

# Lab 7 - Sequential Logic Circuits (III)

---

CDA 3201L-003

Jason Keene and Jacob Manfre

Submitted April 5th, 2015

## Purpose and Objectives

In this lab we set out to implement a nine state sequential circuit using JK flip flops that counts the following series:

☐ 0011, 0101, 1001, 1000, 1011, 1010, 0110, 0100, 0111, ...

## Component List

- Breadboard
- Wiring
- 5v power supply
- TTL 0.5 Htz Function Generator
- 4 x Dual J-K Flip-Flops With Preset And Clear (74LS76)
- 1 x Hex Inverter (74LS04)
- 4 x Quad 2-input OR Gate (74LS32)
- 6 x Quad 2-input AND Gate (74LS08)
- 4x LEDs
- 4x Resistors (470 Ohms 5%)

## Design

We first created a state diagram (fig 1). Using that diagram we created a state table (fig 2) including x input, previous state input, next state output, z outputs, and the required inputs for the JK flip flops to transition to the next state.

A flaw in our design was to include x as an input. This complicated the design as we could have simply ANDed the x input with the clock. Additionally if we didn't use an arbitrary mapping for the states and instead assigned each state the mapping for the output of that state we would have been able to have the output of the JK flip flops be the output of the circuit.

Once we had the state table we were able to do the k-maps for the inputs to the JK flip flops and for the mapping of x and the outputs of the flip flops to the outputs of the circuit (fig 3).

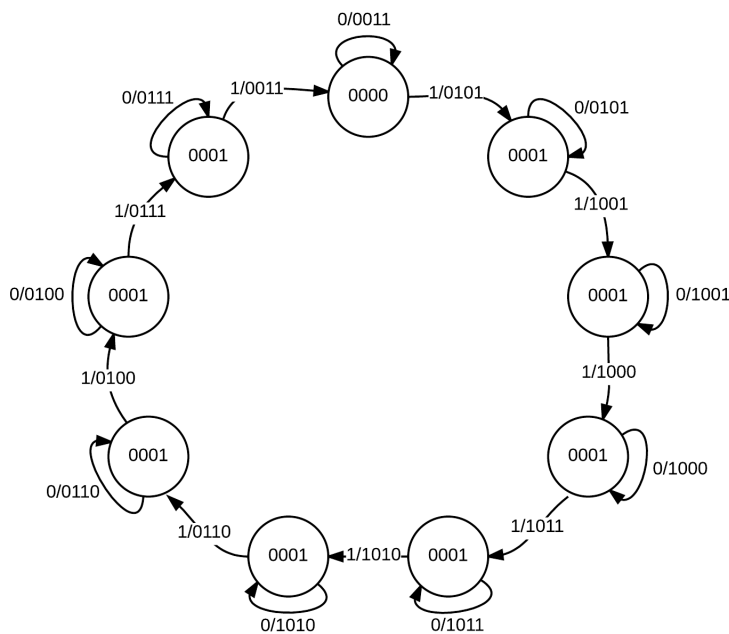
This gave us the necessary equations to implement the design in logisim (fig 4 and fig 5).

## Discussion and Conclusion

While our design worked in simulation it was clearly over complicated. The resulting combinational circuit was so complicated that we failed to implement it in hardware. We learned that if you can get away with it that you should try to assign a meaningful mapping of the flip flop states to the outputs. This simplifies the combinational part of the circuit. We also learned that limiting your inputs as much as possible simplifies the combinational part as well.

## Figures

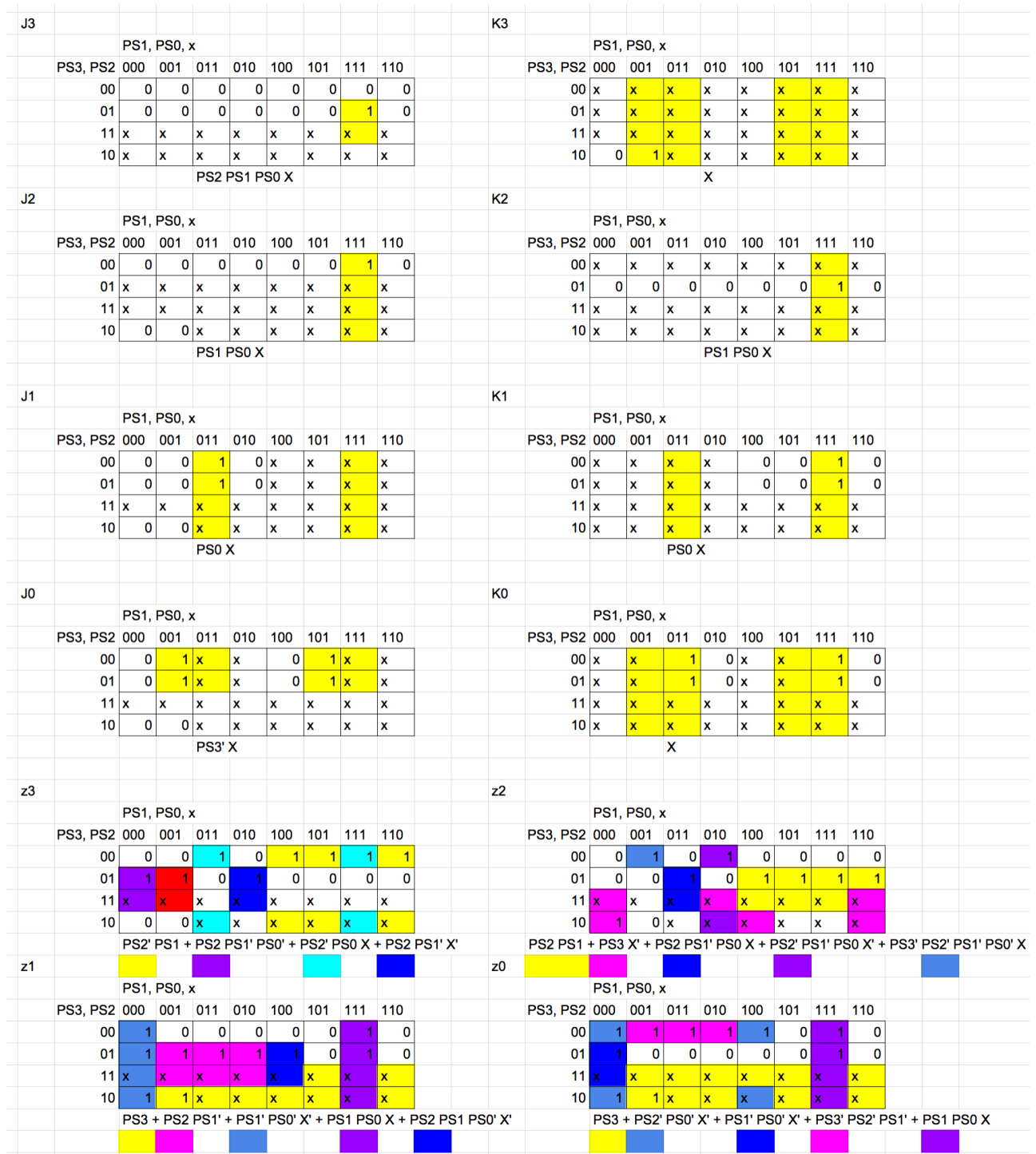
**Figure 1 - State Diagram**



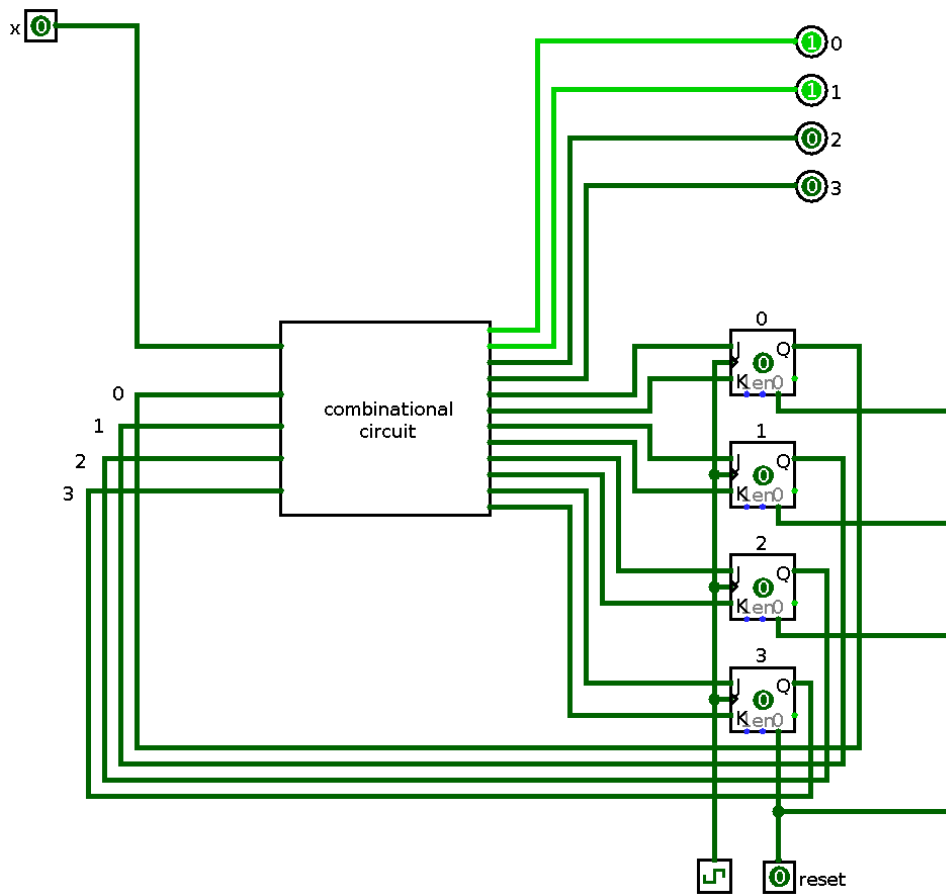
**Figure 2 - State Table**

inputs					outputs																				
ps3	ps2	ps1	ps0	x		ns3	ns2	ns1	ns0	z3	z2	z1	z0		J3	K3		J2	K2		J1	K1		J0	K0
0	0	0	0	0		0	0	0	0	0	0	1	1		0	X		0	X		0	X		0	X
0	0	0	0	1		0	0	0	1	0	1	0	1		0	X		0	X		0	X		1	X
0	0	0	1	0		0	0	0	1	0	1	0	1		0	X		0	X		0	X		X	0
0	0	0	1	1		0	0	1	0	1	0	0	1		0	X		0	X		1	X		X	1
0	0	1	0	0		0	0	1	0	1	0	0	1		0	X		0	X		X	0		0	X
0	0	1	0	1		0	0	1	1	1	0	0	0		0	X		0	X		X	0		1	X
0	0	1	1	0		0	0	1	1	1	0	0	0		0	X		0	X		X	0		X	0
0	0	1	1	1		0	1	0	0	1	0	1	1		0	X		1	X		X	1		X	1
0	1	0	0	0		0	1	0	0	1	0	1	1		0	X		X	0		0	X		0	X
0	1	0	0	1		0	1	0	1	1	0	1	0		0	X		X	0		0	X		1	X
0	1	0	1	0		0	1	0	1	1	0	1	0		0	X		X	0		0	X		X	0
0	1	0	1	1		0	1	1	0	0	1	1	0		0	X		X	0		1	X		X	1
0	1	1	0	0		0	1	1	0	0	1	1	0		0	X		X	0		X	0		0	X
0	1	1	0	1		0	1	1	1	0	1	0	0		0	X		X	0		X	0		1	X
0	1	1	1	0		0	1	1	1	0	1	0	0		0	X		X	0		X	0		X	0
0	1	1	1	1		1	0	0	0	0	1	1	1		1	X		X	1		X	1		X	1
1	0	0	0	0		1	0	0	0	0	1	1	1		X	0		0	X		0	X		0	X
1	0	0	0	1		0	0	0	0	0	0	1	1		X	1		0	X		0	X		0	X
1	0	0	1	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	0	0	1	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	0	1	0	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	0	1	0	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	0	1	1	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	0	1	1	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	0	0	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	0	0	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	0	1	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	0	1	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	1	0	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	1	0	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	1	1	0		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X
1	1	1	1	1		X	X	X	X	X	X	X	X		X	X		X	X		X	X		X	X

**Figure 3 - K-Maps**



**Figure 4 - Logisim Overview**



**Figure 5 - Logisim Combinational Circuit**

