

Experiment 5 Sequential Logic Circuits

Due date : 14.04.2021 23:00

Abdullah Ekrem Okur

okurabd@itu.edu.tr

1 Introduction

In this experiment, you will implement sequential logic circuits by using the Verilog. Refreshing your knowledge on followings is useful to implement this experiment;

- How to analyze a given sequential circuits,
- How to design circuits with Mealy and Moore models,

You can utilize 7th and 8th slides of BLG231E-Digital Circuits course for this experiment. Also, you can use the modules that you have implemented in previous experiments.

2 Preliminary

Please complete the preliminary work and explain their in detail in your report.

- Analyze the circuit in Part-1 to create a state transition table and a state chart.
- Design and draw the circuit to implement in Part-2 with the minimum number of logic gates.

3 Experiment

Part 1

In this part, you have to implement the circuits in Figure 1 by using Verilog. You need to use **different type flip flop modules**, **logic gates** and **clock signal** to synchronize the circuit. Report the state transition table for this circuits and draw state diagram graphically. Also, show your simulation results (timing diagram) by testing your circuits. You may add reset input to set initial values.

NOTE: You cannot use always block.

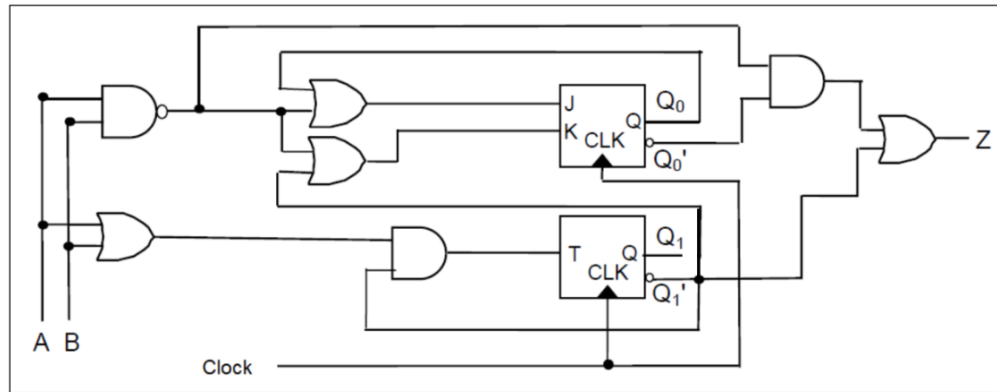


Figure 1: Sequential circuit with flip flops

Part 2

Build a 4 bit counter circuit that counts 0 to 14 in a circular way (in an other word, after 14, it returns to 0). Block diagram of the counter is given in Figure 2. Here, input X is the counting direction. X=0 means counting DOWN (2-1-0-14-13-12-...) while X=1 means counting UP (13-14-0-1-2...). You have to use D or JK flip flop modules instead of reg type in this part.

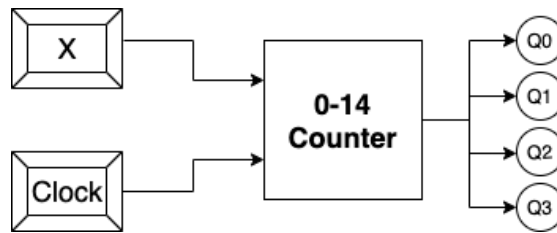


Figure 2: 4 bit counter block diagram.

Part 3

In this part, please implement a 16-bit up-down counter that have 16-bit input data, load, clock, direction, 3-bit increment/decrement value and clear inputs using **reg** type parameter and **always block**. The output of the module is 16-bit current value of the counter. When the clear input is zero, current value of the counter becomes zero. When the load input is zero, input data is loaded to the the current value register. Otherwise, the current value of the counter is increased or decreased according to direction input and increment/decrement value on each positive edge of the clock input. For example, when the direction input is one and increment/decrement value is 2 (and clear=1, load=1), the module increases the counter with number 2. When the direction input is zero and decrement number is 3, the module decreases the counter with 3.

Part 4

In this part, implement below counter modules using 16-bit up-down counter module that is implemented in Part 3;

- A counter that counts up from 0 to 40 in circular way with increment value 2.
- A counter that counts up from 350 to 371 in circular way with increment value 3
- A counter that counts down from 93 to 5 in circular way with decrement value 4.
- A counter that counts up from 22525 to 22535 in circular way with increment value 1.

Each module has clock input, initiate input, and 16-bit current value output. When the initiate input is zero, your counter will be set any number which is in the range of the counter. For example, for the 0 to 20 counter, you can set the current value as 0 when the initiate input is zero. The clock input will be used for update the current value.

4 Report

- You should show your work of Preliminary study on the report in detail.
- You can use any software tool for your circuit designs and other figures. You may attach them to the report as figures by properly referencing them in the text. Also please do not put any handwritten note in the report.
- Please use the table attributes of Latex. You can check out online Latex table generators (for example: <https://www.tablesgenerator.com>).
- Your report should contain information about the results of your simulations. If your implementations are not fully correct, discuss what the source of the errors might be in your report.
- You must cite your source used in the report. You can not directly copy and paste any sentences from any source even if you cite this document. You should read the document, write this information in your own words, then you must cite the resource.
- For further details about the report, please check Ninova e-learning system.

5 Submission Policy

- You should upload your experiment codes and report on Ninova, and please, do not send your experiment files via e-mail.

Experiment 5 Sequential Logic Circuits

- Please submit 2 Verilog Design files for the experiment codes. One of them contains modules' codes, the other one consists of simulation codes.
- Please add comments to the code to clarify your intends.
- Your reports must be written with Latex format. Latex report template is available on Ninova. You can use any Latex editor whichever you want. If you upload your report without Latex file, you directly get 0 as your report grade. You should upload both .pdf and .tex files of your report.
- Please do not forget that late submissions are not accepted.
- Please make sure that you have written your full name and student ID number to every document you are submitting.
- Please only write the names of members who have worked on the experiment.
- Please do not hesitate to contact me (okurabd@itu.edu.tr) for any question.