EXPERIMENT 8

FLIP FLOPS AND SEQUENTIAL CIRCUITS

I. INTRODUCTION

1. Objectives

The objective of this experiment is to become familiar with the basic operational principles of flip-flops and counters.

II. PRELIMINARY WORK

1. When you use switches like push buttons you face the bouncing problem due to the mechanical contact. The input signal produced by a switch is shown below. The circuit may be triggered many times although triggering only once is intended. The debouncing circuit shown in Figure 7-1 is a practical solution to this problem. Explain the operation of this circuit.

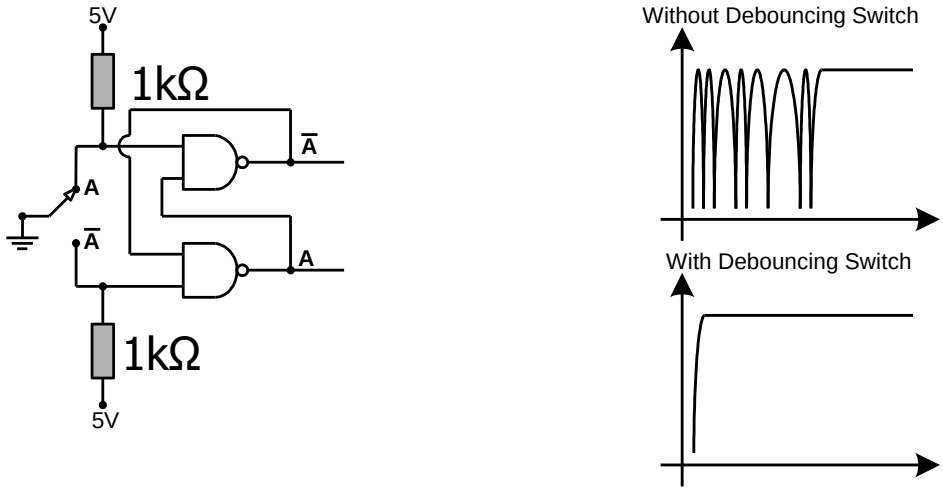


Figure 7-1. The schematic of a debouncing circuit and signals that can be observed with and without a debouncer.

2. Design a circuit with two inputs, X, Y, and one output Q_{n+1}, which has the truth table given in Table 7-1, by using a J-K flip flop.

Table 7-1. The truth table of the circuit asked to be designed.

X	Y	Q_{n+1}
0	0	0
0	1	Q_n'
1	0	Q_n
1	1	1

3. Figure 7-2 shows a circuit constructed with elementary gates and JK flip flops, in which all of the FFs are positive edge triggered. Write down the present state $Q1_n$, $Q2_n$, $Q3_n$ in terms of the previous state $Q1_{n-1}$, $Q2_{n-1}$, $Q3_{n-1}$, and prepare a table showing the state of the circuit after each clock pulse, starting from the (0, 0, 0) initial state.

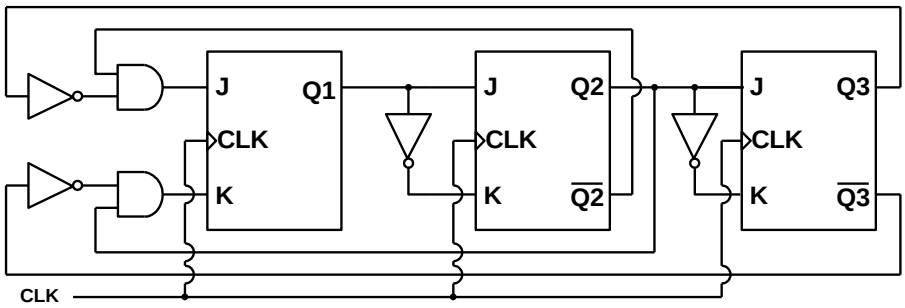


Figure 7-2. The schematic of a circuit constructed with elementary gates and JK FFs.

4. Figure 7-3 shows the schematic of a frequency divider circuit. Verify the operation of this frequency divider. Determine the frequencies of each output if the input CLK frequency is f.

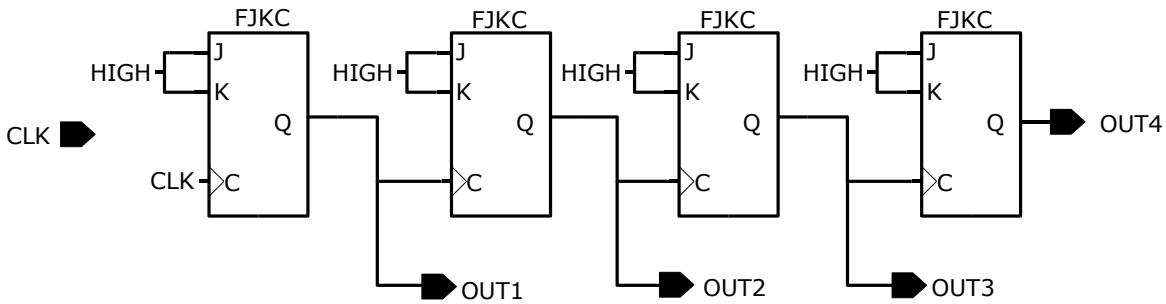


Figure 7-3. The schematic of a frequency divider circuit implemented with JK FFs.

5. It is known that the circuit in Figure 7-3 is also a ripple **down** counter. Modify the circuit so that it counts **up**.

6. Design a BCD counter using a 4-bit **binary counter** (not a ripple counter) and simple gates.

III. EXPERIMENTAL WORK

Debounce Circuit

1. Construct the debounce circuit given in Figure 7-4. Connect node A to channel A1 and node B to channel A2 of the scope. Choose normal triggering from trigger mode, set the triggering level to 1V and the triggering edge to rising edge. Press the "single" key. Operate the switch and observe the waveforms on the scope. Comment on the result.

when the switch is off

- ❖ connection 1: open circuit
- ❖ connection 2: short circuit

when the switch is on

- ❖ connection 1: short circuit
- ❖ connection 2: open circuit

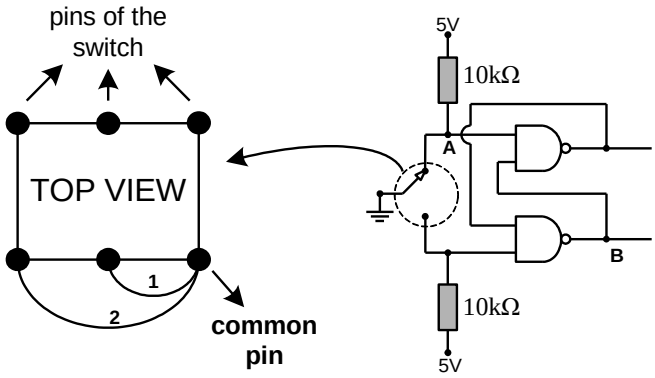


Figure 7-4. The schematic of a debouce circuit and the connections of the switch.

NOTE :

Do not change the circuit; you will use the same circuit in the next part.

Up/Down BCD Counter

2. Construct a BCD counter on the protoboard. You will use one 74LS190, one 74LS47, and one 7-segment display. Apply a CLK signal of 1Hz from the function generator and observe the output. The **(CE)'** input of 74L5190 must be **low** and the outputs Q_0, Q_1, Q_2, Q_3 must be connected to the A, B, C, D inputs of 74LS47 respectively. Figure 7-5 shows the pin connections of the seven segment display.

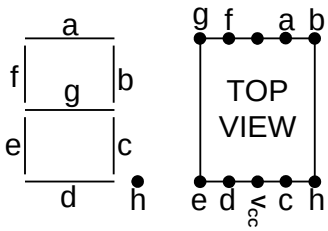
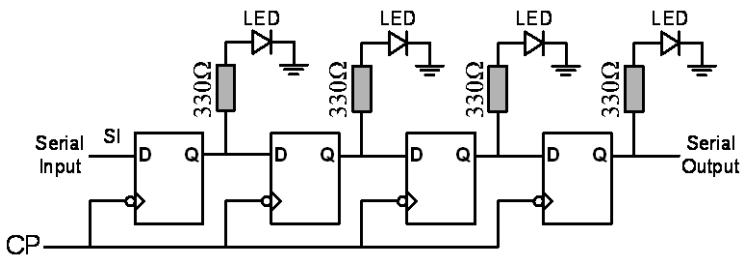


Figure 7-5. Pin connections of the 7-segment display.

3. Now, apply CLK signal from a simple switch, and then from a debounce switch. Observe the difference. Did you have what you were expecting?

Shift Register

4. Construct the shift register given in Figure 7-6 by using two 74LS74 ICs.



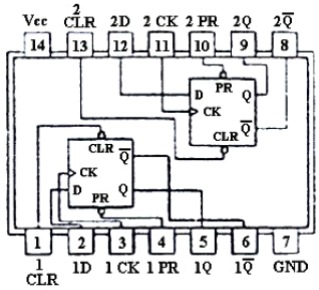


Figure 7-6. The schematic of a 4-bit shift register & the pin configuration of the 74LS74.

5. Connect the output of the function generator with settings of 5V amplitude, 2.5V offset, and 1kHz frequency, to the clock inputs of the D flip-flops. Connect the D input of the first D-FF to LOW and observe the LEDs.

6. Connect the D input of the first D-FF to HIGH and observe the LEDs.

7. Assume that we want to have only one HIGH output, and this HIGH output will circulate through the registers. That is, this HIGH output should go from the output of first D-FF to that of 4th D-FF and return back to the output of the 1st D-FF again and continue to circulate as shown in Table 7-2. What should you do to observe the bits correctly?

Table 7-2. Outputs of the DFFs that circulate a single HIGH bit.

	Output Of 1st DFF	**Output Of 2nd DFF**	**Output Of 3rd DFF**	**Output Of 4th DFF**
1st CLK Pulse	1	0	0	0
2nd CLK Pulse	0	1	0	0
3rd CLK Pulse	0	0	1	0
4th CLK Pulse	0	0	0	1
5th CLK Pulse	1	0	0	0
6th CLK Pulse	0	1	0	0
…	…	…	…	…

8. Explain the operation of the Shift Register.

*****In the rest of the laboratory, you will work on Xilinx.*****

Binary Ripple Counter

9. In this 1st Part, you will design a circuit to use the CLK generator on the Xilinx board. For this purpose, construct the circuit given in Figure 7-7, and create a hierarchical view (name as CLKGEN).

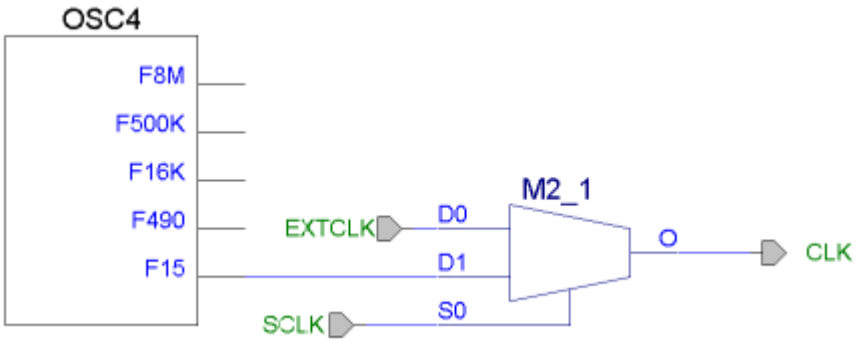


Figure 7-7. The schematic of the CLK generator circuit.

IMPORTANT NOTE :

OSC4 is the internal CLK generator of the Xilinx board. However, since this block is an analog block, it is not possible to simulate it using the digital simulator of Xilinx. For this reason, during digital simulations, you will apply an external clock (EXTCLK). The circuit chooses the internal clock or the external clock with SCLK input. Shortly, during the simulations you will apply LOW to SCLK, and assign BC0 bit of the binary counter in the stimulator to EXTCLK. The frequency of the internal CLK is 15 Hz.

10. Construct a frequency divider which is given in the preliminary part of the experiment, in order to divide the input frequency by 16. Simulate the circuit and create a hierarchical view (name as FDIV). Explain the operation of the circuit. Note that using CLKGEN and FDIV together, you obtain an internally generated, almost 1Hz CLK signal.

11. Construct the 4-bit ripple up counter that you designed in the preliminary work. Simulate your circuit and create a hierarchical view (name as CNT4). During simulations, assign BC0 to CLK.

12. Now, use CLKGEN, FDIV, and CNT4 together in order to construct a ripple up counter which uses 1Hz internal generated CLK of Xilinx board, as seen in Figure 7-8. First perform "Functional" simulation to verify the operation of the circuit. Then perform "Timing" simulation to observe the delays between output bits. Note that, during Timing simulation, you need to apply HIGH to CLR signal for a certain time in order to initialize the initial states of the FFs. What is the reason for the delays of the outputs?

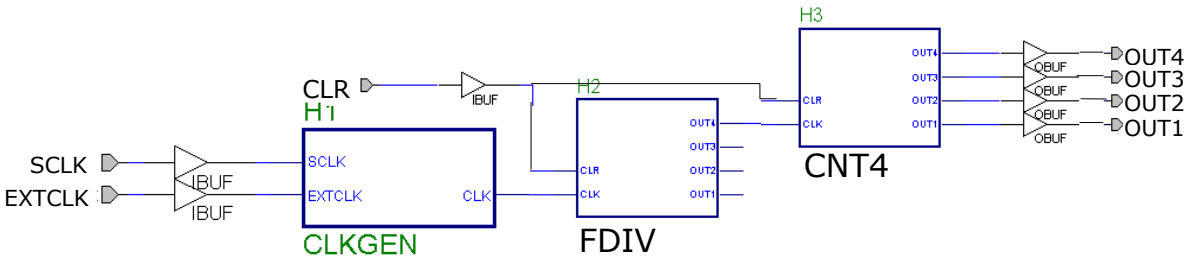
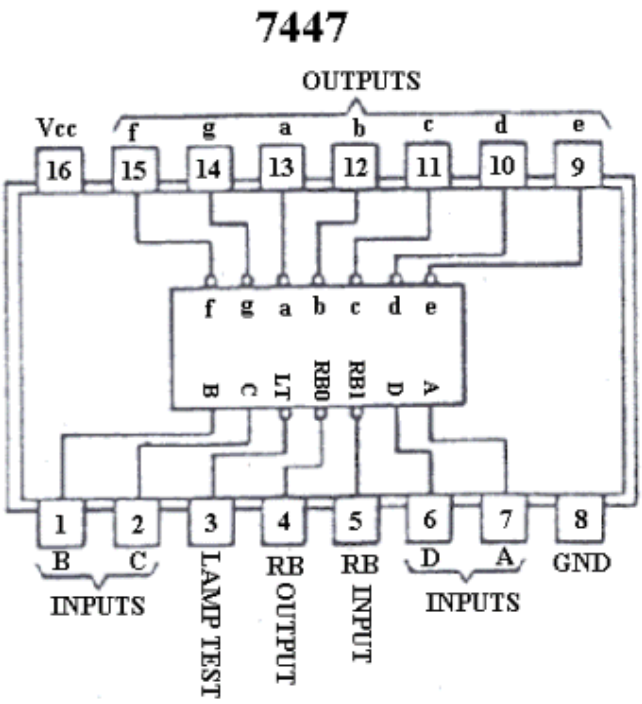


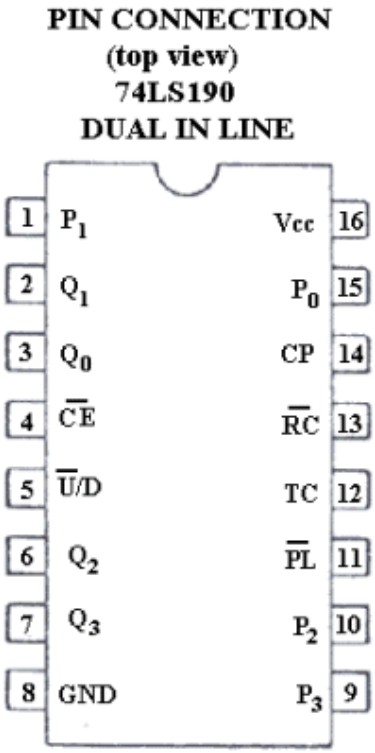
Figure 7-8. The schematic of a ripple up counter by using CLKGEN, FDIV and CNT4.

13. Download the circuit to Xilinx board. Test your circuit using both internally generated and external CLK signals.

BCD Counter

14. Construct a BCD counter using CLKGEN and FDIV similar to the previous case, one CB4CE as the 4-bit binary counter, and some simple gates which you have designed in the preliminary work. Simulate your circuit and download to Xilinx board.





PIN NAMES FOR 74LS190

CE'	Count Enable (Active LOW) Input
CP	Count Pulse (Avtive HIGH going edge) Input
U'/D	Up/Down Count Control Input
PL'	Parallel Load Control (Active LOW) Input
P_n	Parallel Data Inputs
Q_n	Flip-Flop Outputs
RC'	Ripple Clock Outputs
TC	Terminal Count Output

IC LIST FOR EXPERIMENT 7

74LS190 BCD up/down counters

74LS47 7-segment display driver

74LS00 Four 2-input NAND gates

74LS74 D-FF

7-segment display