COMP ENG 2DI4
Digital Logic

Lab 4: Programmable & Sequential Logic

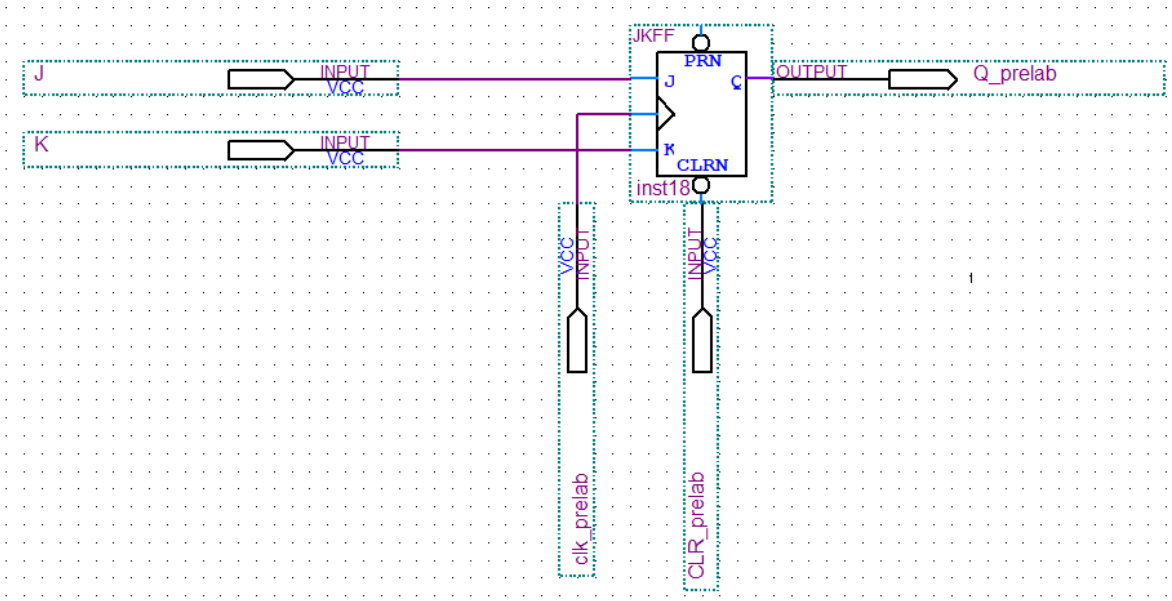
Due date: Friday, November 19th, 2021

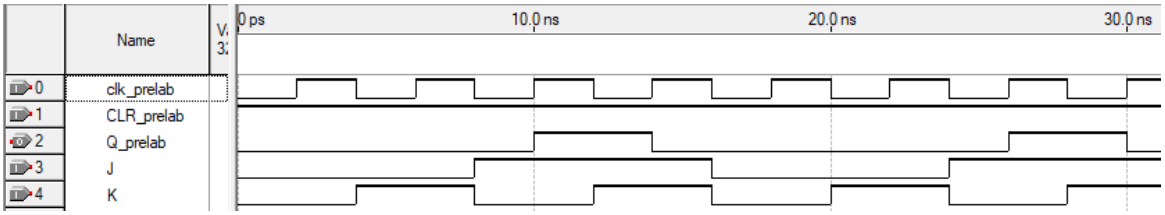
Submitted By:
Jil Shah
Shiv Thakar

Pre-Laboratory Preparation

1. The JK flip-flop is a widely used and available device for the implementation of sequential logic circuit design. The JK is easily used like an SR, T, or D flip-flop.
   a. JK Flip flop

J	K	Q(t+1)	What it does
0	0	Q(t)	No change
0	1	0	reset
1	0	1	set
1	1	Q'(t)	complement



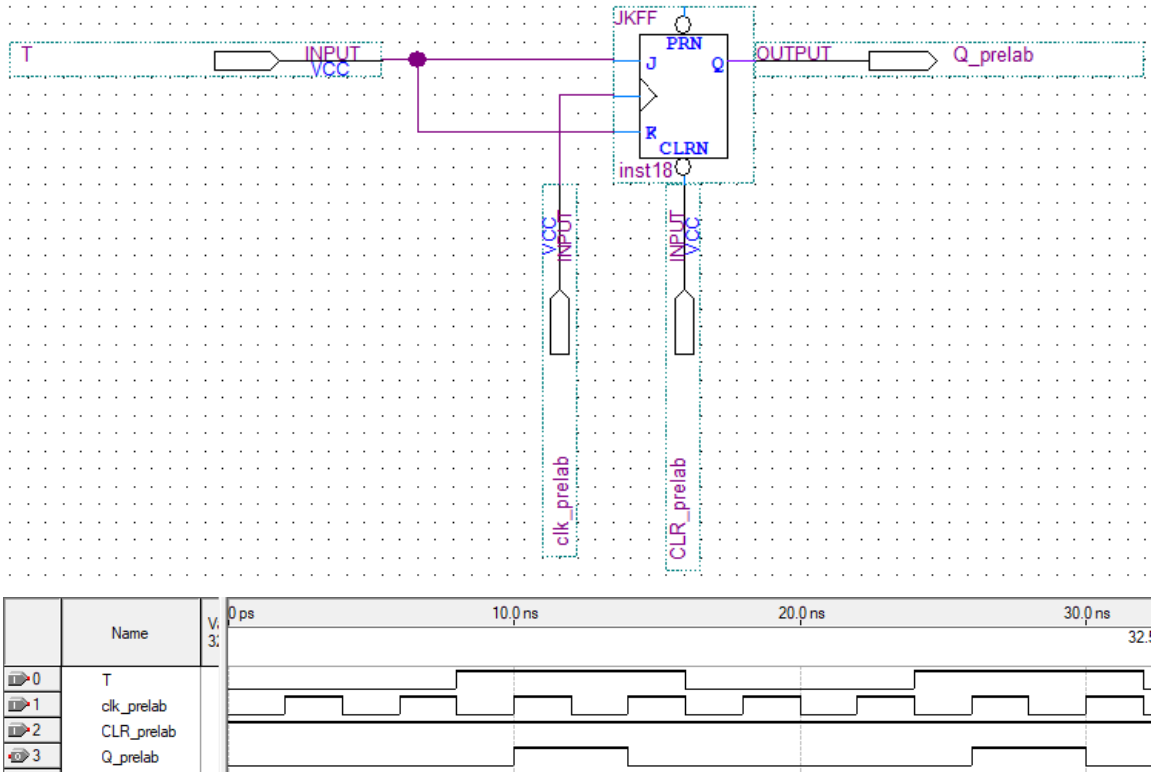


   b. Make Note:

      i. The device above is rising edge. Flip Flops are considered to be edge sensitive.
      ii. A preset in a JK flip flop drives the value of the input to a set state.
      iii. A clear in a JK flip flop drives the input value to a reset state.

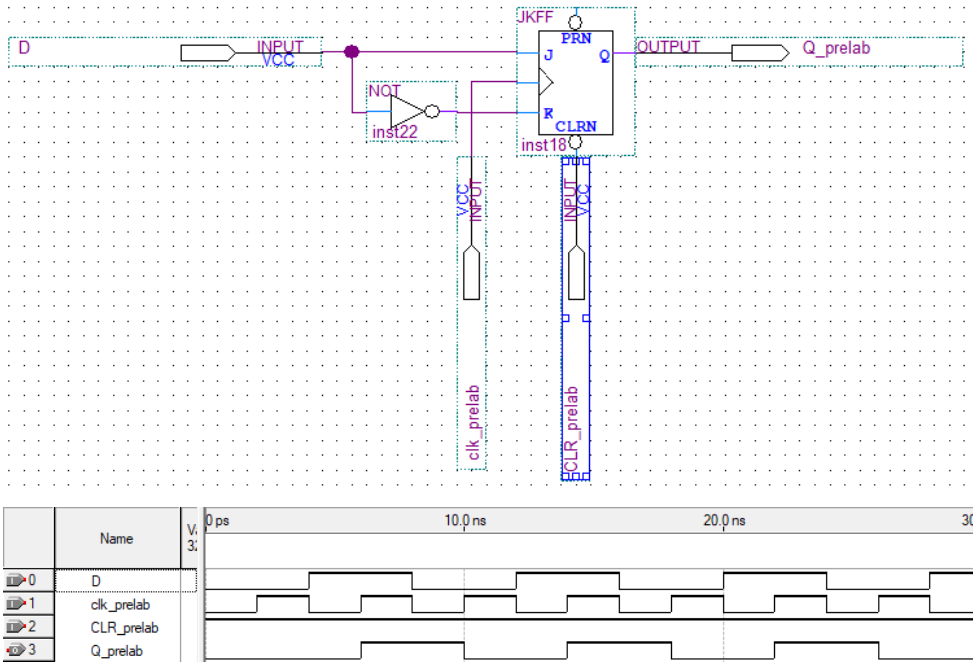
   c. T Flip Flop

T	Q(t+1)	
0	Q(t)	No Change
1	Q'(t)	Complement



d. D flip flop

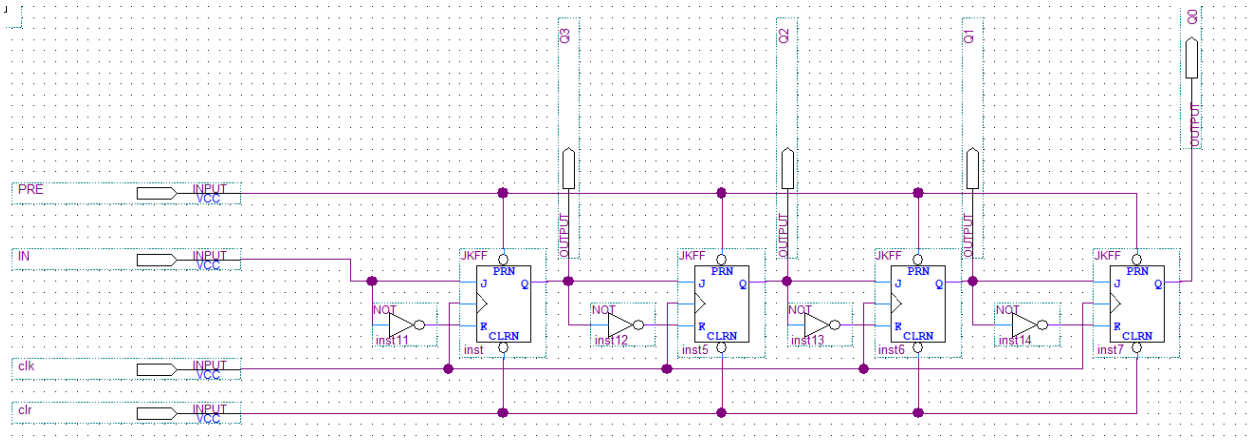
D	Q(t+1)	
0	0	reset
1	1	set

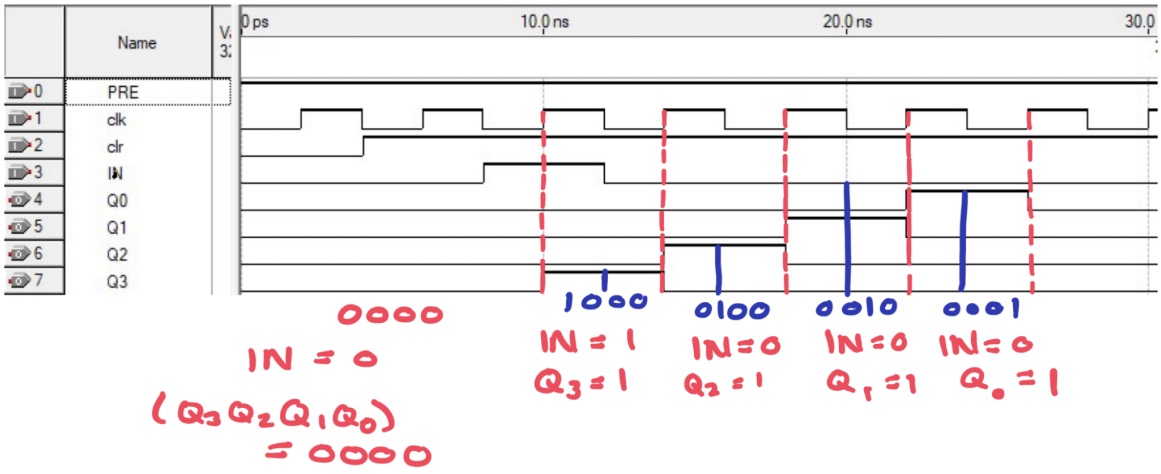


4.5.1 Registers

Left-to-Right Shift Register

Circuit

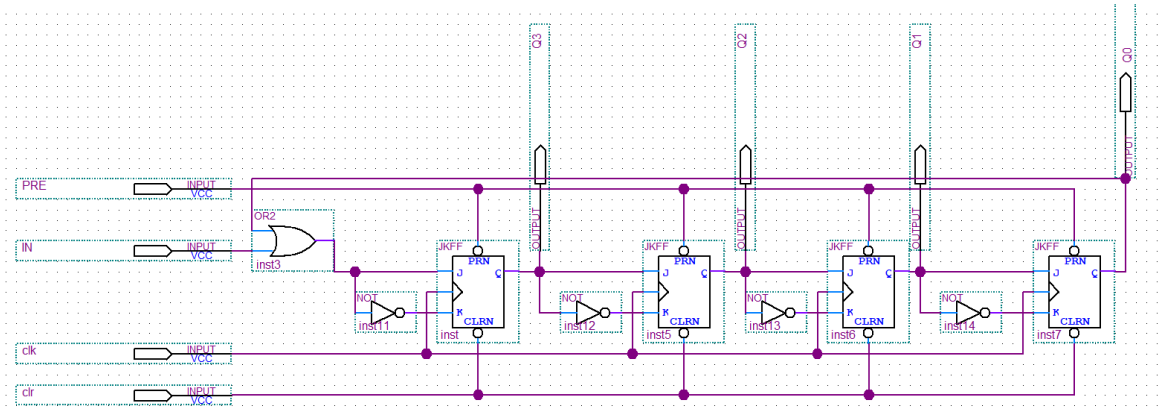


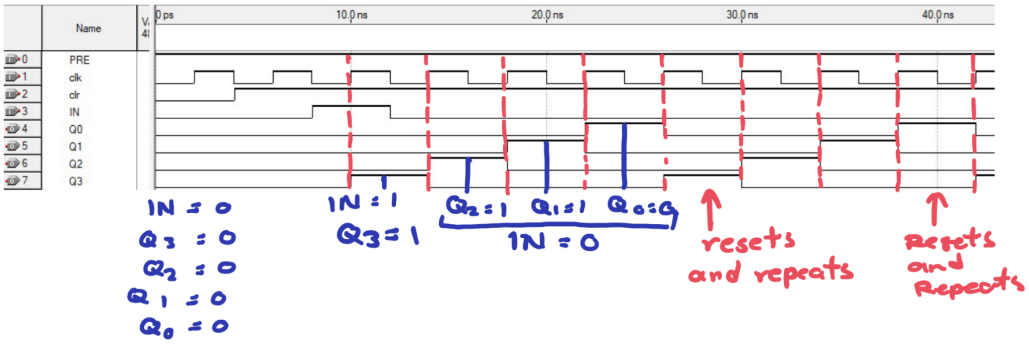


Description

After building the circuit of a left-to-right shift register in Quartus, we conducted a functional simulation that would test the inputs and outputs of the register. A left-to-right shift register is a group of flip flops that is a ble to store bits of data and can shift the bits left or right depending on the inputs of the register. We then tested the circuit, first by setting the preset to HI, with a 2 Hz clock. The outputs we received are shown above.

Circular Shift



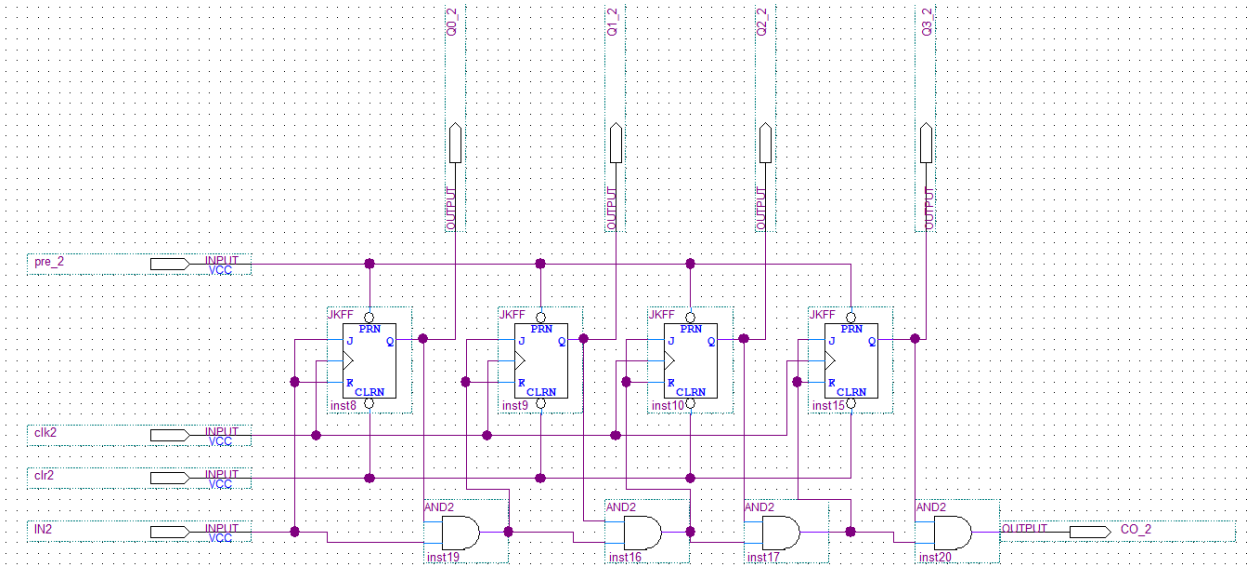


Description

By modifying the circuit used in part 1, we changed the shift register into a circular shift register. By connecting the output of the last bit to an OR gate (IN OR Q0) is used as the input of Q3 (the first bit).

4.5.2 Counters

Circuit:



Test Cases Description:

For the 16Hz - Clocked circuit, we tested this circuit by doing the following calculations and setting the inputs to the following values:

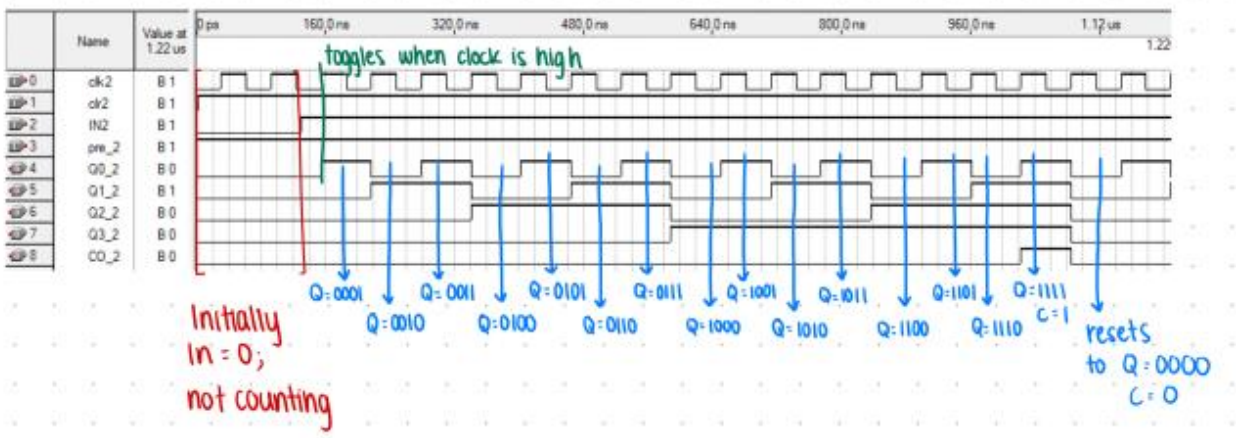
$Time\ Period\ for\ Clock\ =\ 1/16Hz\ =\ 0.0625s$

$Clock\ Time\ flips\ bits\ every\ 31.25\ ns$. This is because a full period will be 62.5ns.

After two periods, with no changes in inputs, the In2 pin will flip to 1 from 0.

The clear is always 1 and the preset is also always 1.

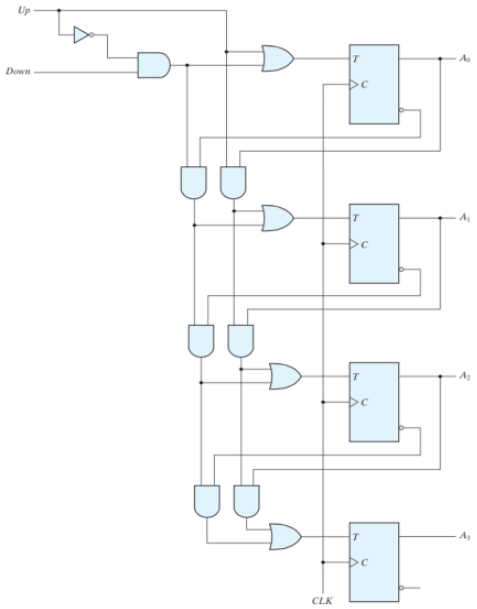
With this testing method, our counter circuit was implemented currectly and with a frequency of 16MHz.



$T\,(Q0) \approx 281\,ns - 156\,ns = 125\,ns \rightarrow f = 1/125 = 0.008\,GHz = 8\,MHz$
$T\,(Q1) \approx 465\,ns - 218\,ns = 247\,ns \rightarrow f = 1/247 \approx 0.004\,GHz \approx 4\,MHz$
$T\,(Q2) \approx 837\,ns - 342\,ns = 495\,ns \rightarrow f = 1/495 \approx 0.002\,GHz \approx 2\,MHz$
$T\,(Q3) \approx 1581\,ns - 589\,ns = 992\,ns \rightarrow f = 1/992 \approx 0.001\,GHz \approx 1\,MHz$
$T\,(CO) \approx 3007\,ns - 1024\,ns = 1983\,ns \rightarrow f = 1/1983 \approx 0.0005\,GHz \approx 0.5\,MHz$

4.5.3 Up-Down Counter

For a 4-bit input up-down counter that goes from 0-9, we know that the binary values will increase from 0000 to 1001. From the lecture notes, we have looked into a 4-bit synchronous up-down counter.



Truth Table for the Up-Down Counter (used for testing purposes)

	Present				In	Next State				Outputs				FF inputs			
	a	b	c	d	x	a	b	c	d	Y3	Y2	Y1	Y0	TA	TB	TC	TD
0	**0**	**0**	**0**	**0**	0	**1**	**0**	**0**	**1**	0	0	0	0	1	0	0	1
0	**0**	**0**	**0**	**0**	1	**0**	**0**	**0**	**1**	0	0	0	0	0	0	0	1
1	**0**	**0**	**0**	**1**	0	**0**	**0**	**0**	**0**	0	0	0	1	0	0	0	1
1	**0**	**0**	**0**	**1**	1	**0**	**0**	**1**	**0**	0	0	0	1	0	0	1	1
2	**0**	**0**	**1**	**0**	0	**0**	**0**	**0**	**1**	0	0	1	0	0	0	1	1
2	**0**	**0**	**1**	**0**	1	**0**	**0**	**1**	**1**	0	0	1	0	0	0	0	1
3	**0**	**0**	**1**	**1**	0	**0**	**0**	**1**	**0**	0	0	1	1	0	0	0	1
3	**0**	**0**	**1**	**1**	1	**0**	**1**	**0**	**0**	0	0	1	1	0	1	1	1
4	**0**	**1**	**0**	**0**	0	**0**	**0**	**1**	**1**	0	1	0	0	0	1	1	1
4	**0**	**1**	**0**	**0**	1	**0**	**1**	**0**	**1**	0	1	0	0	0	0	0	1
5	**0**	**1**	**0**	**1**	0	**0**	**1**	**0**	**0**	0	1	0	1	0	0	0	1
5	**0**	**1**	**0**	**1**	1	**0**	**1**	**1**	**0**	0	1	0	1	0	0	1	1
6	**0**	**1**	**1**	**0**	0	**0**	**1**	**0**	**1**	0	1	1	0	0	0	1	1
6	**0**	**1**	**1**	**0**	1	**0**	**1**	**1**	**1**	0	1	1	0	0	0	0	1
7	**0**	**1**	**1**	**1**	0	**0**	**1**	**1**	**0**	0	1	1	1	0	0	0	1
7	**0**	**1**	**1**	**1**	1	**1**	**0**	**0**	**0**	0	1	1	1	1	1	1	1
8	**1**	**0**	**0**	**0**	0	**0**	**1**	**1**	**1**	1	0	0	0	1	1	1	1
8	**1**	**0**	**0**	**0**	1	**1**	**0**	**0**	**1**	1	0	0	0	0	0	0	1
9	**1**	**0**	**0**	**1**	0	**1**	**0**	**0**	**0**	1	0	0	1	0	0	0	1
9	**1**	**0**	**0**	**1**	1	**0**	**0**	**0**	**0**	1	0	0	1	1	0	0	1

Assuming values between 9-15, result in don’t care values. We can complete K-map minimizations for inputs and outputs

TA

DC/BA	00'	01'	10'	11'
00'	1	1	1	1
01'	1	1	1	1
10'	1	1	1	1
11'	1	1	1	1

TB

DC/BA	00'	01'	10'	11'
00'	0	(x)	(x)	!(x)
01'	!(x)	(x)	(x)	!(x)
10'	!(x)	0	0	!(x)
11'	!(x)	0	0	!(x)

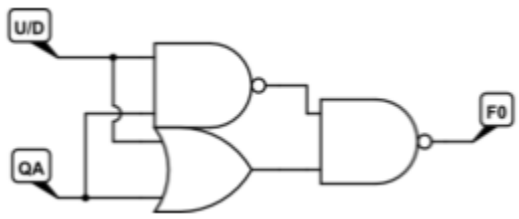
TC

DC/BA	00'	01'	10'	11'
00'	0	0	(x)	0
01'	!(x)	0	(x)	0
10'	!(x)	0	(x)	!(x)
11'	!(x)	0	(x)	!(x)

TD

DC/BA	00'	01'	10'	11'
00'	!(x)	0	0	0
01'	0	0	(x)	0
10'	!(x)	(x)	(x)	!(x)
11'	!(x)	(x)	(x)	!(x)

We choose random values for the don't care and tried to keep to maximize the group sizes by using the three different states, !x , x, and 0. We used x and !x because the 5 variable k-map was combines such that a 4 variable k-map was used regardless of the x input. As shown above the TA=1, the simplified circuit below can be used to determine TB, TC, and TD.



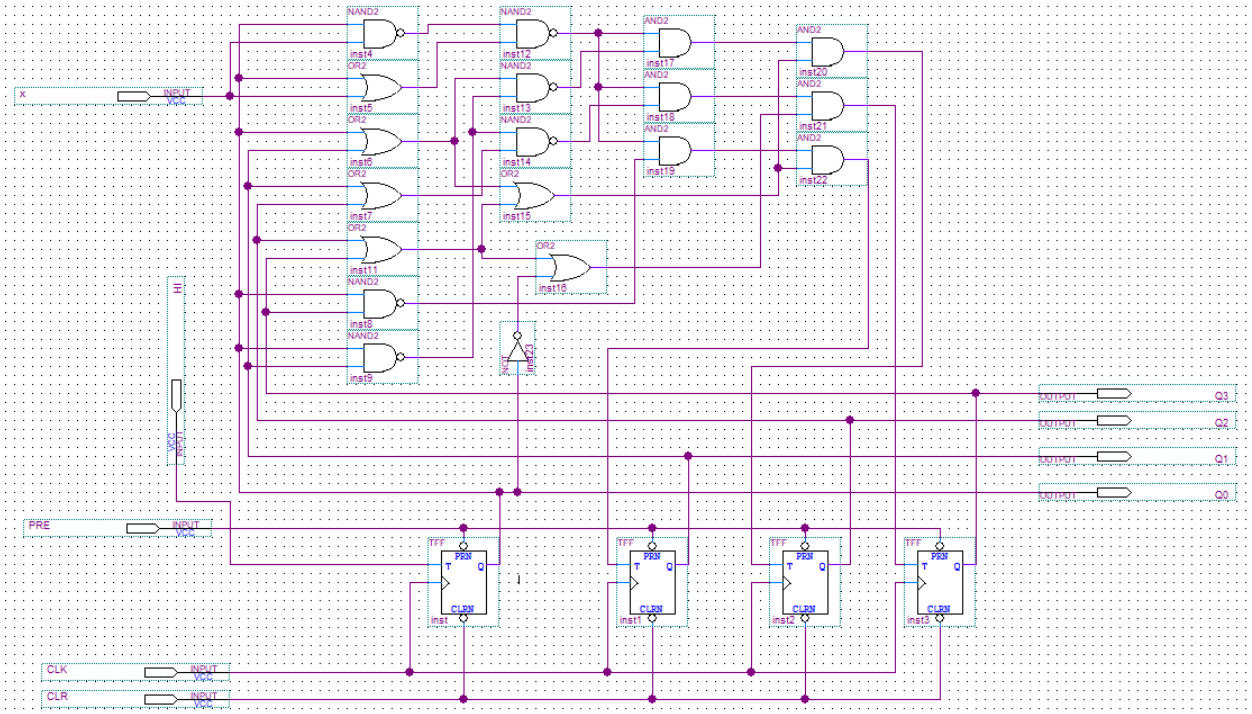
F0

DC/BA	00'	01'	10'	11'
00'	!(x)	(x)	(x)	!(x)
01'	!(x)	(x)	(x)	!(x)
10'	!(x)	(x)	(x)	!(x)
11'	!(x)	(x)	(x)	!(x)

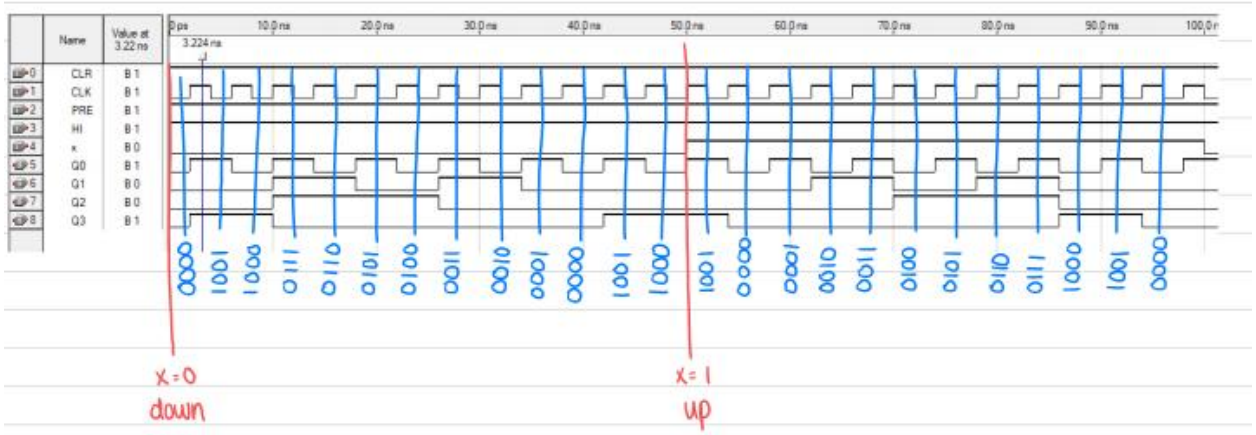
F0 = x * A + !x*A

Using the characteristics table and the truth table generated we built the circuit directly in Quartus.

Circuit:



Simulation Results



The simulation results match the truth table outputs for the 4 outputs.