

# Verilog Practice Part 3 - Sequential Circuits

Weeks 11-12 - **UNGRADED**

BBM233 Digital Design Lab - 2024 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.

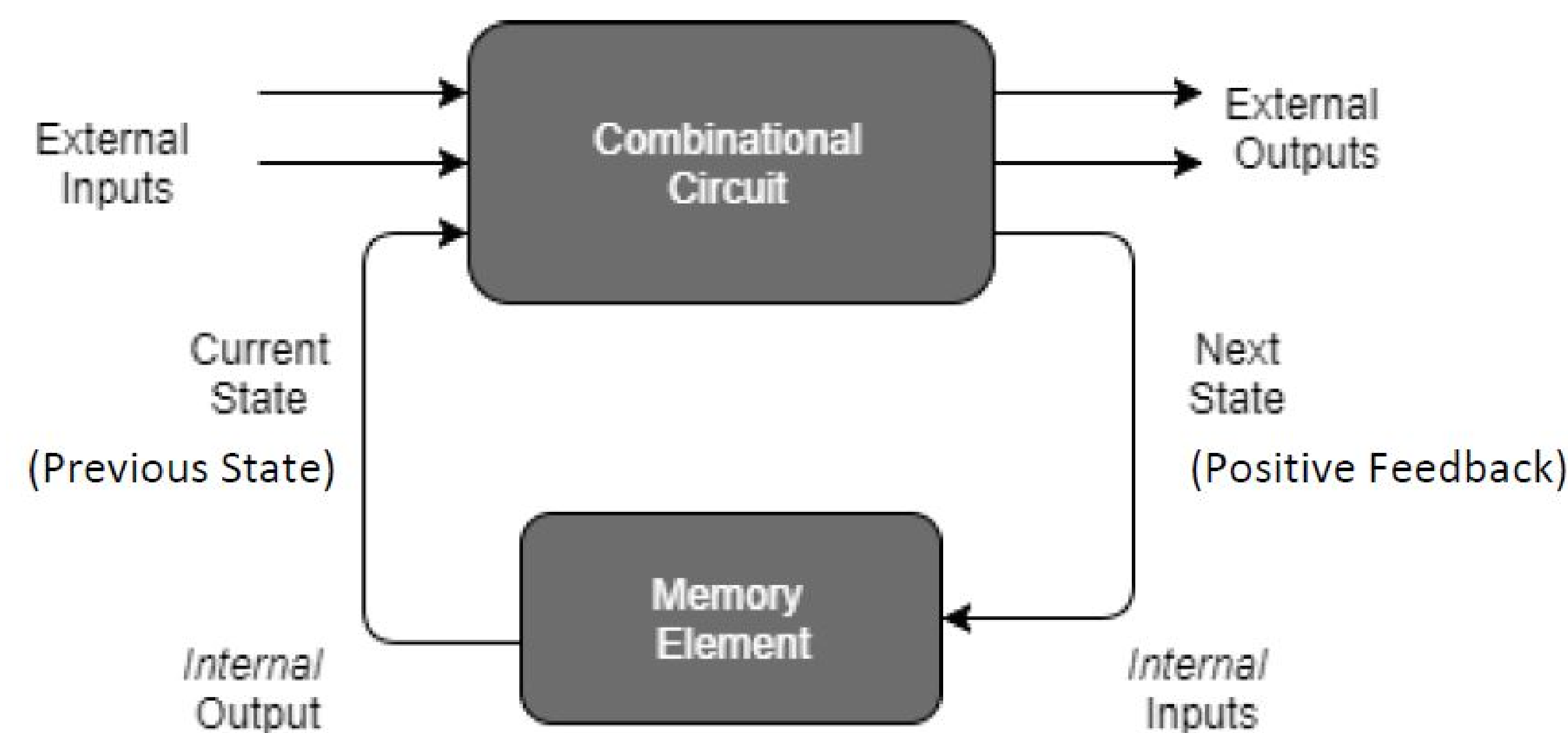


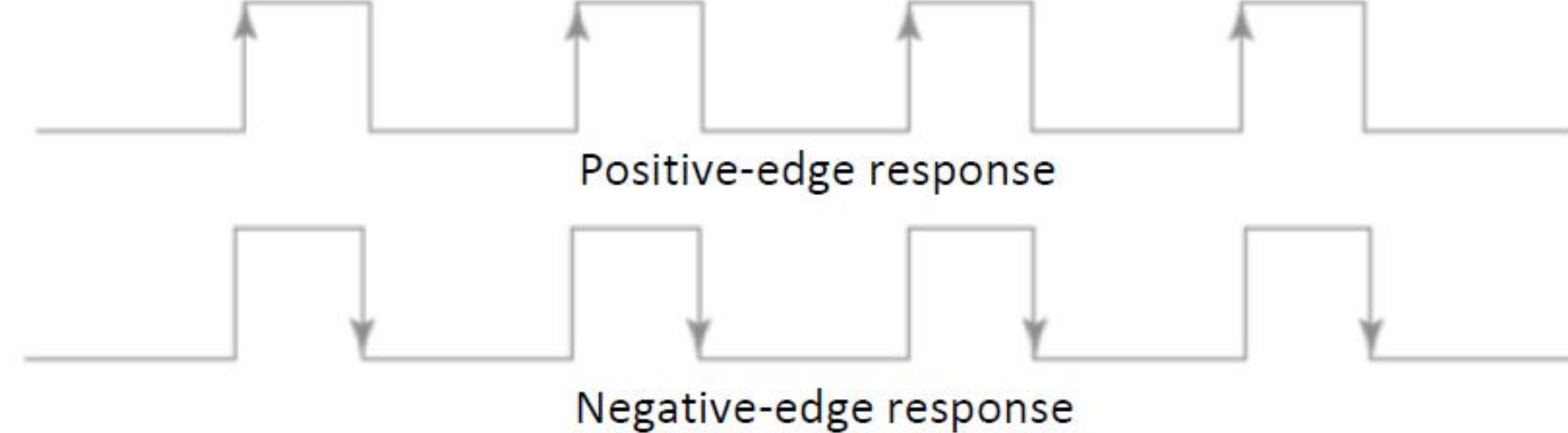
Figure: Sequential Circuit

### Storage Elements

Latches: level-sensitive

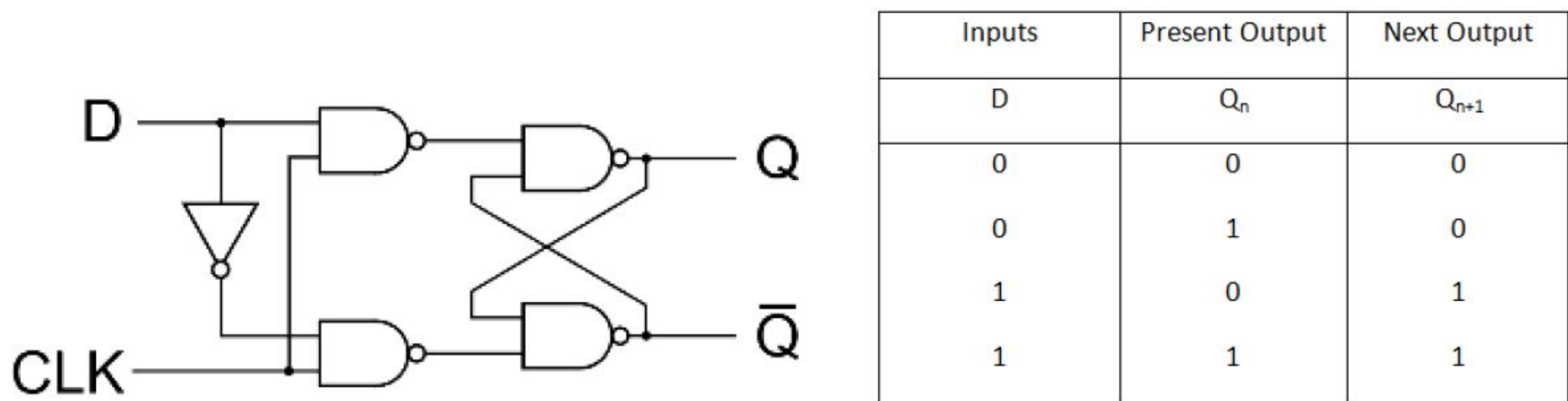


Flip-flops: edge-sensitive



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



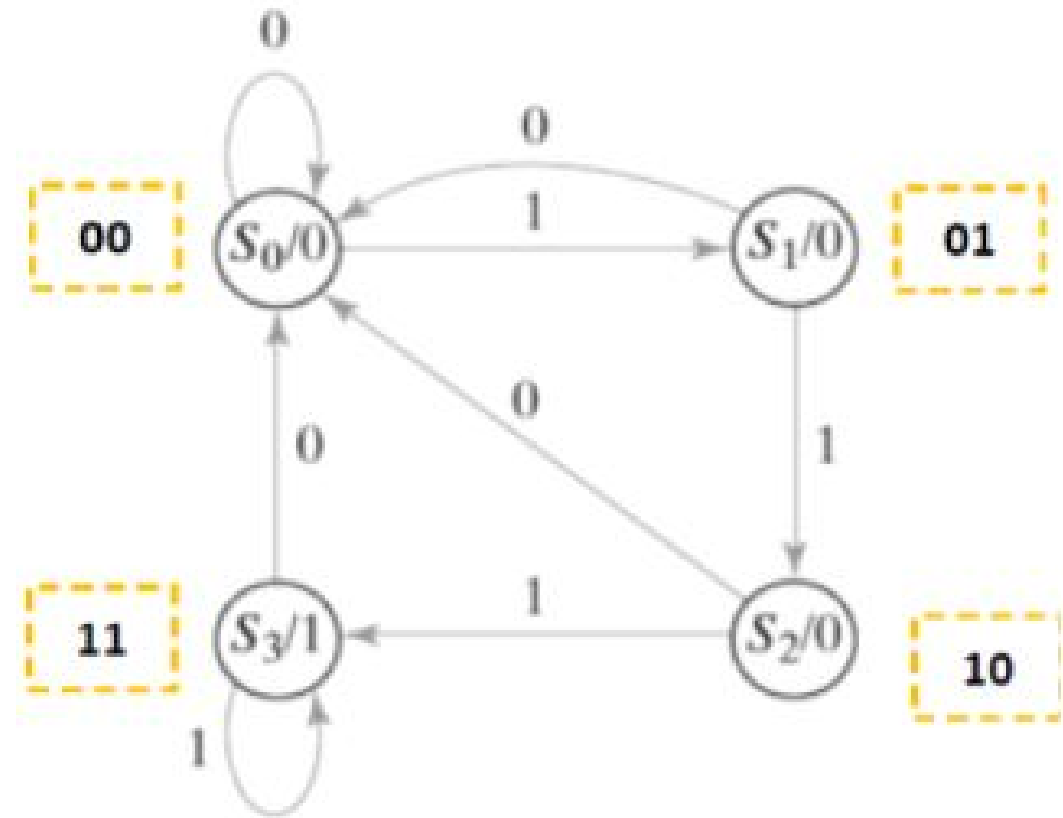
Characteristic Equation  
 $Q(t+1) = D$

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:

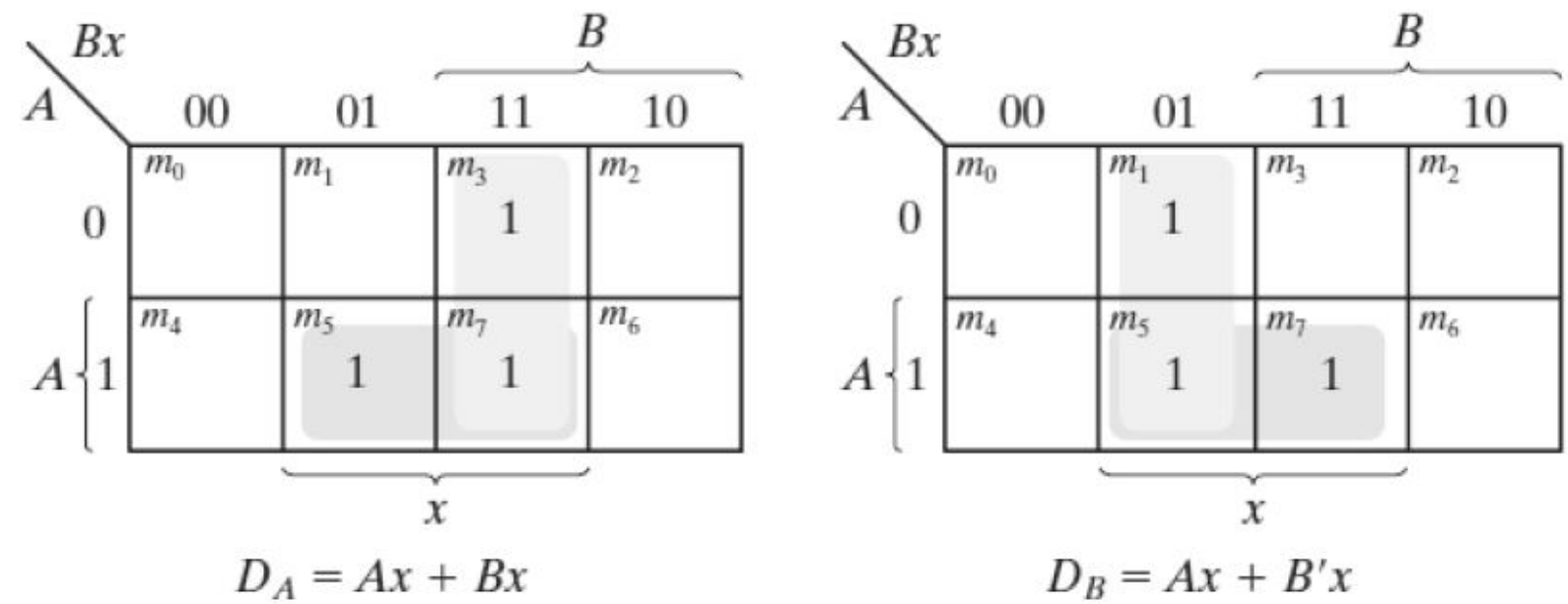
- 1 Create a state transition diagram from the description. Reduce the number of states if necessary, and assign binary values to the states.



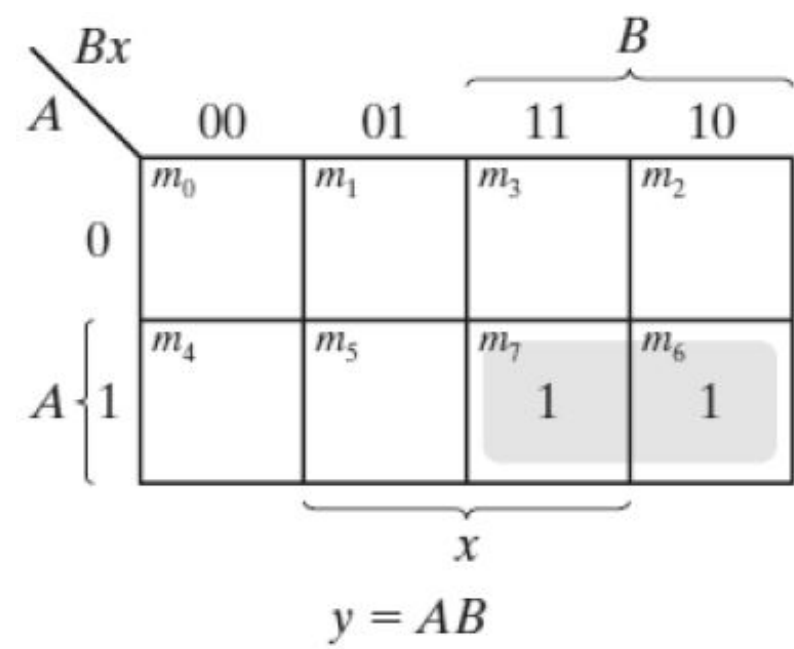
- 2 Convert the state transition diagram into a state transition table (binary coded state table).

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

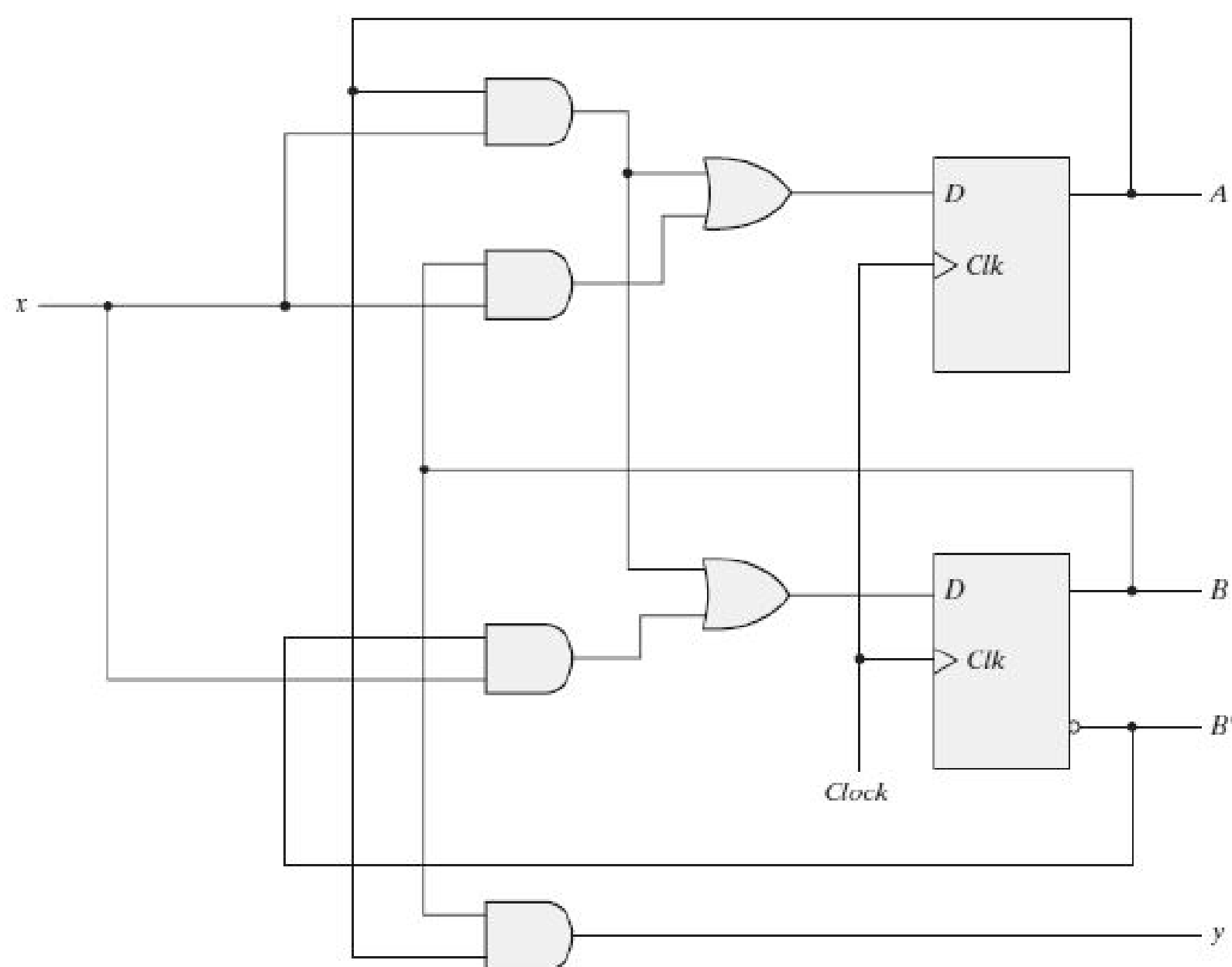
- 3 Determine the number of flip-flops needed and choose flip-flop types. Derive their excitation tables (if designing a sequential circuit with flip-flops other than the D type), and derive input and output equations from the state table. Minimize the functions for the flip-flop inputs (e.g. using Karnaugh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



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



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





# Verilog Practice Part 3 - Sequential Circuits

# Weeks 11-12 - UNGRADED

BBM233 Digital Design Lab - 2024 Fall

# Introduction

In the movie ‘Bender’s Big Score,’ an intricate level 87 binary code is discovered, possessing the unique ability to enable paradox-free time travel. However, its use comes with a significant risk — the potential to jeopardize the very fabric of reality.

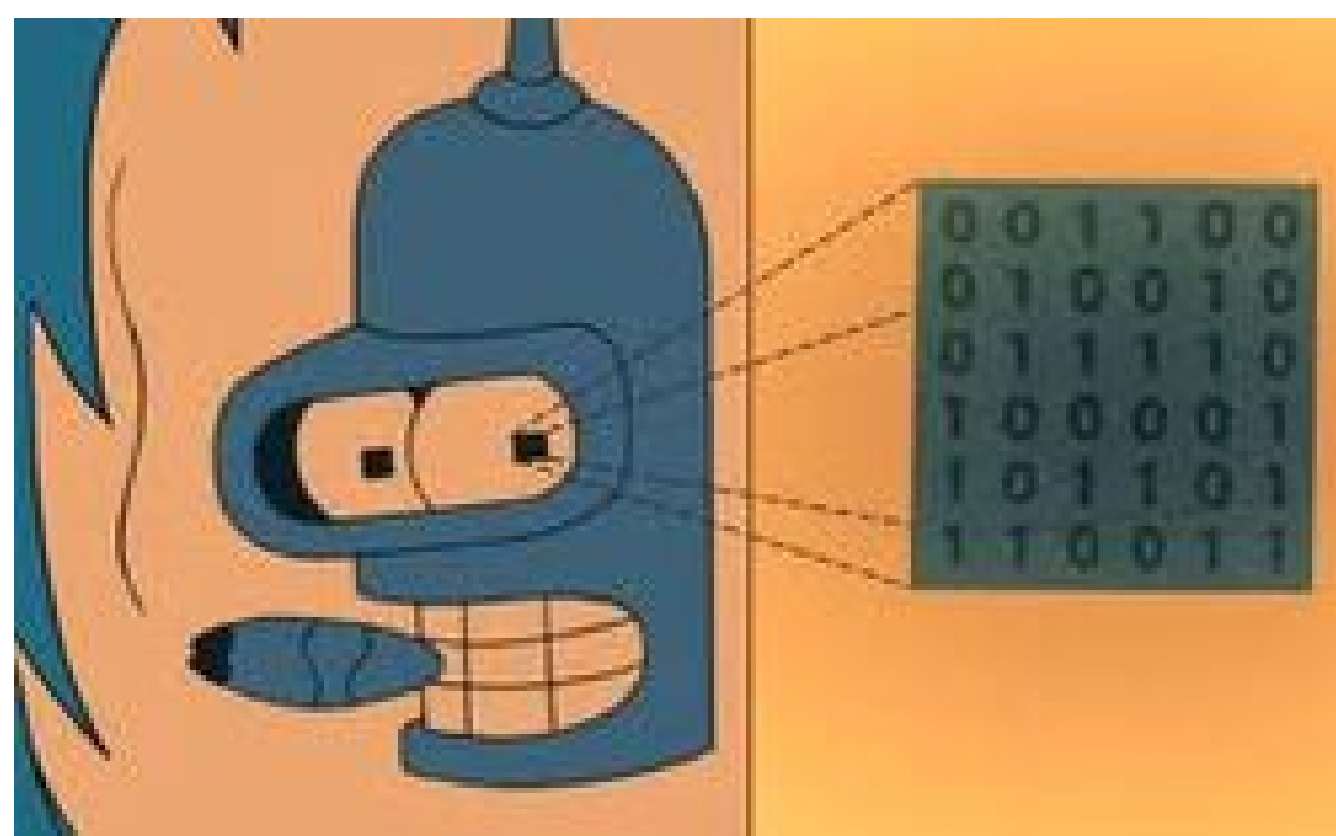


Figure 1: The time code

Sparked by the discovery of this binary code, Professor Farnsworth's curiosity is ignited. He becomes intrigued by the possibility of harnessing lower-level binary codes to achieve supernatural phenomena.



Figure 2: Professor Farnsworth in his Chamber of Understanding

Motivated by an insightful dream, Professor Farnsworth sets out to create a finite state machine that is intended to test various binary sequences produced by a generator module fitted to Bender, potentially triggering supernatural effects at a lower level. Immersed in his Chamber of Understanding, the Professor dedicates several engaging minutes to conceive the blueprint of this extraordinary machine:

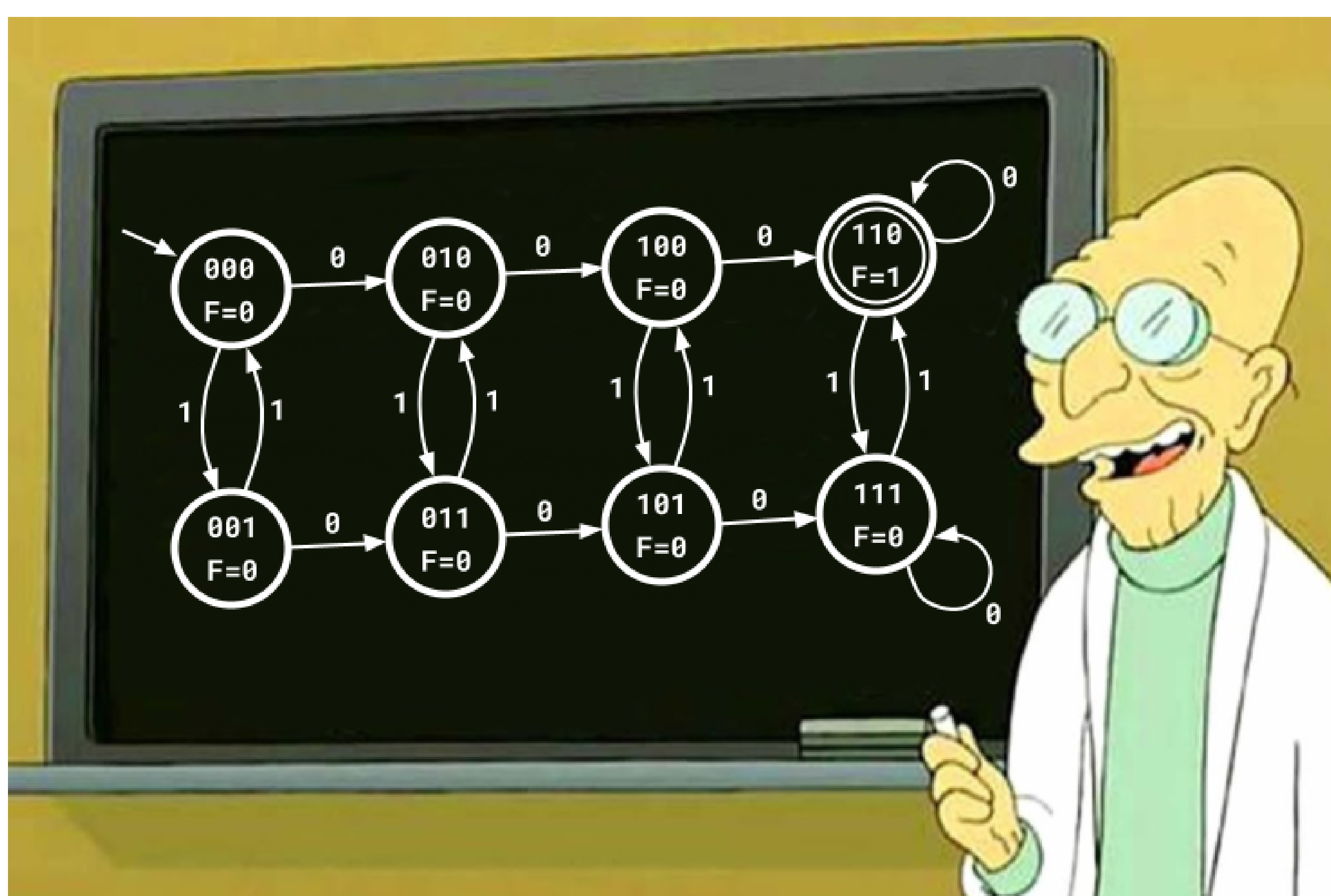


Figure 3: Diagram of the machine that recognizes at least three 0's and an even number of 1's.

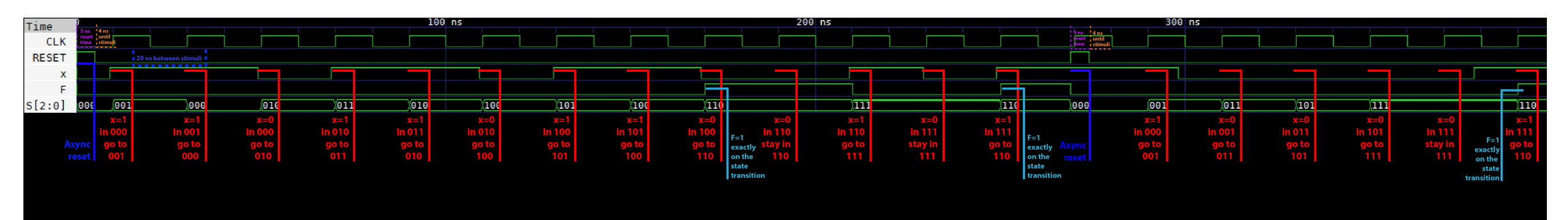
## System Specifications

The state diagram of the machine is illustrated in Figure 3. It has eight 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. **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.