1 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 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 Di is copied onto the value of Qi. Thus it takes four clock cycles for the original value of Din to appear as Dout. The entire contents of the shift register can be initialized by resetting each flip-flop with a reset signal.

3 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

![Figure 1: 4-bit Shift Register](image-url)
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 Q3, Q2, Q1, and Q0, 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.
5. 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 Din 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.
6. Build and simulate an edge-triggered D-type flip flop as described in the VHDL Reference guide, Section A.10.2 on page 20 of Appendix A of the lab manual. Build and simulate a 3-bit D register the same way, with an asynchronous reset signal as described in Section A.10.4.

4 In The Lab

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

BEWARE: there is a problem with the digital switch board in the lab. The transition of one switch can cause a "glitch" on the other signals. If those signals are connected to something that it edge-activated, such as a clock signal, your circuit will not appear to function correctly. One way to solve this is to make the clock active on a negative edge (not the usual positive edge). The reason this works is tha the glitch appears to create spurious positive edges.

Also: you must set maxplus2 appropriately to allow the clock to come in on a regular I/O pin.