Digital Design Lab
EEN 315

Lab 2
3-bit Loadable Up/Down Sequential Counter

Group 2
Andrew O'Neil-Smith
(Partner: Austin Clifton and Amy Zamon)

Sayan Maity, TA

University of Miami

October 4, 2013

Abstract

This lab was to build a counter and make it loadable. One switch would dictate whether the display would increment up or down to and from seven for each clock cycle. Another switch selected between counting and loading. The last three switches were used to load values when you incremented the clock cycle. The result was a circuit that is an up and down counter with a loading function.

Table of Contents

Overview

For this lab, it is important to understand state machines. Boolean algebra is also required. Knowledge from the previous lab about NAND gates, D Flip-Flops, multiplexors, and seven segment displays is also required.

Objectives

The objective of this lab was to utilize MUXs to go from a counter to a loadable input. The results are shown on the seven segment display. It was to show the advantage of a medium scale integration implementation.

Equipment

Description	Chip Number	Quantity
Quad 2---Input NAND gates	7400	1
Data selectors/Multiplexers	74151	3
Quad 2-line to 1-line Data selectors/Mult iplexers	74157	1
Dual Positive Edge-Triggered D FF	7474	2

Description

First we had to make the Truth Tables, K-Maps, MUX design tables, and Next State Tables for building the counter and loader. The 2:1 MUX will decide to display the loaded value or the value from the 8:1 MUX. We used 0,1,U, and U' as inputs to the 8:1 MUX. The inputs for the 2:1 MUX were coming from a switch (load) and an output from one of the MUXs. There was also a select switch going to the 2:1 MUX. Each output went to a D Flip-Flop. The flip-flop outputs went to both the LED display and back into the 8:1 MUX. We had to ground the MSB for the display to get it to work. The D Flip-flips were also sent the clock pulse.

Specifications

Use multiplexors only. Use three 2:1 MUXs and three D flip flops.

Design Synthesis

Truth Table

U/D	X0	X1	X2	P0	P1	P2
0	0	0	0	1	1	1
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	1	0	1
0	1	1	1	1	1	0
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	1	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1
1	1	1	1	0	0	0

When U/D is 0, it counts down. When it is 1, it counts up. X values are the present state. P values are next state for the counter.

K-Maps

X0/X1 X2/ U/D	**P0**			
	1	0	1	0
	0	0	1	1
	0	1	0	1
	0	0	1	1

X0/X1 X2/ U/D	**P1**			
	1	0	0	1
	0	1	1	0
	1	0	0	1
	0	1	1	0

X0/X1 X2/ U/D	**P2**			
	1	1	1	1
	1	1	1	1
	0	0	0	0
	0	0	0	0

Boolean Equations

$P0 = X'_0X'_1X'_2U' + X_0X_1U' + X_0X'_2U + X_0X'_1X_2$

$P1 = X'_1X'_2U' + X_1X'_2U + X'_1X_2U + X_1X_2U'$

$P2 = X'_2$

MUX Design

Complete Logic Diagram

Answers to the questions in the lab handout

1.

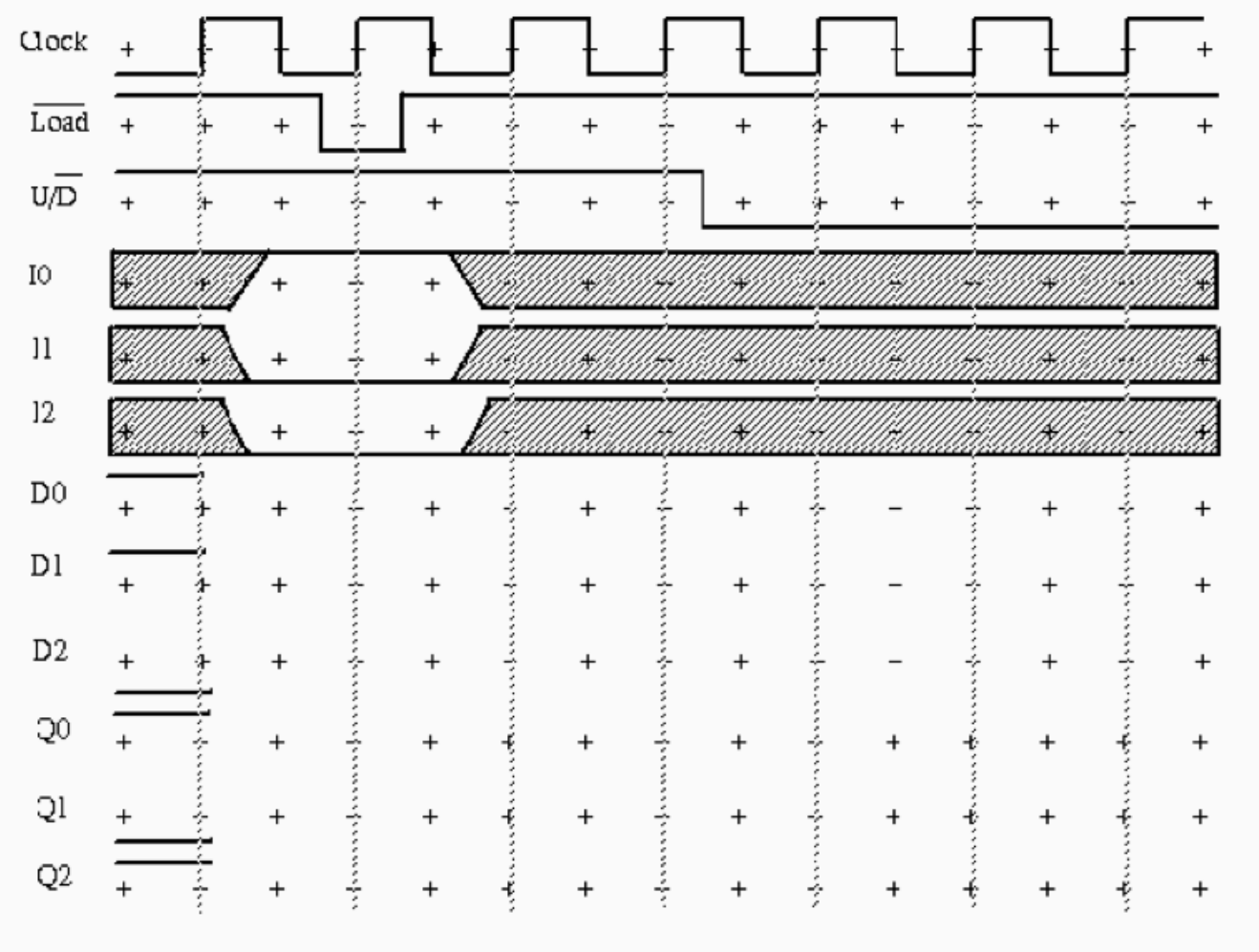
It would be less convenient to us 4:1 with no gates. It requires extra MUXs. It would likely be more costly to implement because there are more physical ICs. It also takes more physical space to implement.

2.

Synchronous Loading is loading the data with the pulse of a clock input and clock signal.
Asynchronous Loading is loading of data independent of a clock pulse.

3.

Ripple Carry Out is an output that shows high upon completion of the count up or down.



Conclusion

This lab was not as time consuming as the first lab. We got stuck because it is hard to debug the implementation when everything depends on everything else. Our 8:1 MUXs were not working so we moved on to the load circuit and finished that. Then we went back to the other part and were able to get it to work. I learned how to use the 8:1 and 2:1 MUXs and got a better understanding for how counters work.

Works Cited