Digital Design Lab
EEN 315 Section 3G

Lab 2
Three-bit Loadable Up-Down Synchronous Counter Attached to a 7-Segment Display and a Sequence Detector

Group 5
Nathan Paternoster
(Partner: Michael Finale, Daniel Bennett)

Sayan Maity, TA

University of Miami

10/13/13

Abstract

This lab aims to demonstrate the implementation of Medium Scale Integration (MSI) and structured design through designing a three-bit counter. This counter will have two capabilities: to be able to count up or down in the range of 0-7 and to be loaded with an initial number. The up/down input (U) will be '0' when the counter counts down and '1' when it counts up. The load input (Load) will be '0' when the counter is counting and '1' when inputting a load value. The load value will be input with three bits of input called I_0, I_1, and I_2. In total there will be five inputs. There will be three outputs X_0, X_1, and X_2 to represent the numbers being counted.

When implementing the combinational logic of this circuit we will use a special type of MSI chip called a multiplexer (MUX). The multiplexer is a *data selector* and will select any **one** of its inputs to be the output at any given time. This will reduce circuit complexity and the amount of integrated circuits required.

Table of Contents

Overview

The counter will contain two main components. The first will control the actual counting and whether or not the counter is counting up or down. The second component will control the loading inputs. They will be connected like so:

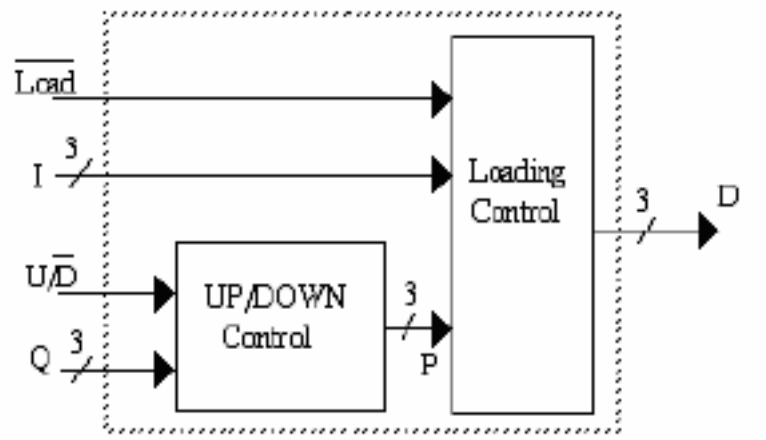


Figure 2
Modular Block Diagram of the combinational Circuit for a counter

The UP/DOWN Control is a sequence detector and is designed with MUX's for the combinational logic and flip-flops for the sequential logic. There will be a single 8-to-1 MUX and a single D flip-flop for each output variable.

The Loading Control will contain an extra set of 3 MUX's (2-to-1) that will allow the input of a load.

Objectives

To count from 0-7 when counting up and 7-0 when counting down. The load line will allow for the counter to start counting from any specified number (0-7). The counter will wrap back to 0 from 7 when counting up and back to 7 from 0 when counting down.

Equipment

Description	Chip Number	Quantity
2-Input NAND gate	7400	1
8-line to 1-line Multiplexers	74151	3
Quad 2-line to 1-line Multiplexers	74157	3
Dual positive-edge triggered D flip-flops	7474	3

Description

The first step to design the up/down counter component is to draw the next state diagram. This will be done using the Moore model. There will be eight states S_0 to S_7. The output Z is negligible and simply equals the current state. When U=0 the states will count downwards and when U=1 the states will count upwards.

The second step is to write the next state table. There will be no output column Z as Z=current state. Following the next state table, the third step will be to write the transition table, assigning each state to be represented by three bits. The three input bits will be called Q_0, Q_1, and Q_2 (present state) and the three output bits will be called X_0, X_1, and X_2 (next state).

The fourth step is to determine the inputs for each multiplexer (X) in terms of U with select lines Q_0, Q_1, and Q_2. We will use multiplexer design tables. Because there are three select lines we will have $2^3 = 8$ inputs.

To design the second component of the circuit, the load control, we will insert a second set of multiplexers between the output of the 8-to-1 MUX's we just designed and the D flip-flops. These will be three 2-to-1 MUX's and will accept their respective output of the 8-to-1 MUX's (X_0, X_1, or X_2) as their first input. The second input will be I_0, I_1, or I_2. These allow us to directly input the state we want

to ‘load’ the circuit with. The select line for the 2-to-1 MUX’s will be the load line input (Load) for all three. The outputs will be called D_0, D_1, and D_2 as they will be the D flip-flops input.

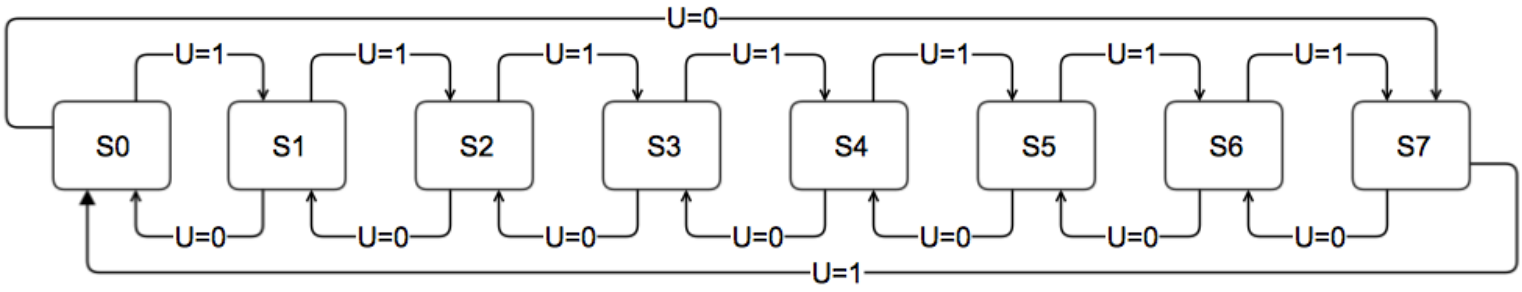
Finally, the three D flip-flops will accept their respective input from the 2-to-1 MUX’s and will output X_0, X_1, or X_2 (which we defined as the next state). We will feed back these outputs into the select lines of the 8-to-1 MUX’s (Q_0, Q_1, and Q_2) since the present state = next state at every clock pulse.

Specifications

Must use MSI components (multiplexers). Must design counter to be loadable and up/down capable.

Design Synthesis

Step 1: the next state diagram. Output Z is not shown.



Step 2: the next state table. U is the up/down input. U=0 counts down and U=1 counts up.

Present State/Z	U = 0	U = 1
S_0	S_7	S_1
S_1	S_0	S_2
S_2	S_1	S_3
S_3	S_2	S_4
S_4	S_3	S_5
S_5	S_4	S_6
S_6	S_5	S_7
S_7	S_6	S_0

Step 3: the transition table. Each state is represented by three binary bits (Q_0, Q_1, and Q_2). X_0, X_1, and X_2 represent each of the 8-to-1 MUX's. Q = present state and X = next state.

			U = 0			U = 1		
Q_0	Q_1	Q_2	X_0	X_1	X_2	X_0	X_1	X_2
0	0	0	1	1	1	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	0	0	1	0	1	1
0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	1
1	0	1	1	0	0	1	1	0
1	1	0	1	0	1	1	1	1
1	1	1	1	1	0	0	0	0

Step 4: the multiplexer design tables. Using MUX's instead of logic gates saves money on components and makes for a simpler circuit. The present state bits Q_0, Q_1, and Q_2 will remain static and only U will be varied.

$Q_0Q_1Q_2$	000	001	010	011	100	101	110	111
U=0	1	0	0	0	0	1	1	1
U=1	0	0	0	1	1	1	1	0
	U'	0	0	U	U	1	1	U'

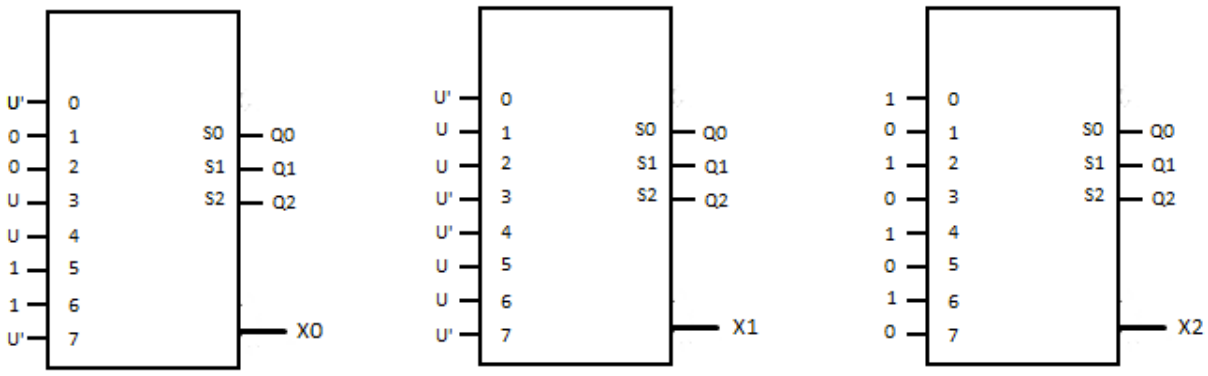
X_0

$Q_0Q_1Q_2$	000	001	010	011	100	101	110	111
U=0	1	0	0	1	1	0	0	1
U=1	0	1	1	0	0	1	1	0
	U'	U	U	U'	U'	U	U	U'

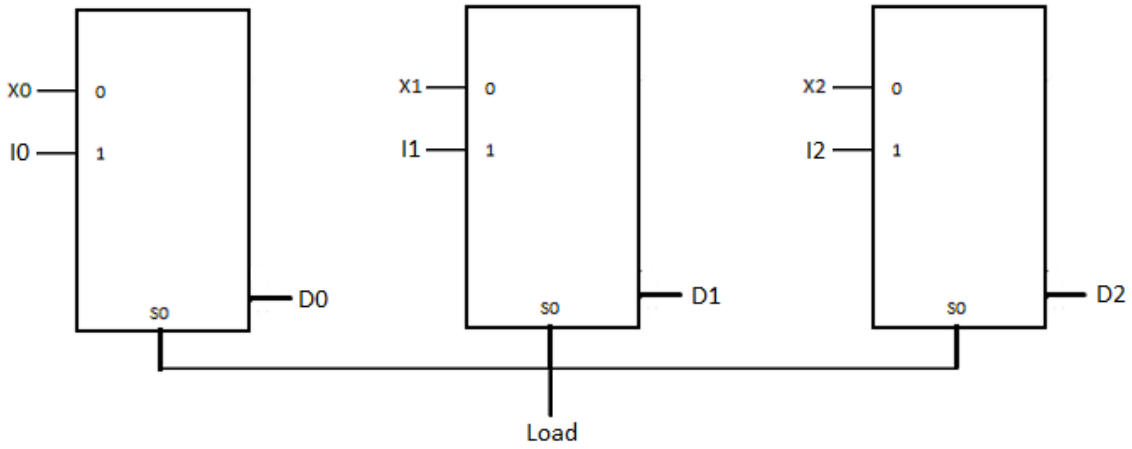
X_1

$Q_0Q_1Q_2$	000	001	010	011	100	101	110	111
U=0	1	0	1	0	1	0	1	0
U=1	1	0	1	0	1	0	1	0
	1	0	1	0	1	0	1	0

X_2

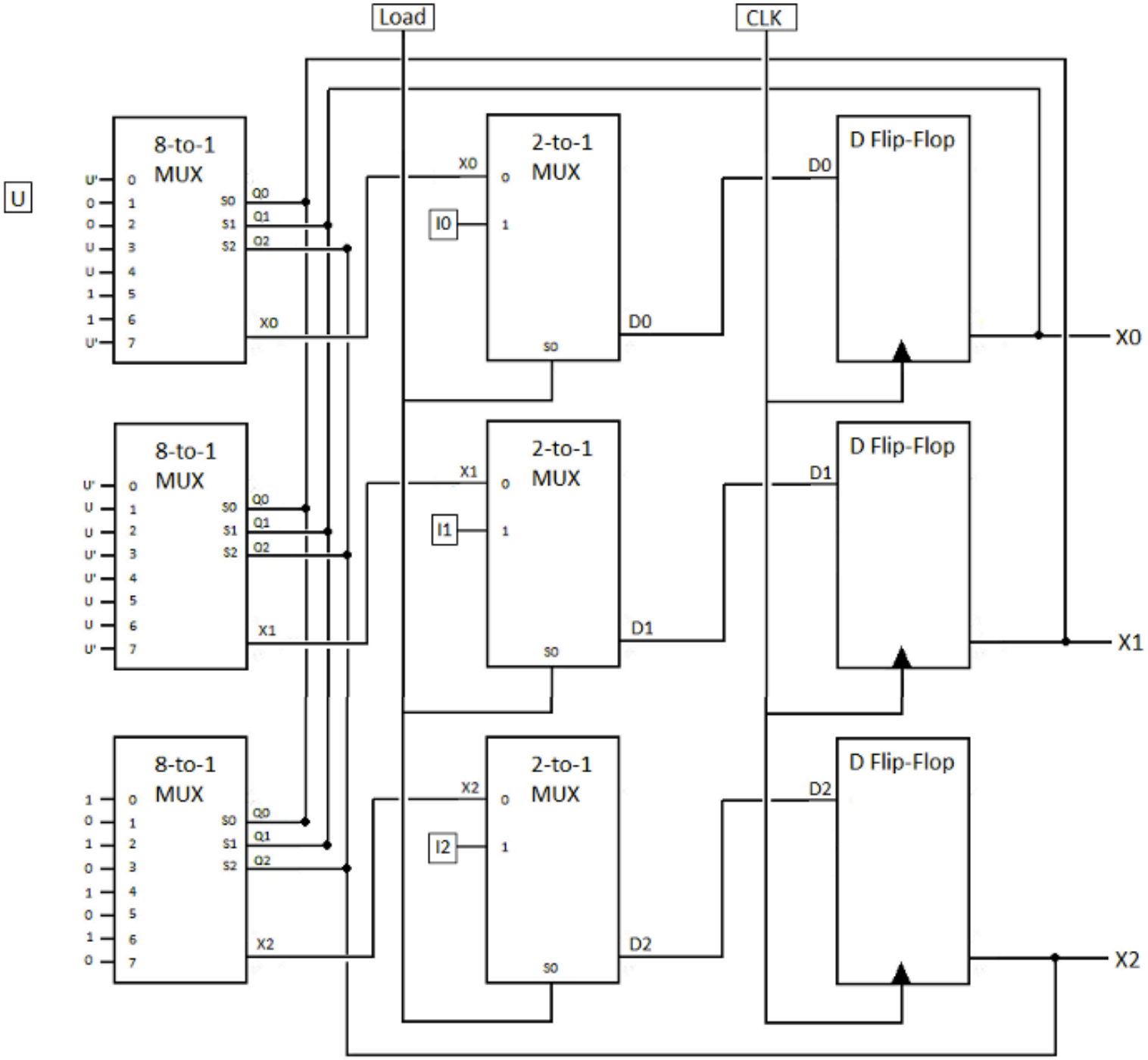


The three 8-to-1 MUX's will be configured liked so. The select lines (S_0, S_1, and S_2) are shown using the present state variables Q_0, Q_1, and Q_2. The outputs X_0, X_1, and X_2 represent the next state of the counter. X_0 is the most significant bit and X_2 is the least significant bit.



The three 2-to-1 MUX's will be configured in this way. They will accept the outputs from the previous three MUX's and the three external inputs (I's) for their two inputs and the load line for their one select line. When the circuit is counting (Load = 0) the input will be taken from the previous MUX's (X's), and when the circuit is accepting a load (Load = 1) the input will be taken from the external inputs (I's). The 2-to-1's output is called D_0, D_1, and D_2, as they will be the inputs to the D flip-flops.

Complete Logic Diagram

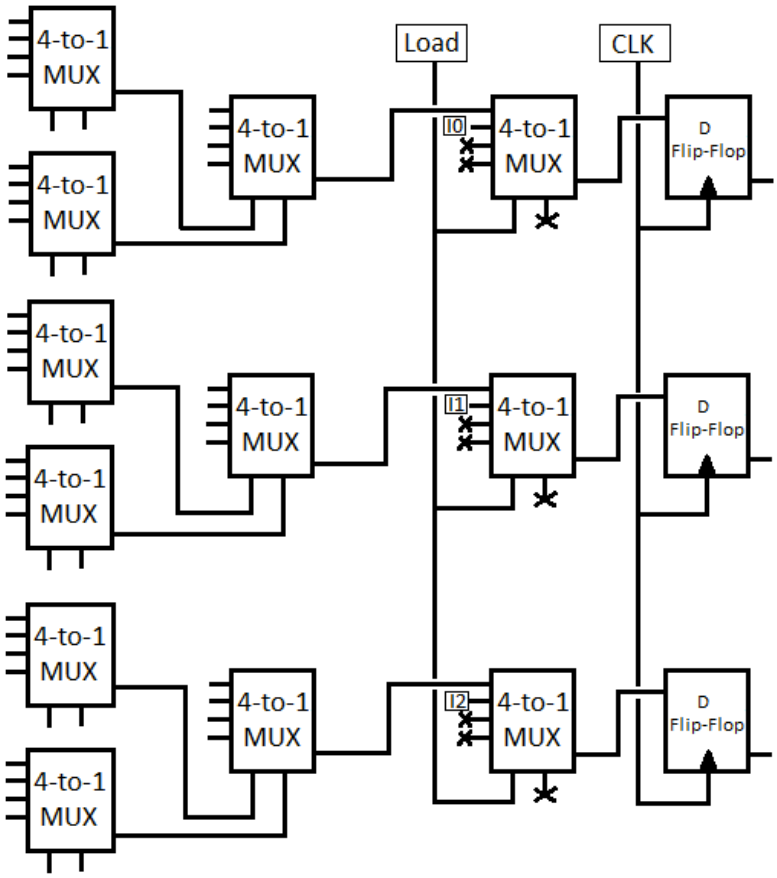


Results and Simulations

The counter successfully counted up when U=1 and down when U=0 at every clock pulse. The user could input a 3-bit load at any time. We encountered no major problems in our testing.

Answers to the questions in the lab handout

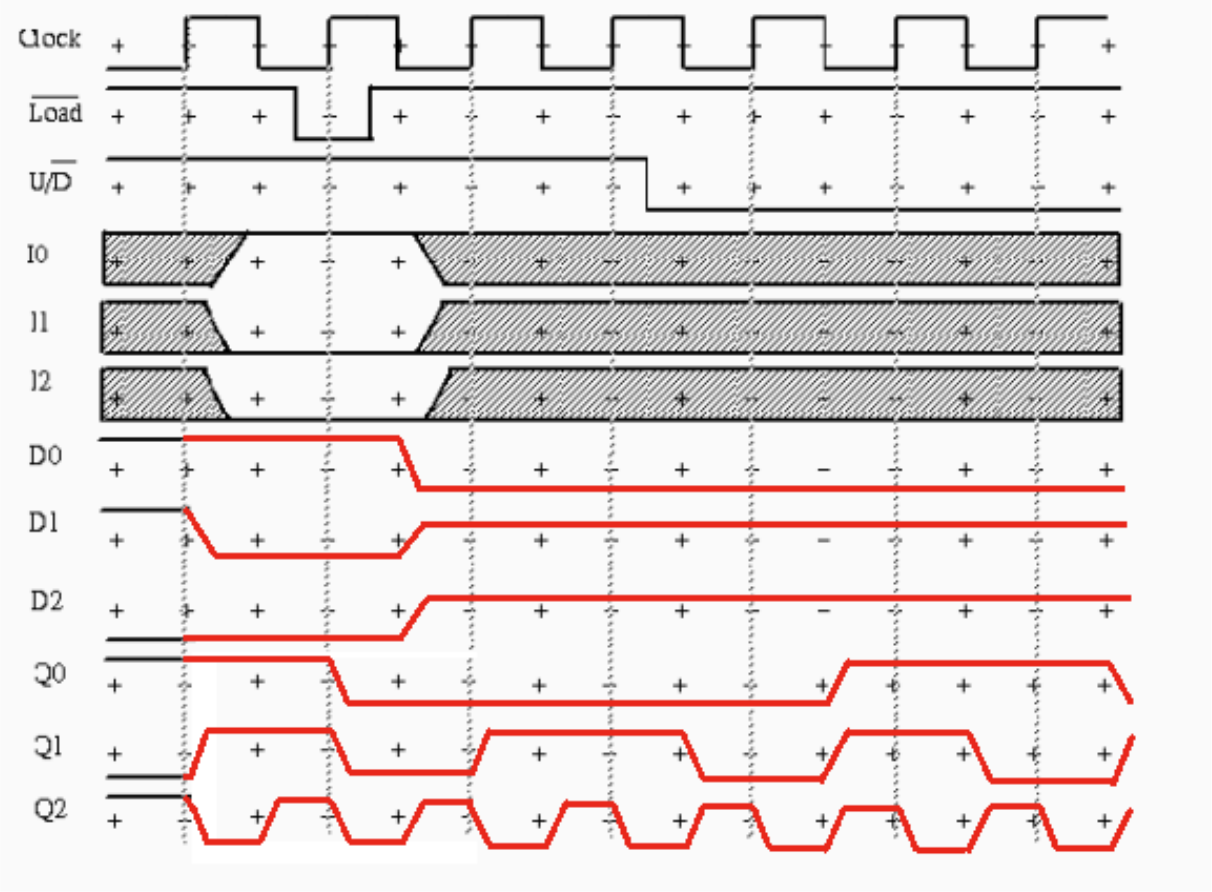
1. *Modify your initial design using 4:1 MUX's and no gates. Is it more convenient to use the MUX's? Explain.*



Using only 4-to-1 MUX's the counter would look something like this. It is less convenient because we would only be able to compare two variables at once so it would require 3 MUX's to achieve what we could achieve with a single 8-to-1 MUX. For the load control, we would have to neglect some of the inputs, wasting more space.

2. *Define the following terms:*
    a. *Synchronous loading* – A type of loading where the data is loaded at every clock pulse and every flip-flop is synched to the same clock.
    b. *Asynchronous loading* – A type of loading where the data is loaded independently from the clock pulse.
    c. *Ripple-carry output (RCO)* – An output signal that outputs high when the last state of the counter is reached. It is useful for determining when the counter has started over counting from the initial state.

3. *Complete the timing diagram given, for the circuit implemented.*



Conclusion

Our circuit ended up successfully built on our second try. On the first build, we rushed through the process and didn't check for errors while connecting the many inputs to the MUX's. On the second build, we made sure to check every connection we made and every component we used. This lab taught us the importance of debugging and thoroughly testing each IC to make sure we will achieve our desired results on the first build every time. Additionally, this lab exposed me to designing digital circuits using Medium Scale Integration as a means of simplification and ease.

Works Cited

None

Signed OFF

[Included]