MÜHENDİSLİK FAKÜLTESİ
FACULTY OF ENGINEERING
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY

DUMLUPINAR BULVARI 06800
ÇANKAYA ANKARA/TURKEY
T: +90 312 210 23 02
F: +90 312 210 23 04
ee@metu.edu.tr
www.eee.metu.edu.tr

# **EXPERIMENT 8.** Flip-Flops and Sequential Circuits

## I. Introduction

### I.a. Objectives

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

## II. Preliminary Work

**1. Design** a circuit with two inputs and one output, i.e. X, Y, $Q_{n+1}$ respectively, which has the truth table given in Table 8.1 by using a JK flip-flop. **Show** your design with full justification.

**Table 8.1** Truth table of the circuit which is mentioned in II.1

| **X** | **Y** | **$Q_{n+1}$** |
|---|---|---|
| 0 | 0 | $Q_n$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $Q_n'$ |

**2.** Figure 8.1 shows a circuit constructed with elementary gates and JK flip-flops, in which all of the flip-flops are positive edge triggered. **Write down** the present states of $Q_{1,n}$, $Q_{2,n}$, and $Q_{3,n}$ in terms of the previous states $Q_{1,n-1}$, $Q_{2,n-1}$, $Q_{3,n-1}$. **Prepare** a table showing the state of the circuit after each clock pulse, starting from the state (1, 1, 1).



(e-mail: yunuscangultekin@gmail.com)

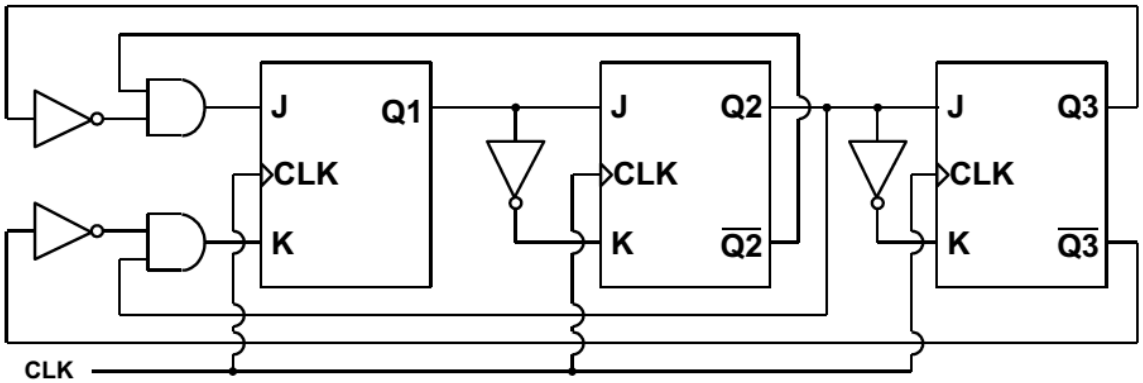


**Figure 8.1** The schematic of the circuit which is mentioned in II.2

3. **Design** a frequency divider circuit with four JK flip-flops. There should be one input clock with frequency of $f_{clock}$ and four outputs whose frequencies will be $f_{clock}/2, f_{clock}/4, f_{clock}/8, f_{clock}/16$. **Show** your design with full justification.

4. It is known that the circuit you designed in Part II.3 is also a ripple down counter. **Modify** the circuit by adding elementary gates so that it counts up and has a *clear* input so that when the clear input is HIGH all outputs are LOW. **Show** your design with full justification.

5. **Design** a BCD counter using a 4-bit ripple up counter that you designed in Part II.4 and elementary gates. **Show** your design with full justification.

6. **Design** a BCD counter with Verilog. How you can create a Verilog module in Quartus II is explained in the Experimental Work section. **Show** your simulation results.

## III. Experimental Work

### III.a. Up/Down BCD Counter

1. **Construct** the 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 <u>low</u> 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 8.2 shows the pin connections of the seven segment display.

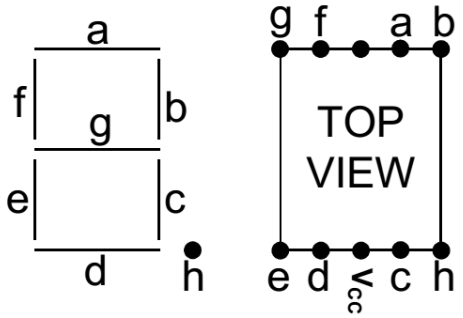


**Figure 8.2** Pin connections of the 7-segment display

### III.b. Shift Register

2. **Construct** the shift register given in Figure 8.3 by using two 74LS74 ICs.

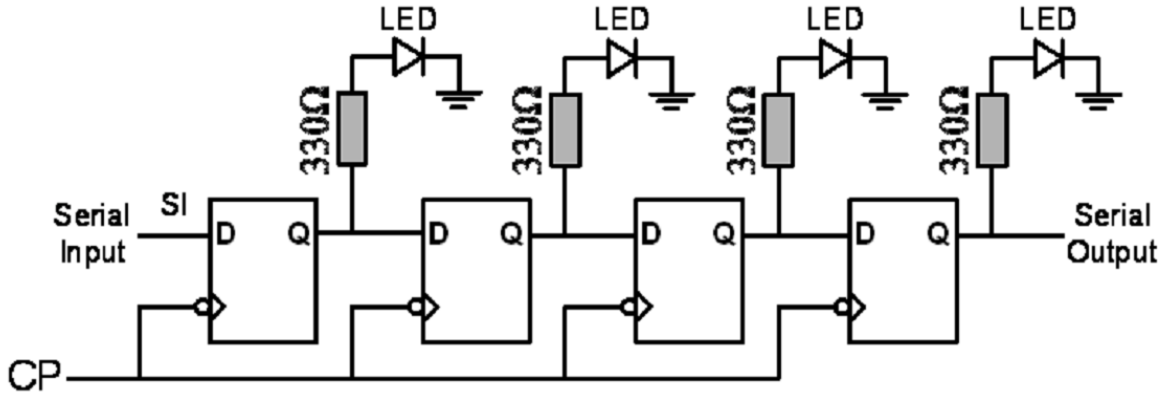


**Figure 8.3** The schematic of a 4-bit shift register

3. **Connect** the output of the function generator with settings of 5 $V_{p\text{-}p}$ amplitude, 2.5 V offset, and 1 Hz frequency, to the clock inputs of the D flip-

flops. Connect the D input of the first D flip-flop to LOW and observe the LEDs.

4. Connect the D input of the first D flip-flop to HIGH and observe the LEDs.

5. 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 8.2. **What** should you do to observe the bits correctly?

**Table 8.2** Outputs of the D flip-flops that circulate a single HIGH bit

| | **o/p of 1st D flip-flop** | **o/p of 2nd D flip-flop** | **o/p of 3rd D flip-flop** | **o/p of 4th D flip-flop** |
|---|---|---|---|---|
| **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** |
| **…** | **…** | **…** | **…** | **…** |

6. Explain the operation of the Shift Register.

**IN THE REST OF THE LABORATORY YOU WILL WORK ON QUARTUS II**

### III.c Binary Ripple Counter

7. Firstly, in order for the binary counter to work; an external clock should be applied to the FPGA. For this purpose, you should use a GPIO pin as an input and apply a clock from this pin. Basically you will connect the function generator's output to this pin.

   **NOTE:** You should adjust the signal generator such that it gives 3.3 $V_{p-p}$ square wave of 16 Hz frequency and 1.65 V offset.

8. **Set** the function generator output to a square wave of 3.3 $V_{p-p}$, 1.65 V offset and at 16 Hz frequency. This will be our externally generated clock.

9. **Construct** the frequency divider which you designed in the preliminary part of the experiment, in order to divide the input frequency by 16. **Simulate** the circuit and **create** a symbol for hierarchical design purpose.

10. **Explain** the operation of the circuit. Note that using externally generated clock and frequency divider together, you will obtain a 1Hz clock.

11. **Construct** the 4-bit ripple up counter that you designed in the preliminary work. **Do not forget** to add a *clear* input. **Simulate** your circuit functionally and **create** a symbol for hierarchical design purpose.

12. Now, use frequency divider and 4-bit ripple up counter that you have just designed together in order to **construct** a ripple up counter which uses 1 Hz clock. Assign a push-button to the *clear* input of your design. **Perform** "Functional" simulation to verify the operation of the circuit.

13. **Test** the circuit on FPGA board.

**III.d BCD Counter**

14. Construct a BCD counter using the frequency divide and 4-bit ripple up counter that you have designed in Part III.c and elementary gates. Assign a GPIO pin to the clock input of your design. Simulate your circuit functionally and test it on FPGA board.

**III.e BCD Counter with Verilog**

15. In this part you will implement the BCD counter that you have designed in preliminary work.

16. Open a new project.

17. Open a new Verilog file by **New -> Design Files -> Verilog HDL File**. The name of the file must be same as module name.

18. Write your code here. When you are done, don't forget to save.

19. **Analyze & Synthesize** your module. After analysis & synthesis is completed successfully, you can see your module's schematic view from **Tools->Netlist Viewer -> RTL Viewer**.

**20.** The rest is same with the previous labs (**Do** Functional Simulation, **assign** pins to inputs and outputs, **compile** your design). Please **use** one of the push button as the clock input. **Test** your design on the FPGA.

## V. FPGA Pin Assignment Codes

| *Signal Name* | *FPGA Pin No.* | *Description* | *I/O Standard* |
|---|---|---|---|
| SW[0] | PIN_AB12 | Slide Switch[0] | 3.3V |
| SW[1] | PIN_AC12 | Slide Switch[1] | 3.3V |
| SW[2] | PIN_AF9 | Slide Switch[2] | 3.3V |
| SW[3] | PIN_AF10 | Slide Switch[3] | 3.3V |
| SW[4] | PIN_AD11 | Slide Switch[4] | 3.3V |
| SW[5] | PIN_AD12 | Slide Switch[5] | 3.3V |
| SW[6] | PIN_AE11 | Slide Switch[6] | 3.3V |
| SW[7] | PIN_AC9 | Slide Switch[7] | 3.3V |
| SW[8] | PIN_AD10 | Slide Switch[8] | 3.3V |
| SW[9] | PIN_AE12 | Slide Switch[9] | 3.3V |

| *Signal Name* | *FPGA Pin No.* | *Description* | *I/O Standard* |
|---|---|---|---|
| KEY[0] | PIN_AA14 | Push-button[0] | 3.3V |
| KEY[1] | PIN_AA15 | Push-button[1] | 3.3V |
| KEY[2] | PIN_W15 | Push-button[2] | 3.3V |
| KEY[3] | PIN_Y16 | Push-button[3] | 3.3V |

| *Signal Name* | *FPGA Pin No.* | *Description* | *I/O Standard* |
|---|---|---|---|
| LEDR[0] | PIN_V16 | LED [0] | 3.3V |
| LEDR[1] | PIN_W16 | LED [1] | 3.3V |
| LEDR[2] | PIN_V17 | LED [2] | 3.3V |
| LEDR[3] | PIN_V18 | LED [3] | 3.3V |
| LEDR[4] | PIN_W17 | LED [4] | 3.3V |
| LEDR[5] | PIN_W19 | LED [5] | 3.3V |
| LEDR[6] | PIN_Y19 | LED [6] | 3.3V |
| LEDR[7] | PIN_W20 | LED [7] | 3.3V |
| LEDR[8] | PIN_W21 | LED [8] | 3.3V |
| LEDR[9] | PIN_Y21 | LED [9] | 3.3V |

**Figure 8.4** DE1 SoC switch, LED, and push-button pin assignment descriptions

| Signal Name | FPGA Pin No. | Description | I/O Standard |
|---|---|---|---|
| GPIO_0[0] | PIN_AC18 | GPIO Connection 0[0] | 3.3V |
| GPIO_0 [1] | PIN_Y17 | GPIO Connection 0[1] | 3.3V |
| GPIO_0 [2] | PIN_AD17 | GPIO Connection 0[2] | 3.3V |
| GPIO_0 [3] | PIN_Y18 | GPIO Connection 0[3] | 3.3V |
| GPIO_0 [4] | PIN_AK16 | GPIO Connection 0[4] | 3.3V |
| GPIO_0 [5] | PIN_AK18 | GPIO Connection 0[5] | 3.3V |
| GPIO_0 [6] | PIN_AK19 | GPIO Connection 0[6] | 3.3V |
| GPIO_0 [7] | PIN_AJ19 | GPIO Connection 0[7] | 3.3V |
| GPIO_0 [8] | PIN_AJ17 | GPIO Connection 0[8] | 3.3V |
| GPIO_0 [9] | PIN_AJ16 | GPIO Connection 0[9] | 3.3V |
| GPIO_0 [10] | PIN_AH18 | GPIO Connection 0[10] | 3.3V |
| GPIO_0 [11] | PIN_AH17 | GPIO Connection 0[11] | 3.3V |
| GPIO_0 [12] | PIN_AG16 | GPIO Connection 0[12] | 3.3V |
| GPIO_0 [13] | PIN_AE16 | GPIO Connection 0[13] | 3.3V |
| GPIO_0 [14] | PIN_AF16 | GPIO Connection 0[14] | 3.3V |
| GPIO_0 [15] | PIN_AG17 | GPIO Connection 0[15] | 3.3V |
| GPIO_0 [16] | PIN_AA18 | GPIO Connection 0[16] | 3.3V |
| GPIO_0 [17] | PIN_AA19 | GPIO Connection 0[17] | 3.3V |
| GPIO_0 [18] | PIN_AE17 | GPIO Connection 0[18] | 3.3V |
| GPIO_0 [19] | PIN_AC20 | GPIO Connection 0[19] | 3.3V |
| GPIO_0 [20] | PIN_AH19 | GPIO Connection 0[20] | 3.3V |
| GPIO_0 [21] | PIN_AJ20 | GPIO Connection 0[21] | 3.3V |
| GPIO_0 [22] | PIN_AH20 | GPIO Connection 0[22] | 3.3V |
| GPIO_0 [23] | PIN_AK21 | GPIO Connection 0[23] | 3.3V |
| GPIO_0 [24] | PIN_AD19 | GPIO Connection 0[24] | 3.3V |
| GPIO_0 [25] | PIN_AD20 | GPIO Connection 0[25] | 3.3V |
| GPIO_0 [26] | PIN_AE18 | GPIO Connection 0[26] | 3.3V |
| GPIO_0 [27] | PIN_AE19 | GPIO Connection 0[27] | 3.3V |
| GPIO_0 [28] | PIN_AF20 | GPIO Connection 0[28] | 3.3V |
| GPIO_0 [29] | PIN_AF21 | GPIO Connection 0[29] | 3.3V |
| GPIO_0 [30] | PIN_AF19 | GPIO Connection 0[30] | 3.3V |
| GPIO_0 [31] | PIN_AG21 | GPIO Connection 0[31] | 3.3V |
| GPIO_0 [32] | PIN_AF18 | GPIO Connection 0[32] | 3.3V |
| GPIO_0 [33] | PIN_AG20 | GPIO Connection 0[33] | 3.3V |
| GPIO_0 [34] | PIN_AG18 | GPIO Connection 0[34] | 3.3V |
| GPIO_0 [35] | PIN_AJ21 | GPIO Connection 0[35] | 3.3V |

**Figure 8.5** DE1 SoC GPIO pin assignment descriptions

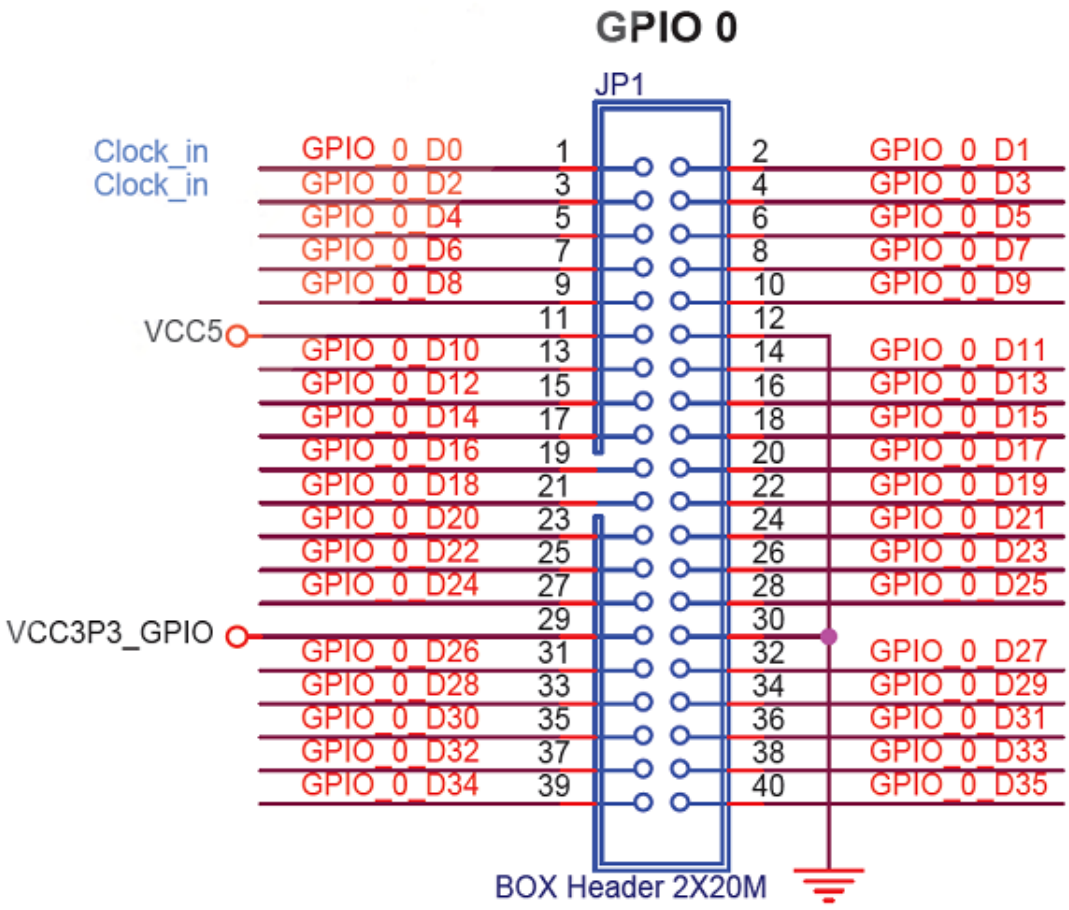


**Figure 8.6** DE1 SoC GPIO-0 pin diagram

## VI. Required IC List

| | |
|---|---|
| 74LS190 | BCD up/down counter |
| 74LS47 | 7-segment display driver |
| 74LS00 | 2-input NAND gates |
| 74LS74 | D Flip-Flops |
| - | 7-segment displays |

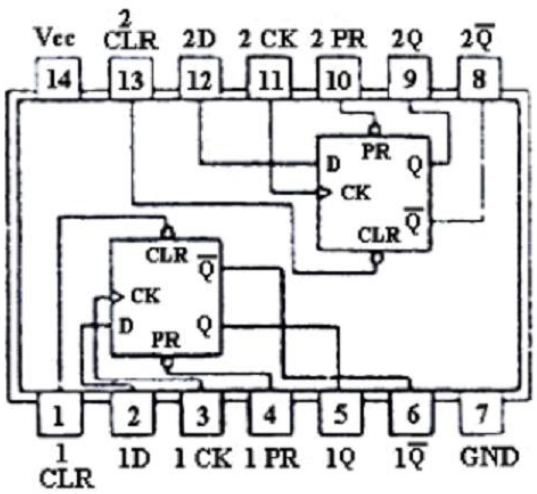


**Figure 8.7** Pin Diagram for 74LS74 IC

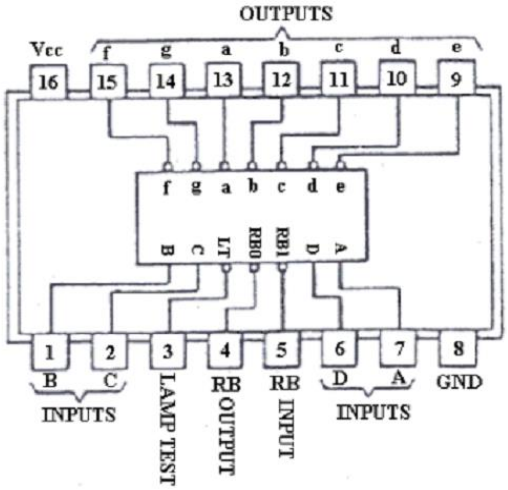


**Figure 8.8** Pin Diagram for 74LS47 IC

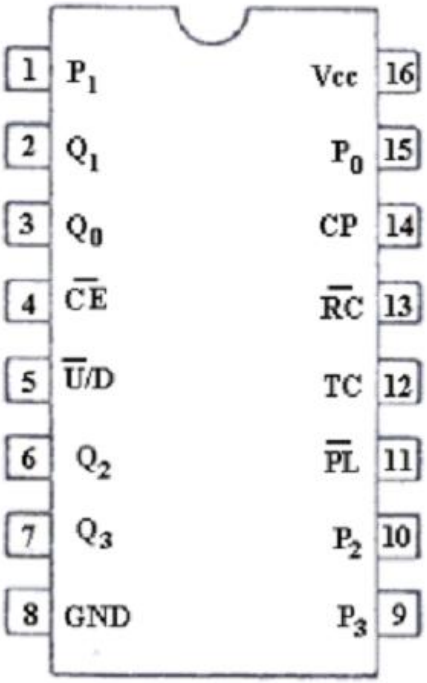


**Figure 8.9** Pin Diagram for 74LS190 IC

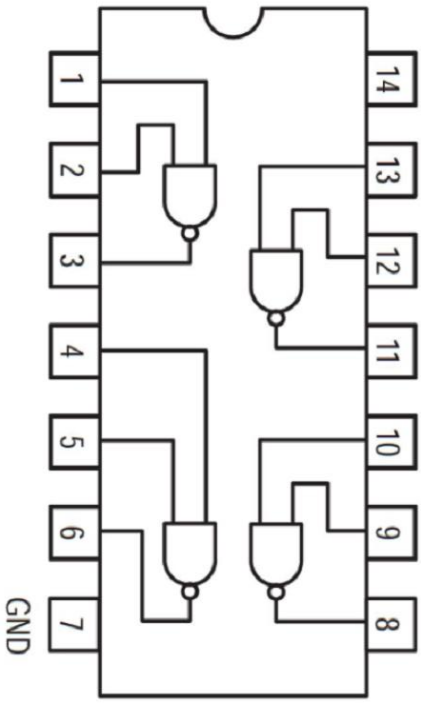


**Figure 8.10** Pin Diagram for 74LS00 IC

**Table 8.3** Pin names for 74LS190 IC

| | | | |
|---|---|---|---|
| **CE'** | Count Enable (Active LOW) i/p | **$P_n$** | Parallel Data i/p's |
| **CP** | Count Puls (Active HIGH going edge) i/p | **$Q_n$** | Flip/Flop o/p's |
| **U'/D** | Up/Down Count Control i/p | **RC'** | Ripple Clock o/p's |
| **PL'** | Parallel Load Control (Active LOW) i/p | **TC** | Terminal Count o/p |