

# MEMORY ELEMENTS AND SEQUENTIAL LOGIC

ECE 221

Fall 2003

## I. OBJECTIVE

The objective of this experiment is to study how an SR latch can be used as a clock and to analyze a simple sequential circuit.

## II. BACKGROUND

Most inputs to sequential systems are not connected directly to mechanical switches because those types of switches cause the input signals to bounce between a '1' and '0' value. The bouncing is due to how the mechanical parts are built (a solid steady contact may not be made 100% of the time). The figure below shows how a SR latch can be used as a debouncing circuit.

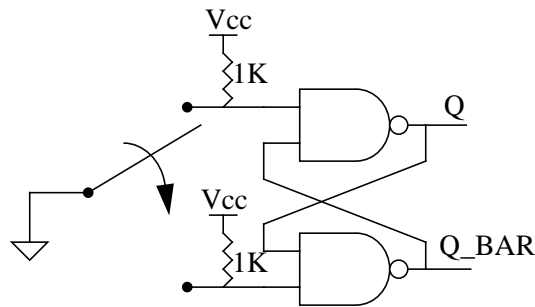


FIGURE 1. Debouncing Circuit Using 2-Input NAND Gates

By attaching the circuit to a mechanical switch you are guaranteed that no bouncing will occur.

## III. PRE-LAB

1. Answer the following questions about the SR latch shown above.
  - Create a truth table for the latch. What values on the inputs will cause you to "hold" the current values of Q and Q\_BAR?
  - Describe how this circuit eliminates bouncing? In other words, what happens if the switch is in-between the two inputs?
2. Review the Altera design flow from the last lab.

## IV. PROCEDURE

1. Build a SR latch debouncing circuit described in the Background section using 2-input NAND gates on a protoboard. This latch will serve as the manual clock for the other circuit you will build. The switch can be modeled using a wire between ground and the inputs. Connect an LED to the output Q of your latch to monitor its operation. Make sure your circuit is operating properly before going to the next step.
2. Create an Altera schematic of your BCD sequence recognizer using your Mentor schematic as a guide. Follow the procedure from the previous lab. The notes given below will also help you in your development.
  - Use the parts in the prim library for the simple gates, flip flops, power and ground. For the D flip

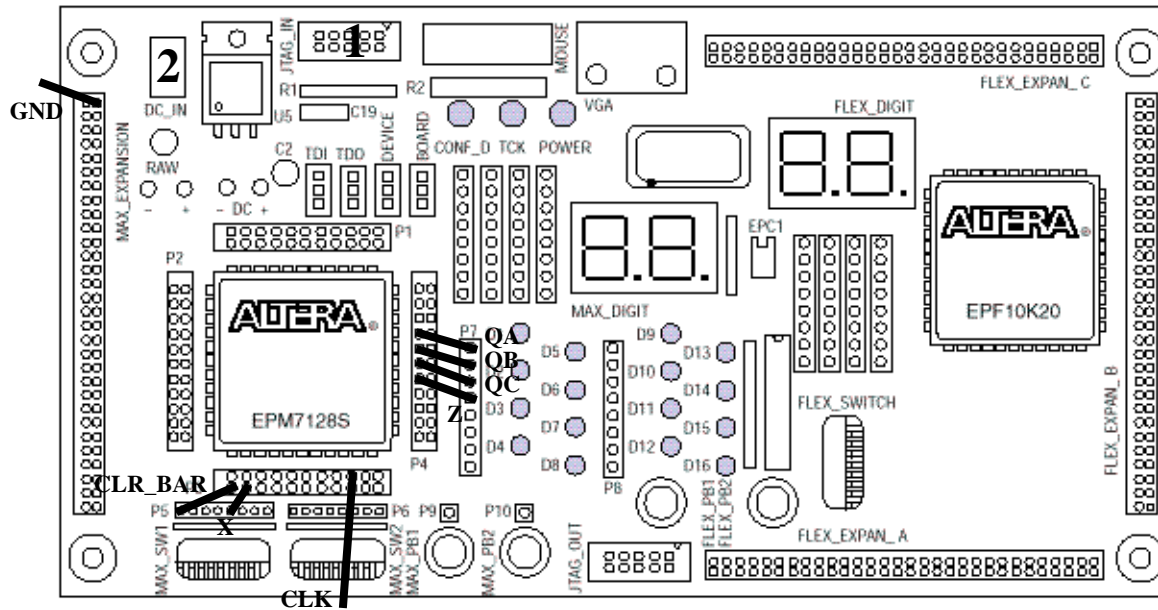
flop use the **dff** component (which should match your Mentor flip flop except it does not have a **Q\_Bar** output) and for the +5 volts use the **vcc** component. Make sure you also add a title block and set the size to **C. r**

- Assign output ports to the not only your output **Z** but also the outputs of the three flip flops. These will represent your state. Use as the port names **QA**, **QB** and **QC**. Remember that the lights on the Altera board are Active-0 so you will need to invert your outputs.
- Make sure you assign the correct device and then assign the following pin numbers to your design:

CLR_BAR	33
X	35
CLK	50
QA	69
QB	67
QC	65
Z	63

- After saving your design, set the project to the current file so that the compiler know to compile this design.
  - Before you compile your design, select **Global Project Logic Synthesis** under the **Assign** menu. When the window appears, click off the **Clock** and the **Clear** under the **Automatic Global** setting and hit **OK**. This will allow the compiler to route your clock signal to pin 50 instead of to the dedicated clock pin (pin 83).
  - Once completed print out your design schematic and put it in the Appendix.
3. Once your schematic is created verify its operation. Instead of using a vector file, you will manually create the waveforms in the waveform editor as outlined below:
- Open up the waveform editor.
  - Select **Enter Nodes from SNF** under the **Nodes** menu. Then when the window pops up, hit the **List** button, followed by the => and then **OK**. All of your ports should now appear in the waveform editor window.
  - Create waveforms by holding down the mouse and selecting a portion of the time you want changed and then clicking on the **0** or **1** value box on the left-hand side palette. If you want to create a clock, select the clock signal and click on the **C** value on the same palette. Then when the clock generation window appears, hit **OK**. Simulate two BCD values, one that causes the output to be asserted and one that doesn't
  - Remember to save the waveforms before simulating.
  - Once the simulation is correct, print out the waveforms and put them in the Appendix.
4. Wire up the Altera education board as shown on the next page:
- Connect the two inputs, CLR\_BAR and X, to switches.
  - Connect the CLK to the Q of your debouncer circuit. Since we are using two different boards you should have a common GND between them. tie the GND from the Altera board to the ground on your extra protoboard. The GND on the Altera board can be found on pins 2, 4 and 6 on the expansion slot (which is the two columns of holes to the left of the chip. Connect a wire from one of those holes to the ground line on your protoboard.

Figure 1. UP 1 Education Board Block Diagram



5. Program the Altera education board and verify that your design is working correctly. Once you have your design working correctly, show the operation of your circuit to the instructor or a TA and get his/her signature.

## V. SUMMARY and ANALYSIS

Write a paragraph summarizing today's lab and then separately answer the following questions.

1. If you cannot guarantee that the initial power-up state will be the reset state (000) what should your output be in any unused states? Why?
2. Could you use the function generator at your lab station to create a clock that would run your sequential circuit? If so, how? If not, why not?