ECE 152A LABORATORY 3

OBJECTIVE:

Implement a counter using only D-flip-flops as memory elements. This counter will reset to an initial state of all zeroes with a reset switch on the bread-board. This counter takes a push button switch as input. If the pushbutton switch is ON, then the counter counts upwards from 0 to 5 and repeats these numbers indefinitely. If the pushbutton switch is OFF, then the counter counts downwards from 5 to 0 and repeats this pattern indefinitely. If the pushbutton switch is turned from ON to OFF at any point in the sequence, the counter which was counting up starts counting down from that point onwards. For example, assume that you the pushbutton switch is ON and the sequence is 0, 1, 2 and then the pushbutton switch is turned OFF. Then, the complete sequence will be as follows: 0, 1, 2, 1, 0, 5, 4, 3, 2, 1, 0, ... until the pushbutton switch becomes ON again, in which case it will again switch direction. Similarly, when the pushbutton switch is turned from OFF to ON at any point in the down-counting sequence, the counter starts counting up from that point onwards.

Reset:

When you power up your board the D-flip-flops will power up in random state and it is usually desired that the flip-flops be brought to a known initial state. Following this rule, you will have to reset the flip-flops in the design to count 0 by operating the *reset switch* on your bread-board. For example, in the above sequence 0, 1, 2, 1, 0, 5, 4, 3. Assume that reset is applied at this point, then the circuit should **synchronously** reset to 0. Assuming that the pushbutton switch did not change at reset, the resulting sequence is 0, 1, 2, 1, 0, 5, 4, 3, 0, 5, 4, 3, 2, 1, ... The synchronous nature of the reset will follow from the way we will build this machine (it will be a Moore machine; hence the outputs will change only at the active clock edge). So, synchronous reset will be guaranteed by the way we build the circuit.

Deliverables and Due Dates:

Lab # 3 problem set is due beginning of your lab section.
This comprises Parts 1, 2, 3 and 4 below.

Lab # 3 check-out is the following week. This comprises Part 5 below.

Grading:

Grading will be done as follow for the steps completed on time:

- Lab # 3 Problem Set:
  - Part 1: 5%
  - Part 2: 10%
  - Part 3: 25%
  - Part 4: 20%
- Lab # 3 Check-out:
  - Part 5: 40%

Design Strategy:

- **Part 1:**
  - On paper, play with some sample input combinations to understand the operation of this design completely. For example:
    - What should be the initial state when the reset switch is hit?
    - What will happen when you press the up-down switch (it is a switch that decides if the counter counts up or down) when it is counting in certain direction?
    - What will happen if reset is asserted and the up-down switch changes position in the same clock cycle? What will be the next state?
  - You will be designing a Moore machine for this lab.
  - Definition:
    - Moore Machine: If the output of a sequential circuit is a function of only the present state (i.e. not any of the current inputs), then this sequential circuit is referred to as Moore machine.

- **Part 2:**
  - Clearly define:
    - The state variables you use.
    - The inputs and outputs you use.
  - Draw a state table for this counter (refer to Figure 8.79 (a) of the text-book for a sample state table).
    - Clearly show each combination of present states and inputs that you use.
    - Clearly show all don't care combinations in your state table.
    - Clearly show the next states of the flip-flops.
    - Show the outputs (7-segment display) (for each valid combination of the next state) in your state table.
  - Draw a state diagram for this counter (refer to Figure 8.23 and 8.28 of the text-book for a sample state diagram).
    - Show all the valid states in the state diagram with the values of the inputs clearly shown besides each edge connecting the nodes.
- **Part 3:**
  - Carefully draw a K-map (make sure you covered all the 1's in the K-map and did not forget to put X's for don't care states) for each of the ***outputs*** and the ***next-state variables*** of this circuit (***next-state*** are the outputs of the D-flip-flops and ***outputs*** are the inputs of the 7-segment display).
  - Use logic minimization techniques to arrive at a minimum sum-of-products for each of these.
  - Note: You may need 5-variable K-maps in doing this. Make sure you group terms on this K-map to follow adjacency relationships on a 5-dimensional Boolean cube.

- **Part 4:**
  - Map the resulting design to a schematic that uses D-flip-flops as memory elements.
    - Carefully transform the Boolean equations obtained from the K-maps into a schematic of logic gates and D-flip-flops.

- **Part 5:**
  - Implement the design on your breadboard using TTL chips, the seven-segment display and a push-button switch. Carefully transform the schematic of the part 4 into an actual design using TTL chips.
  - As a design strategy, prepare a rough design with input switch, TTL chips, 7-segment display and wires on a paper. This will tremendously help you in troubleshooting your wiring on bread-board.

Implementation Notes:

1. Wire the outputs of the counter to a seven-segment display.

2. For providing the clock signal to the D-flip-flops you have to use a function generator. In order to see the count, set the input signal frequency to 1 Hz. Remember that the offset should be 1.25 V and the amplitude 2.5 for the square wave.

3. D-flip-flops: You may use positive edge-triggered D flip-flops for your design. Read the data sheet of this component for correct timing and to ensure functionality. The ClrN input on this D flip-flop is an asynchronous reset input. The "N" in "ClrN" stands for "negative" indicating the bubble. When ClrN == 0, the D flip-flop does an asynchronous reset. In this lab, connect the ClrN and PRE to "1" in order to disable asynchronous "reset" and "preset".

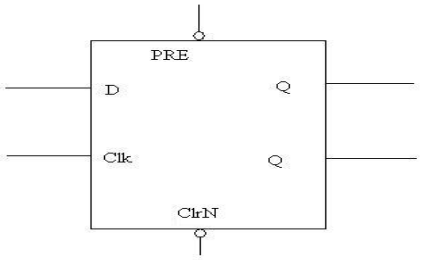

Figure 1: D Flip-Flop Schematic Symbol

4. Push Button Switch:

- The push button switch should be wired as shown in the figure below Push button switches typically have a lot of "*bounce*" in the output signal when the button is pressed. Severe bounce in a signal can lead to improper or damaging logical values to be propagated into the circuit. The addition of capacitor will clean up the signal to make an appropriate logic high or logic low level.

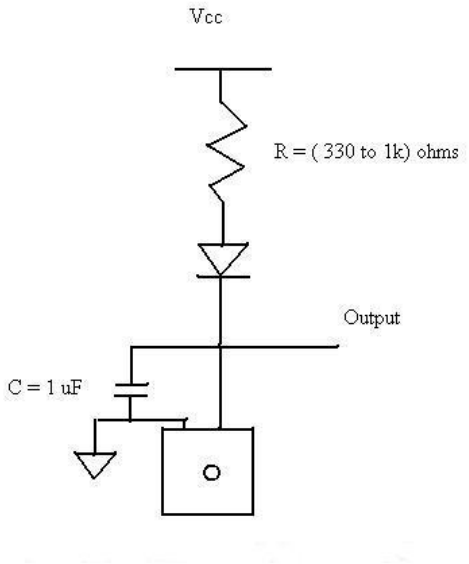

Figure 2: Schematic for Connecting the Switch

5. Use TTL chip 74LS48 (BCD to 7 segment decoder) to convert the binary output of the D-flip-flops into 7 segment display.