Hacettepe University Computer Engineering Department

Experiment 5 - Sequential Circuits in Verilog

Due date: Thursday, 23/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

BBM233 Digital Design Lab - 2021 Fall

AIM

Designing and simulating sequential circuits in Verilog HDL.

BACKGROUND

Sequential Circuits

Sequential circuits are circuits whose outputs depend on both the present inputs and the sequence of past inputs (sequential circuits include memory elements). That is, the previous inputs or state of the circuit will have an effect on its present state.

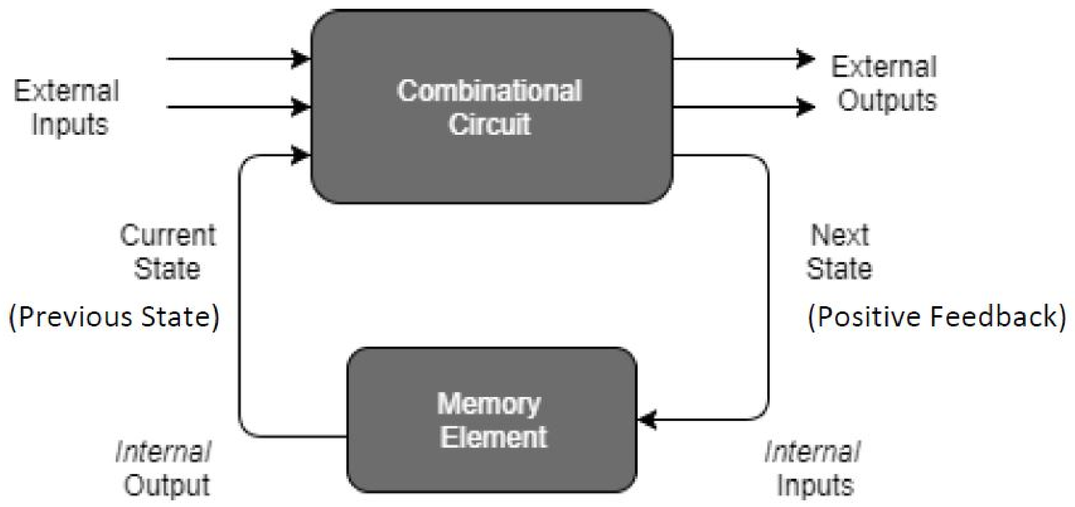
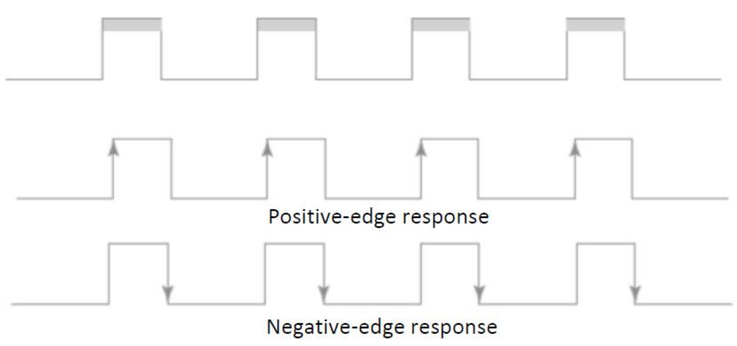


Figure: Sequential Circuit

Storage Elements

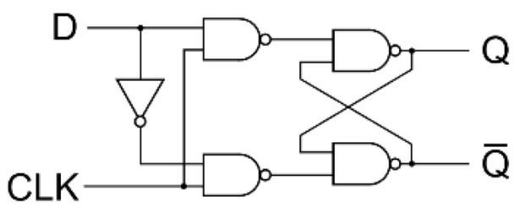
Latches: level-sensitive

Flip-flops: edge-sensitive



A flip flop is a basic building block of sequential logic circuits. It is a circuit that has two stable states and can store one bit of state information. The output changes state by signals applied to one or more control inputs.

Edge-Triggered D Flip Flop



Inputs	Present Output	Next Output
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

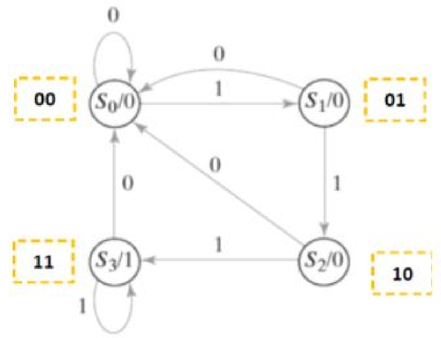
Characteristic Equation
Q(t+1) = D

A basic D Flip Flop has a D (data) input, a clock (CLK) input and outputs Q and Q' (the inverse of Q). Optionally it may also include the PR (Preset) and CLR (Clear) control inputs.

Sequential Circuit Design Steps

To design a sequential circuit, start with the problem definition and use the following steps:

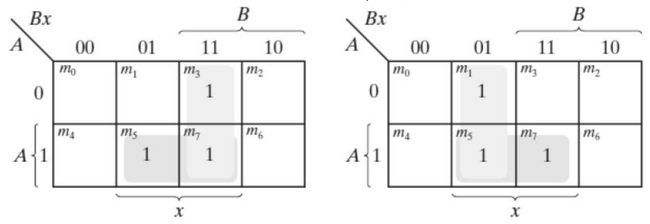
1 Create a state transition diagram from the description. Reduce the number of states if necessary, and assign binary values to the states.



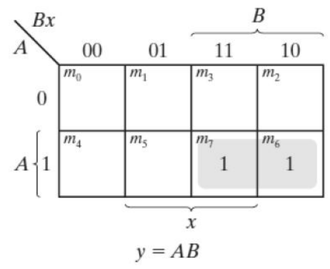
2 Convert the state transition diagram into a state transition table (binary coded state table).

Present State		Input	Next State		Output
A	*B*	*x*	*A*	*B*	*y*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

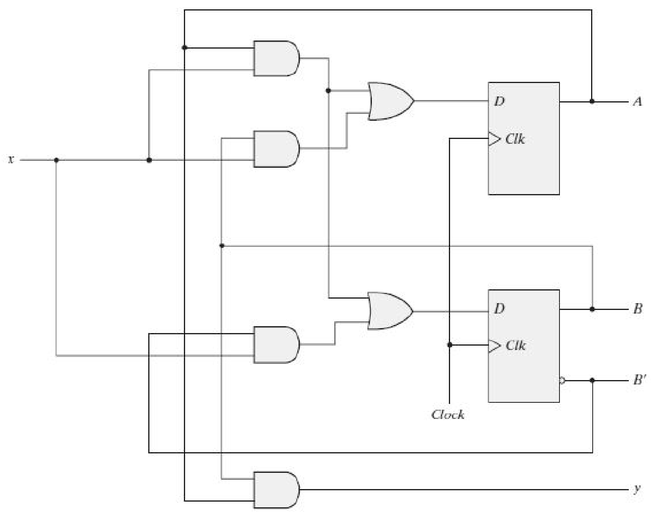
3 Determine the number of flip-flops needed and choose flip-flop types. Derive their excitation tables (if designing a sequential circuit with flip-flops other than the D type), and derive input and output equations from the state table. Minimize the functions for the flip-flop inputs (e.g. using Karnaugh Maps).



4 Determine the combinatorial circuit to represent the output (if any).



5 Finally, use simplified functions to design sequential circuit and obtain the design schematic.



Hacettepe University Computer Engineering Department

Experiment 5 - Sequential Circuits in Verilog

Due date: Thursday, 23/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

BBM233 Digital Design Lab - 2021 Fall

Introduction

In the movie, Bender's Big Score, a level 87 binary code is discovered. The code is used for paradox-free time travel but at the cost of endangering the very fabric of reality.

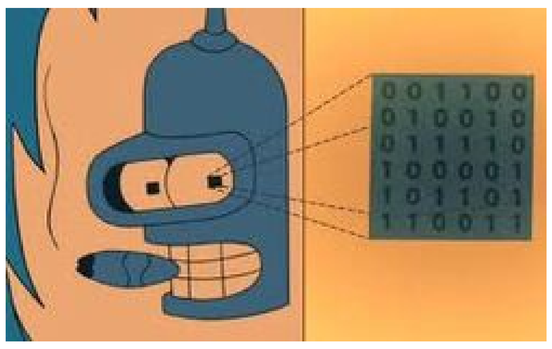


Figure 1: The time code

Following the discovery of this binary code, a spark of curiosity appeared inside Professor Farnsworth. He wanted to know if he could achieve supernatural effects by using lower-level binary codes.

Figure 2: Professor Farnsworth in his Chamber of Understanding

To follow the idea that came to him in a dream, the Professor decided to design a finite state machine that would be used to test binary sequences generated by a generator module installed to Bender, which has a chance to trigger a lower level supernatural effect. After spending some enjoying minutes in his Chamber of Understanding, Professor Farnsworth drew the diagram for his magnificent machine:

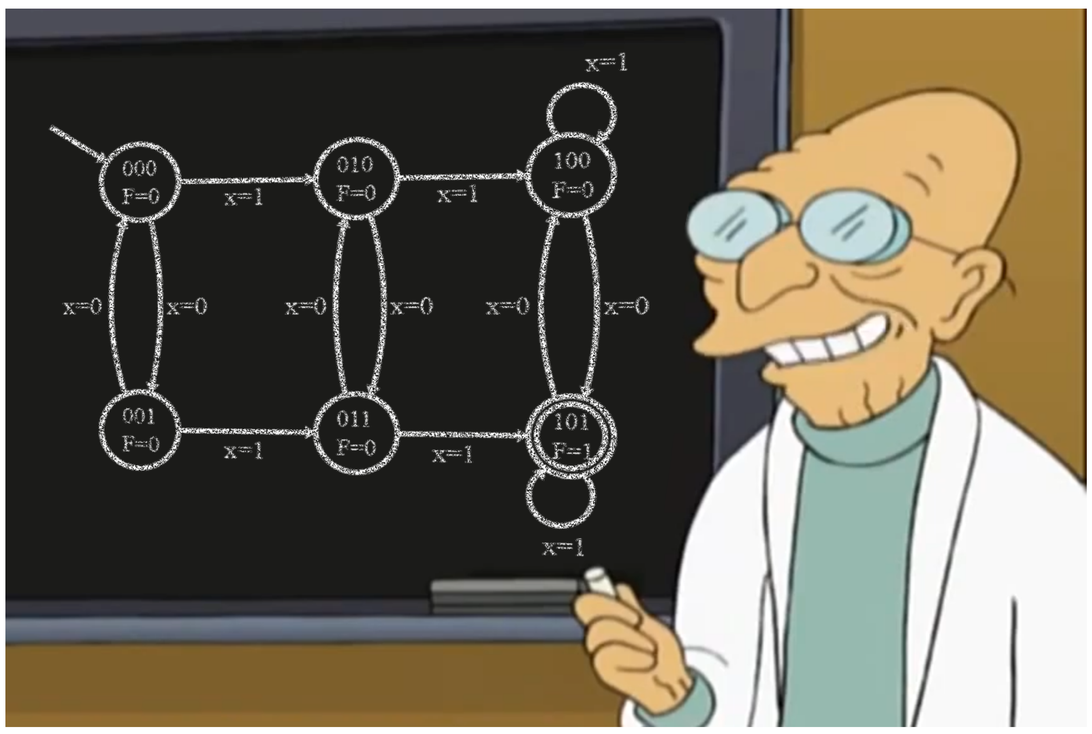


Figure 3: Diagram of the machine which recognizes at least two 1's and an odd number of 0's

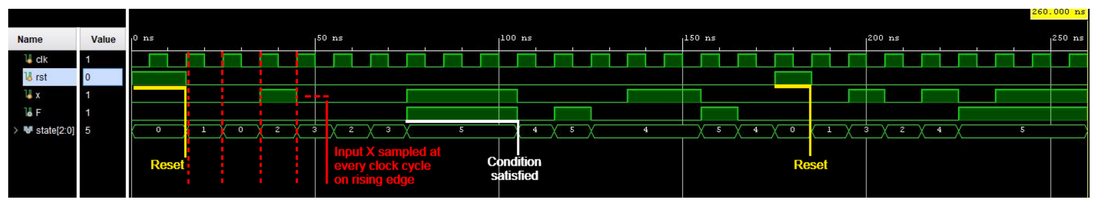
System Specifications

The state diagram of the machine is illustrated in Figure 3. It has six states, one 1-bit input x, and one 1-bit output F.

You should follow the instructions below:

- Use sequential circuit design steps described on the previous page to obtain the design schematic. Use D flip flops as storage elements to store the state information.
- Implement the design in Verilog using **STRUCTURAL design approach following the circuit schematic. Behavioral design will be graded as 0.** Name your D flip flop module as **dff.v** and machine module as **machine_d.v**
- You must use a Rising Edge D Flip-Flop with Asynchronous Reset on High Level. I.e., it should be triggered on the rising edges of the both **clk** and the **rst** signals, as can be seen in the waveform below.
- **You MUST download and use these starter code files before you start working! Do NOT change the I/O port names!**
- Verify the Verilog model of the machine by writing an appropriate testbench **machine_d_tb.v** for all possible test cases.

Make sure to obtain a similar waveform which shows the correctness of your design. And explain the obtained results in your report. **You MUST test for all possible state transitions.**



System Specifications [BONUS for Extra Credit]

Using the diagram illustrated in Figure 3, follow the instructions given in the first part but use **JK flip flops** instead of D flip flops and name your Verilog source code files as **jkff.v**, **machine_jk.v** and **machine_jk_tb.v** (you MUST use starter code!). In this implementation too you should use a positive-edge-triggered JK Flip-Flop with Asynchronous Reset on High Level. I.e., it should be triggered on the rising edges of the both **clk** and the **rst** signals.

For full credit, you must include all design steps as for the mandatory part, especially **the excitation tables for JK flip flops** to show your work.

Hacettepe University Computer Engineering Department

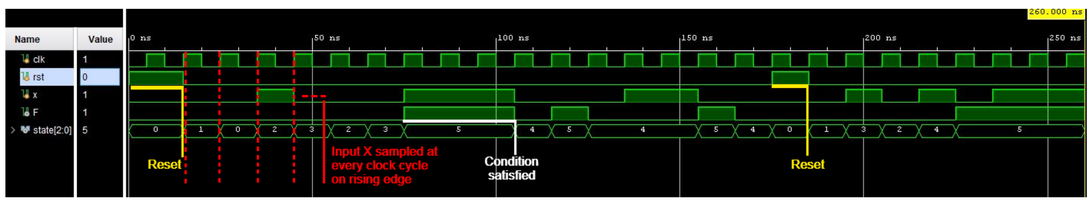
Experiment 5 - Sequential Circuits in Verilog

Due date: Thursday, 23/12/2021, 22:00:00 **Via: https://submit.cs.hacettepe.edu.tr/**

BBM233 Digital Design Lab - 2021 Fall

System Specifications [BONUS for Extra Credit]

Make sure to obtain a similar waveform which shows the correctness of your design. And explain the obtained results in your report. **You MUST test for all possible state transitions to receive extra points.**



PLAGIARISM CONTROL NOTICE

Students must implement their solutions individually. All submissions will be submitted to a plagiarism check. Any submissions that show a high level of similarity will be reported as plagiarism attempts to the ethics committee.

WHAT TO INCLUDE IN THE REPORT

You are encouraged to use this Verilog Assignment Report Template and create your reports in LaTeX. We suggest using Overleaf platform for this. This is not mandatory, but make sure your report has all necessary parts and information. Check this file to see how the report should be structured: Verilog Assignment Report Template PDF

Your report needs to include the following:

1. Include the problem statement.
2. Convert the given state transition diagram to state transition table.
3. Determine the number of D flip flops you need to use to store the state information, and derive the input and output equations from the state transition table. Minimize the functions using K-Maps.
4. Use the simplified functions to obtain the **design schematic** that uses D flip flops.
5. Include three Verilog code solutions for all specified modules: **dff.v**, **machine_d.v**, and **machine_d_tb.v**
6. Include the obtained waveform.
7. Explain the obtained results and how they show that your design is working correctly.
8. You may use any resources, online or otherwise, but make sure to include the references in your report.

Submission via submit.cs

Submissions will be accepted via https://submit.cs.hacettepe.edu.tr/. Note that the deadline for submission of the report and all codes is 23/12/2021 at 22:00:00.

Your submission will include the Verilog codes and a report PDF and it must be in the following format to be accepted:

- b<studentID>.zip
  - dff.v
  - machine_d.v
  - machine_d_tb.v
  - report.pdf

Note that only ZIP archives are accepted!

Automatic grading [VERY IMPORTANT!]

Your submissions will be graded using an automatic grading script. They **will not be** white-box tested. So, you are expected to **match the given wave forms 100%**, to receive full-credit.

You MUST download and use these starter code files before you start working! Do NOT change the I/O ports (names, order, bit-width, etc.)!

Grading

Report: 20%
Verilog codes: 80%
Bonus: 10%