Department of Computer and Software Engineering – ITU
SE201L: Digital Logic Design Lab

Course Instructor: **Ms Aqsa Khalid**	**Dated:** **16-11-2024**
Lab Engineer: **Muhammad Kashif**	**Semester:** **Fall 2024**
Batch: **BSSE 23 A**	

LAB 10 Verilog Implementation and Analysis of a Counter Circuit and Study of Counter IC Operation

Name	Roll Number	Lab Marks
Zunaira Abdul Aziz	BSSE23058	

Checked on: ____________________________

Signature: ____________________________

Verilog Implementation and Analysis of a Counter Circuit and Study of Counter IC Operation

10.1. Introduction

This lab consists of two parts: the first deals with the analysis of a 3-bit counter circuit; verification of its operation by implementation in Verilog and derivation of the state table and state diagram. In the second part, a counter circuit that works as a BCD counter has to be constructed using counter IC and additional circuitry.

10.2. Objectives

This lab will enable the students to achieve the following:

- Analyze a given sequential circuit and derive its state table and state diagram; verify its operation in Verilog
- Implement a sequential circuit in Verilog using flip-flops
- Understand how a BCD counter can be constructed using a binary counter and a few combinational logic gates

10.3. Conduct of Lab

1. Task 1 of this lab has to be performed using ModelSim and Task 2 using the logic trainer in Embedded Lab.
2. Bring printout of this lab manual when you come to perform the lab.
3. You can work and get evaluated in groups of two. However, manual submission has to be separate.
4. If there is difficulty in understanding any aspect of the lab, please seek help from the lab engineer or the TA.
5. If a lab task contains an instruction to show the work to lab engineer, make sure that the lab engineer evaluates and marks on your manual for that task. If your manual is unmarked for this task, it can result in mark deduction.
6. All tasks must be evaluated within the lab time. Print and attach screenshots of code and timing waveforms with your manual and submit it before the next lab.

10.4. Theory and Background

10.4.1. Counters

A counter is a special type of sequential circuit that goes through a prescribed sequence of states upon the application of a *clock* signal.

10.4.1.1. Types of counters

Counters have two basic categories based on the trigger mechanism of their flip-flops: *asynchronous* and *synchronous*. In asynchronous counters, the clock signal triggers only one of the flip-flop, whereas the consecutive flip-flops are triggered by the outputs (followed by some combinational circuitry in some cases) of the preceding flip-flops. In synchronous counters, the clock signal is connected to all flip-flops, thus causing all flip-flops to be triggered synchronously. Depending on the output sequence and count limit, there are numerous types of counters, such as up counter, down counter, Johnson counter, ring counter, decade counter, etc.

10.4.1.2. Sequential circuit analysis and design

Sequential circuit analysis is the process by which, for a given sequential circuit, its state table, state diagram and state equations are derived and its working is verified accordingly. Contrarily, *sequential circuit design* is the process by which, for a given problem statement, a state diagram, state table, state equations and a sequential circuit are obtained. Both of these techniques are discussed in detail in theory. Please refer to Chapter 5 of your textbook, *Digital Design by Morris Mano*.

10.5. Lab Tasks

10.5.1. Task 1: Analysis of a 3-bit synchronous counter [Marks: 10]

1. In this task, you are given the sequential circuit of Figure 10.1 to implement and analyze. This is a *synchronous 3-bit up-counter* with one input *En* and three output bits *A[2] – A[0]*, where *A[2]* is the MSB. Write Verilog module for this circuit using JK flip-flop module from last lab hierarchically inside your counter module. [4]

```
module JK_FF(J,K,clk,reset,Q);
    input wire J,
    input wire K,
    input wire clk,
    input wire reset,
    output reg Q
    always @(posedge clk or posedge reset) begin
        if (reset)
            Q <= 0;
        else begin
            case ({J, K})
                2'b00: Q <= Q;
                2'b01: Q <= 0;
                2'b10: Q <= 1;
                2'b11: Q <= ~Q;
            endcase
        end
    end
endmodule


module UpCounter(
    input wire En,
    input wire clk,
    input wire reset,
    output wire [2:0] A
);
    wire Q0, Q1, Q2;
    JK_FF FF0 (.J(En), .K(En), .clk(clk), .reset(reset), .Q(Q0));
    JK_FF FF1 (.J(Q0 & En), .K(Q0 & En), .clk(clk), .reset(reset), .Q(Q1));
    JK_FF FF2 (.J(Q1 & Q0 & En), .K(Q1 & Q0 & En), .clk(clk), .reset(reset), .Q(Q2));
    assign A = {Q2, Q1, Q0};
endmodule
```

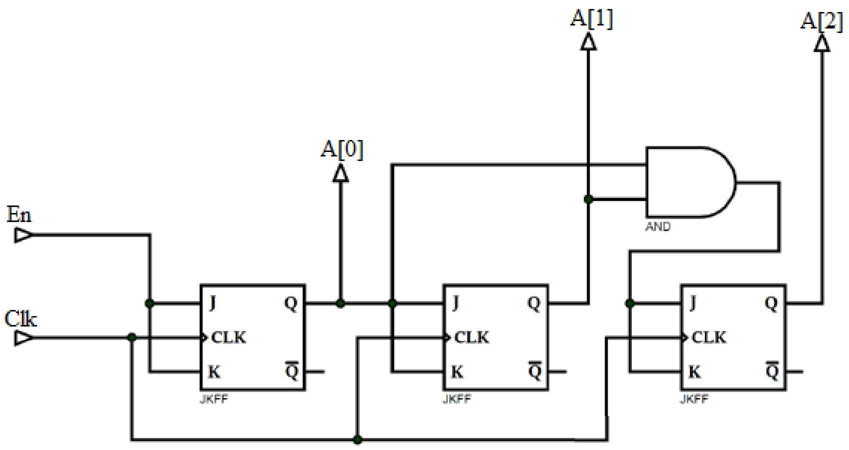


Figure 10.1: Synchronous 3-bit up-counter

2. Write a test bench to test your module. Run the simulation and fill the state table of this circuit in Table 10.1 according to the obtained simulation results. [3]

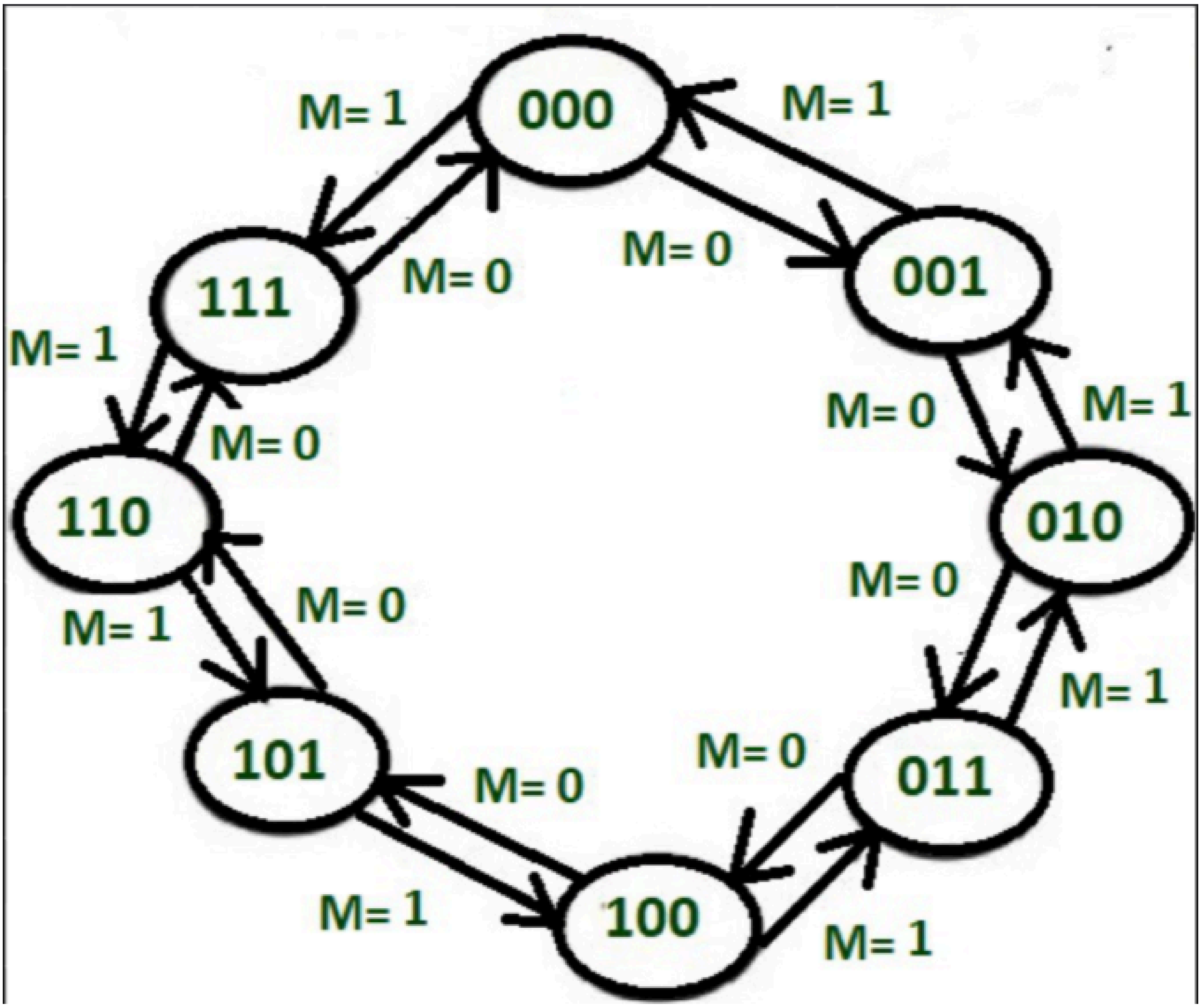
```
module UpCounter_tb();
  reg En, clk, reset;
  wire [2:0] A;
  UpCounter uut (.En(En), .clk(clk), .reset(reset), .A(A));
  initial begin
    clk = 0;
    forever #5 clk = ~clk;
  end
  initial begin
    En = 0; reset = 1;
    #10 reset = 0; En = 1;
  end
endmodule
```

Table 10.1: State table for the 3-bit up-counter

JK flip-flop present states			Input	JK flip-flop next states		
A[2](t)	**A[1](t)**	**A[0](t)**	**En**	**A[2](t+1)**	**A[1](t+1)**	**A[0](t+1)**
0	0	0	0	**0**	**0**	**1**
0	0	0	1	**0**	**1**	**0**
0	0	1	0	**0**	**1**	**1**
0	0	1	1	**1**	**0**	**0**
0	1	0	0	**1**	**0**	**1**
0	1	0	1	**1**	**1**	**0**
0	1	1	0	**1**	**1**	**1**
0	1	1	1	**0**	**0**	**0**
1	0	0	0	**1**	**1**	**1**
1	0	0	1	**0**	**0**	**0**
1	0	1	0	**0**	**0**	**1**
1	0	1	1	**0**	**1**	**0**
1	1	0	0	**0**	**1**	**1**

1	1	0	1	**1**	**0**	**0**
1	1	1	0	**1**	**0**	**1**
1	1	1	1	**0**	**1**	**0**

3. Draw the state diagram of this circuit according to the state table in the space below: [3]



4. Give answers to questions 1 and 2 of Analysis.
5. Take clear screenshots of the code of all modules of this task and the simulation timing waveforms, and paste them in a WORD file such that all fit on no more than two pages. Make sure that the screenshots are legible.

10.5.2. Task 2: BCD counter using 4-bit binary counter [Marks: 10]

1. A BCD counter can be made using a general 4-bit binary counter with an active-high "load" input when used in the following configuration with an AND gate:

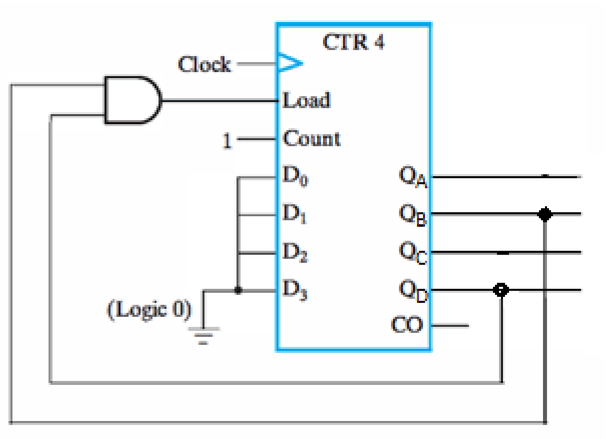


Figure 10.2: BCD counter using 4-bit binary counter

D_0, D_1, D_2, D_3 are counter initial state inputs. (D3 is MSB)
Q_A, Q_B, Q_C, Q_D are counter outputs. (Q_D is MSB)
CO is the carry-out of the counter.

2. In this task, you will implement this BCD counter using the 74193 counter IC, which is a synchronous 4-bit up/down counter. Its pin configuration and pin description are given below:

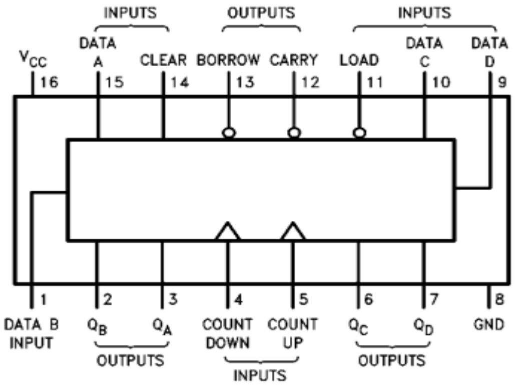


Figure 10.3: Pin configuration of 74193

Q_A, Q_B, Q_C, Q_D: Counter outputs (Q_D is MSB)
Data A, B, C, D: Counter initial state inputs (D is MSB)
Count Up: Clock signal is applied to this pin to make the counter count up
Count Down: Clock signal is applied to this pin to make the counter count down
Clear: Asynchronously stores 0 in all bits of the counter
Load (Active low): Loads initial state of the counter according to data inputs A-D
Carry (Active low): Indicates overflow in case of up-count
Borrow (Active low): Indicates underflow in case of down-count

3. Build the circuit of Figure 10.2 using the counter IC and NAND gate IC (in place of AND as shown in figure) on breadboard. [4]
4. Give answer to question 3 of Analysis.
5. Operate 74193 in *up-count mode* and verify the operation of the BCD counter. You are required to display the outputs QA, QB, QC and QD of 74193 on state monitor LEDs in the correct order and also on any one of the 7-segment displays of the logic trainer. Fill in the following table after each clock cycle: [2]

Table 10.2: BCD counter output

Clock cycle	Decimal value displayed	QD	QC	QB	QA
1st	0	0	0	0	0

2nd	1	0	0	0	1
3rd	2	0	0	1	0
4th	3	0	0	1	1
5th	4	0	1	0	0
6th	5	0	1	0	1
7th	6	0	1	1	0
8th	7	0	1	1	1
9th	8	1	0	0	0
10th	9	1	0	0	1

6. Show working circuit to the lab engineer to obtain credit. [4]

10.5.3. Analysis [Marks: 5]

1. How many states does this circuit have? [1]

7. Which type of state machine does your diagram implement, Mealy or Moore? Give reason. [2]

8. Why do you think we have used a NAND IC in place of AND? [2]

Assessment Rubric

Method:
Lab report evaluation and instructor observation during lab sessions.

Performance	CLO	Able to complete the tasks over 80% (4-5)	Able to complete the tasks 50 – 80% (2-3)	Tasks completion below 50% (0-1)	Marks
1. Teamwork	1	Actively engages and cooperates with other group members in an effective manner	Cooperates with other group members in a reasonable manner	Distracts or discourages other group members from conducting the experiments	
2. Laboratory safety and disciplinary rules	1	Observes lab safety rules; handles the development board and other components with care and adheres to the lab disciplinary guidelines aptly	Observes safety rules and disciplinary guidelines with minor deviations	Disregards lab safety and disciplinary rules	
3. Realization of experiment	3	Conceptually understands the topic under study and develops the experimental setup accordingly	Needs guidance to understand the purpose of the experiment and to develop the required setup	Incapable of understanding the purpose of the experiment and consequently fails to develop the required setup	
4. Conducting experiment	3	Sets up hardware/software properly according to the requirement of experiment and examines the output carefully	Makes minor errors in hardware/software setup and observation of output	Unable to set up experimental setup, and perform the procedure of experiment	
5. Data collection	3	Completes data collection from the experiment setup by giving proper inputs and observing the outputs, complies with the instructions regarding data entry in manual	Completes data collection with minor errors and enters data in lab report with slight deviations from provided guidelines	Fails at collecting data by giving proper inputs and observing output states of experiment setup, unable to fill the lab report properly	
6. Data analysis	3	Analyzes the data obtained from experiment thoroughly and accurately verifies it with theoretical understanding, accounts for any discrepancy in data from theory with sound explanation, where asked	Analyzes data with minor error and correlates it with theoretical values reasonably. Attempts to account for any discrepancy in data from theory	Unable to establish the relationship between practical and theoretical values and lacks the theoretical understanding to explain any discrepancy in data	
7. Computer use	3	Successfully uses lab PC and internet to look for relevant datasheets, carry out calculations, or verify results using simulation	Requires assistance in looking for IC datasheets and carrying out calculation and simulation tasks	Does not know how to use computer to look up datasheets or carry out calculation and simulation tasks	
				Total (out of 35)	