, and turn it into a symbol as you did for the D-latch in part 3. **Instead of using the flip-flop you designed in part 3, use the D flip-flops available in the “prim” library (as the symbol “DFF”).** Make sure that the four outputs Q_3, Q_2, Q_1 , and Q_0 , are available as outputs from your shift register. Recall that there are D flip-flops inside the basic macrocell of the MAX 7128, and you need not build them from scratch.

Design, enter and simulate a circuit that connects the outputs of the shift register ($Q_3Q_2Q_1Q_0$) to the inputs $X_3X_2X_1X_0$ of your circuit from Part 1 of Lab #3. The inputs to your circuit should be a single D_{in} signal and the clock. The point here is to generate the inputs to your circuit from Lab #3, serially, one bit at a time, rather than in parallel, four bits at a time. You will again need define a symbol for your circuit of Lab #3, and use hierarchy in the graphic editor to build this circuit.

5. Build and simulate an edge-triggered D-type flip flop as described in the VHDL Reference guide, Section A.10.2 on page 717 of Appendix A of the text. Build and simulate a 3-bit D register the same way, with an asynchronous reset signal as described in Section A.10.4.
6. Building a waveform generator. Build a 10-bit shift register in the manner that was describe in part 4 above, including the “reset” signal. Connect the output of the final bit’s Q output to the first bit input of the D register so that the shift register actually forms a ring. Notice that a DFF has both a “Reset” input (labelled CLRN) and “Preset” input (labelling PRN), which are active low. CLRN sets the bit to be a low, and PRN sets it to be a high value, asynchronously with the clock. Connect your reset signal to some of the flip flop’s CLRN inputs, and connect some to the PRN input(you choose). In your simulation, activate the reset for an appropriate period, which effectively loads a value into the shift register. Continuous clock of this circuit produces a waveform of your design, based on which flip-flops you reset or present. In the lab you will connect the clock of the flip flops to the on-board system clock that runs at 25MHz. When you operate the circuit, activate the reset, and then observe the output of the

4.0 In The Lab

Implement and test all of the circuits you designed in the preparation. Show each working part to a Teaching Assistant.

Note: you must set maxplus2 appropriately to allow the clock to come in on a regular I/O pin, except for part 6, in which you should use the on-board clock.

For Part 6: use the logic analyzer to observe the output of the osciallator.