EECS 170D
Project #4: Prime Sequence
Enoch Chau 45179973
2/22/2020

Introduction

A 3-bit prime sequencer counts prime numbers from 0 to 7 in binary. This sequencer can be achieved using T flip-flops and logic gates. A T flip-flop can be constructed from a JK flip-flop which can be constructed using NAND and NOR gates and an SR flip-flop. This report will go through making a SR flip-flop, JK flip-flop, T flip-flop and the prime number sequencer.

SR Latch Design

An SR latch can be created using four NAND gates. Previous projects have already addressed creating NAND gates.

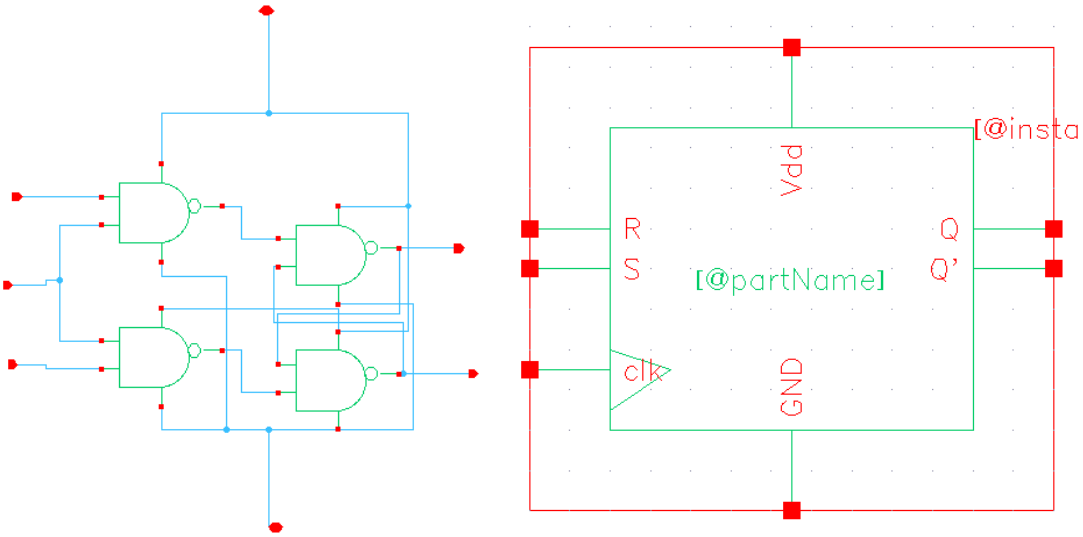

Figure 1: SR Latch Schematic and Symbol

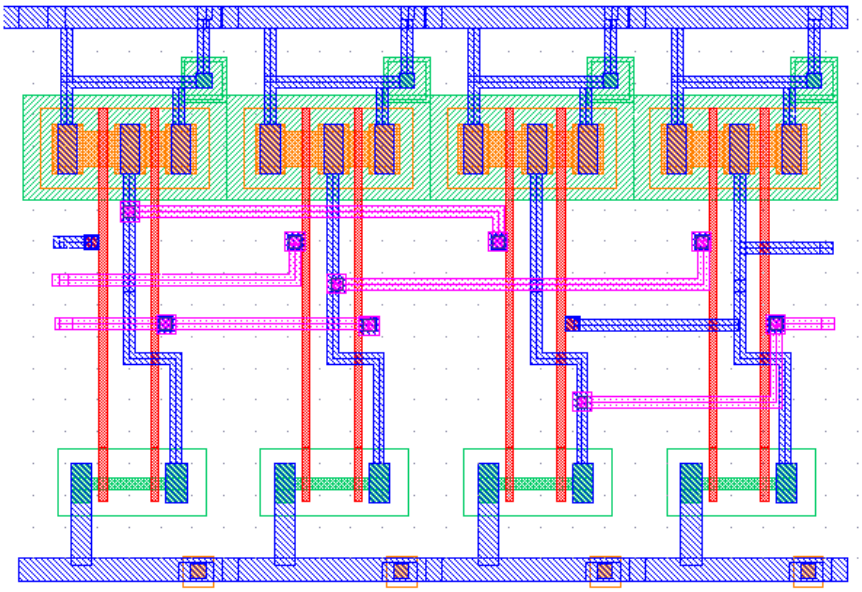

Figure 2: SR Latch Layout

JK Flip Flop Design

This JK flip-flop follows the master-slave layout. It is a negative edge triggered flip-flop. The first JK latch is positive while the second SR latch is negative. Here, the master is made of a JK latch while the slave is an SR latch.

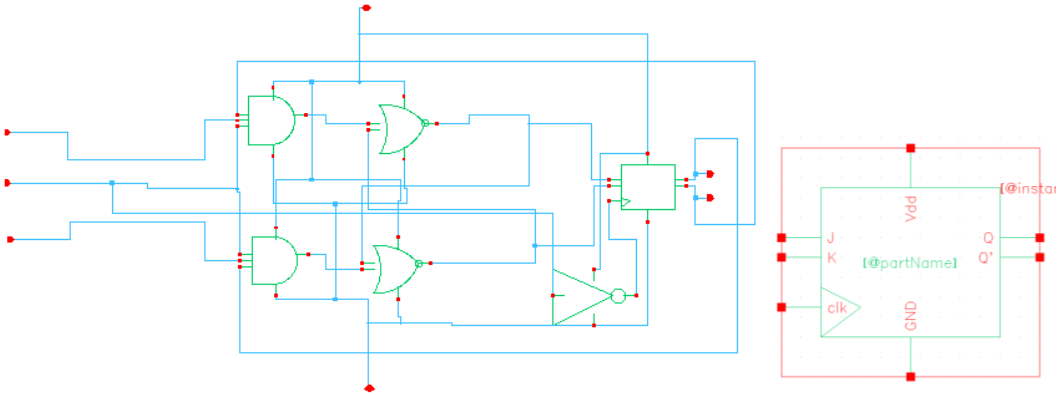

Figure 3: JK Flip-Flop Schematic and Symbol

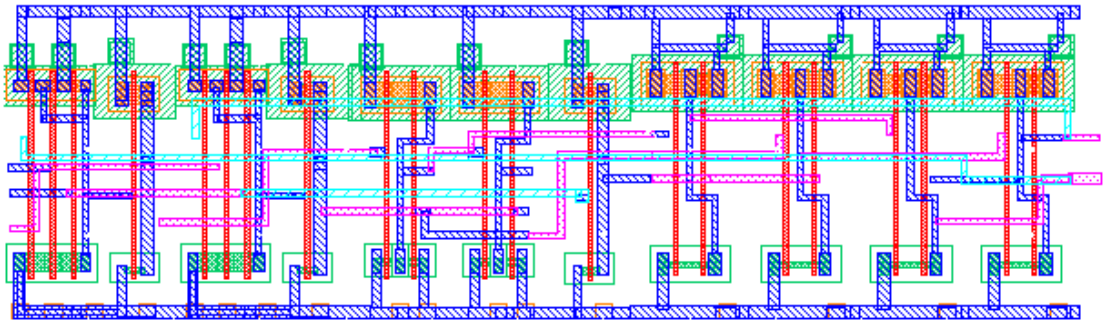

Figure 4: JK Flip-Flop Layout

T Flip-Flop Design

A T flip-flop can be created by simply connecting the J and K inputs of a JK flip-flop.

Prime Sequencer Logic

The sequencer should output prime numbers from 0 to 7. If the sequencer happens to start on non-prime numbers 4 or 6, it will reset to 0. A 3-bit counter can be created using 3 T flip-flops. Having a counter count specific numbers can be achieved using gates. First, all present states, next states, and toggle states were written out. Toggle states can be determined by comparing present and next states. If the bit Q changes between present and next state, the bit T will be 1. If the bit Q does not change, the bit T will be 0. Figure 5 illustrates the states and toggle bits. From Figure 5, we can generate K-maps for each toggle bit. Figures 6,7, and 8 show the K-maps for each toggle bit. From the K-maps, the Boolean equation for each toggle bit can be attained.

$$T2 = Q2*Q0' + Q1*Q0 \quad \text{(eq1)}$$
$$T1 = Q2*Q1 + Q0 \quad \text{(eq2)}$$
$$T0 = Q2'(Q0' + Q1') + (Q2*Q1*Q0) \quad \text{(eq3)}$$

Present State	**Next State**	**Toggle-state**
000	001	001
001	010	011
010	011	001
011	101	110
100	000	100
101	111	010
110	000	110
111	000	111

Figure 5: Q2Q1Q0 State changes & T2T1T0 Toggle States

Q2/Q1,Q0	00	10	11	01
0	0	0	1	0
1	1	1	1	0

Figure 6: T2 – K-map

Q2/Q1,Q0	00	10	11	01
0	0	0	1	1
1	0	1	1	1

Figure 7: T1 – K-map

Q2/Q1,Q0	00	10	11	01
0	1	1	0	1
1	0	0	1	0

Figure 8: T0 – K-map

Prime Sequencer Design

The Boolean equations (eq1, eq2, eq1) are realized using logic gates in conjunction with three T flip-flops.

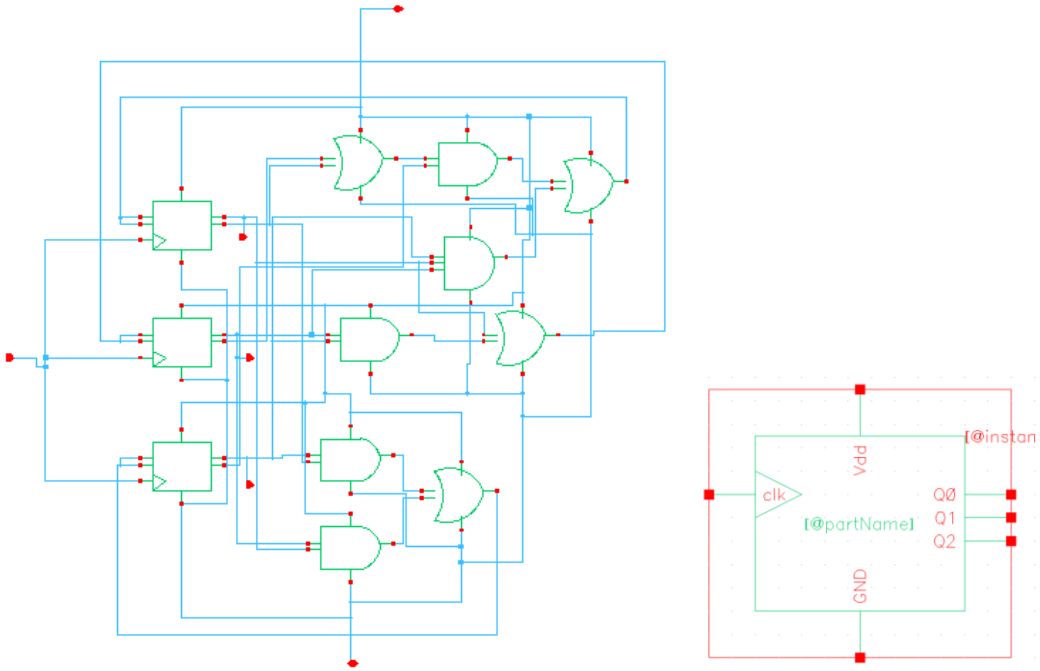

Figure 9: Prime Sequencer Schematic and Symbol

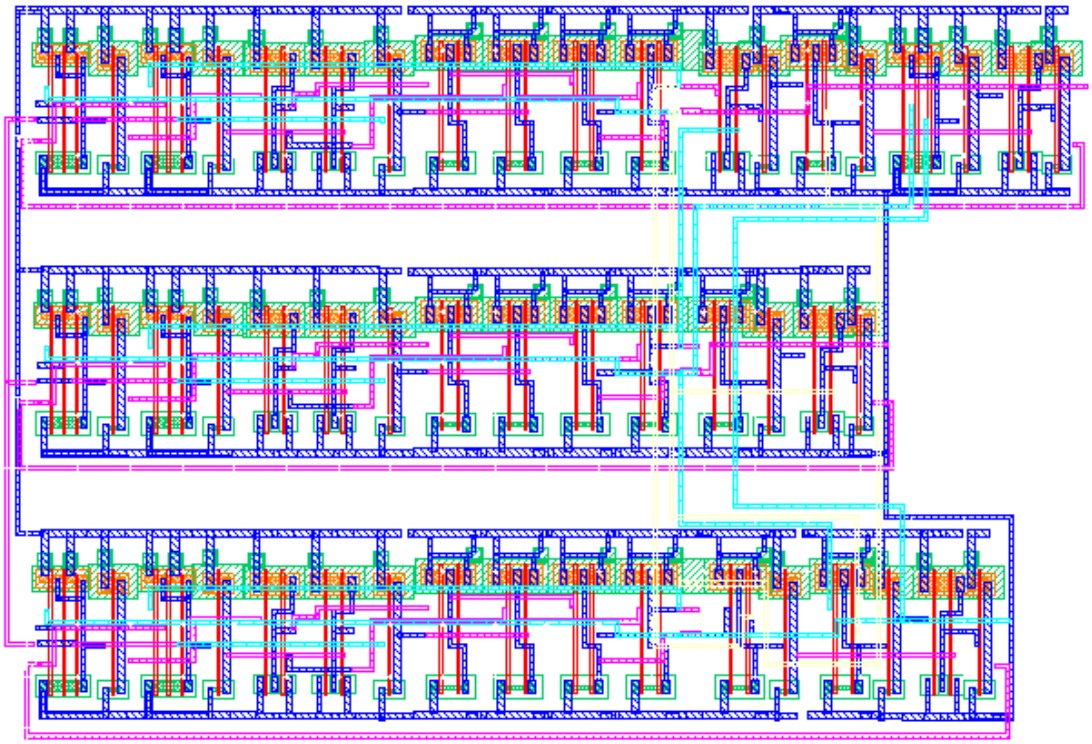

Figure 10: Prime Sequencer Layout

Prime Sequencer Validation

Figure 11 shows how the prime sequencer was tested. A voltage pulse source with a rise time and fall time of 100p was used. Figure 12 shows the outputs when the pulse width was set to 4ns and period set to 8ns. These outputs can be compared to the desired outputs in Figure 5 to see that they match. As stated before, the T flip-flops used are negative edge triggered which can also be observed in Figure 12.

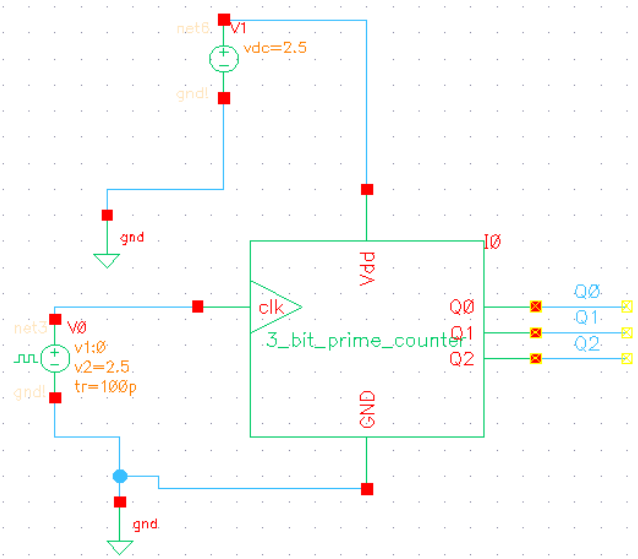

Figure 11: Prime Sequencer Test Circuit

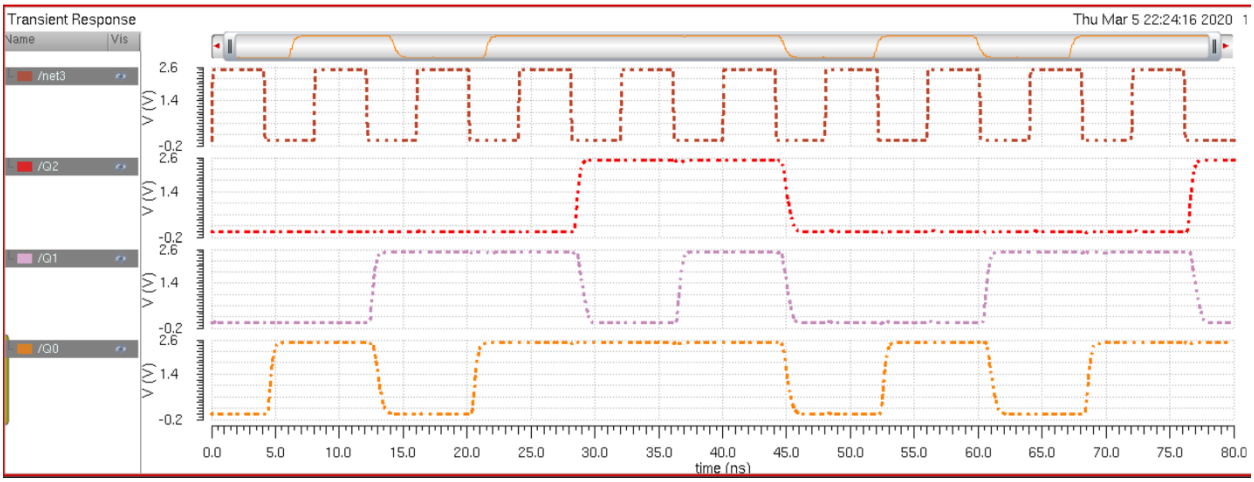

Figure 12: Prime Sequencer Output