Digital Design Lab
EEN 315

Lab 3
Sequence and Tone Generator

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

Sayan Maity, TA

University of Miami
October 22, 2013

Abstract

This lab uses counters to generate different tones. The idea of the lab was to create a State Machine and design a circuit with a modulo-n counter. Part A used a 555 timer to generate a base frequency. Flip Flops were used divide frequency in Part B. Our final circuit worked well.

Table of Contents

Overview

The lab uses combinational and sequential logic. The RCO of a counter was sent to the clock of a JK FF to create frequency division. The FF output is the base frequency divided by the input plus one. We needed to know KMaps, how to use a MUX, and how to do a State Machine.

Objectives

The objective was to first generate the given sequence (0, 11, 14, 13, 8, 9, 12, 3, 2, 10, 5, 6, 1). Then it was to divide the frequency and play it as a tone.

Equipment

Description	Chip Number	Quantity
14 pin Quad gate 2 input each	7400	1
14 pin D FF	7474	2
16 pin 8x1 MUX	74151	4
16 pin 4x1 MUX	74157	1
16 pin counter	74169	1
16 pin JK FF	7476	1
8 pin 4x1 timer	555	1
1.8k and 3.6k	Resistor	2
10nF and 47uF	Capacitor	2

Description

The first step was to generate a Truth Table for the given sequence (0, 11, 14, 13, 8, 9, 12, 3, 2, 10, 5, 6, 1) seen below. Then we made KMaps. The MUX design was relatively easy after that.

The next part was to build the circuit. The Control Lines were the MSBs from the D line. On the 74169, Load' was grounded and 1 was sent to ENT and ENP. The RCO pin was the clock to the JK FF. The outputs went to the LEDs.

Specifications

- Ra & Rb 1.8Kohm and 3.6Kohm based on the ratio.
- Capacitor for 555-IC circuit=> 1 NanoFarad(or 1 KiloPicoFarad)
- Capacitor for Speaker => 47 microFarad
  - given sequence (0, 11, 14, 13, 8, 9, 12, 3, 2, 10, 5, 6, 1)

Design Synthesis

Truth Table

N	QA	QB	QC	QD	QA'	QB'	QC'	QD'
0	0	0	0	0	1	0	1	1
1	0	0	0	1	0	0	0	0
2	0	0	1	0	1	0	1	0
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	0	0	0
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	0	0	1
7	0	1	1	1	0	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	1	1	0	0
10	1	0	1	0	0	1	0	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	0	0	1	1
13	1	1	0	1	1	0	0	0
14	1	1	1	0	1	1	0	1
15	1	1	1	1	0	0	0	0

K-Maps

A/B		**QA**		
C/D	1	0	0	1
	0	0	1	1
	0	0	0	1
	1	0	1	0

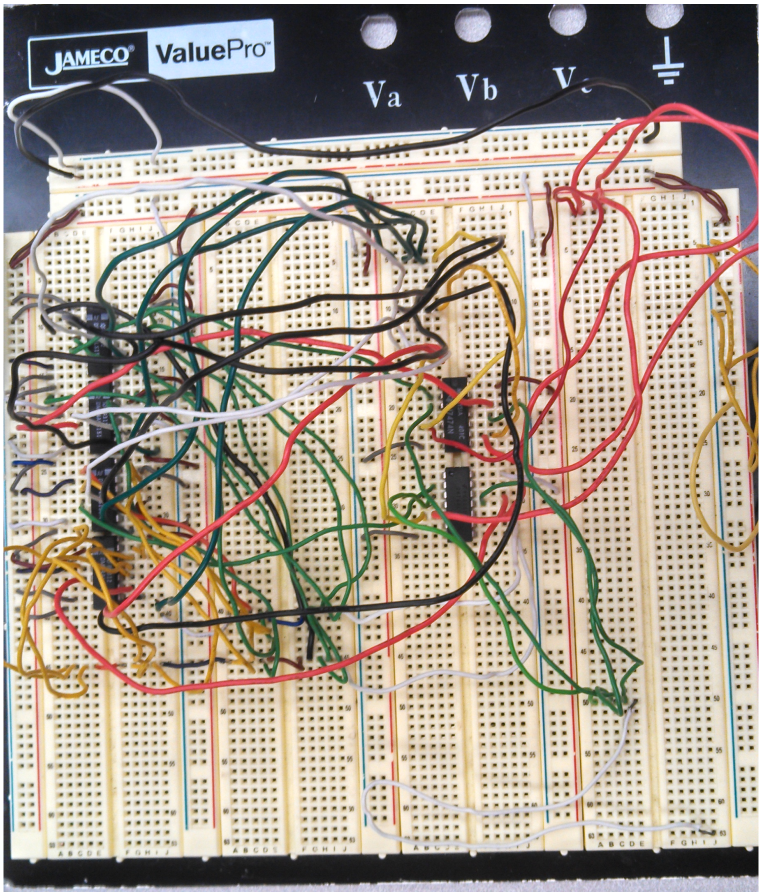
A/B		**QB**		
C/D	0	0	0	0
	0	1	0	1
	0	0	0	1
	0	0	1	1

A/B		**QC**		
C/D	1	0	1	0
	0	0	0	0
	1	0	0	1
	1	0	0	0

A/B		**QD**		
C/D	1	0	1	1
	0	0	0	0
	0	0	0	0
	0	1	1	1

Complete Logic Diagram

Results and Simulations



Sequence generator

Answers to the questions in the lab handout

1. The max base frequency depends on the 555 timer. Minimum is approximately 40MHz for the 74169. Max of 7476 is approximately 30MHz. The output of 555 is approximately 500kHz as a result of the pulse width (1us at +5V)

2.

3.

4.

in order to keep the circuit the same, you need to divide by the clock frequency, except 15 minus the load number. If 10 was loaded, it counts up to 15 and gives the RCO pulse. If you load 10 again, the result is N=6(15-10)

5.

6.

7.

The counter based on which inputs that are active at that pulse would load a number that was unaccounted for. The timing of the IC is so fast which makes it so the sound is not noticeable by the human ear. The following RCO pulse given by the counter will load again based on the previous unaccounted load, yet with the correct value. The error time would be Load Value + 1 clock pulse. Having an IC that detects when the clock button gets pressed prevents this.

Conclusion

We set up the correct sequence in part A. Our tone generator played different sounds for each clock pulse. Our implementation was successful.

Works Cited