ECEN 429: Introduction to Digital Systems Design Laboratory

North Carolina Agricultural and Technical State University

Department of Electrical and Computer Engineering

Ian Parker (Reporter)

Tayanna Lee (Lab Partner)

February, 28, 2019

Lab #5

Introduction:

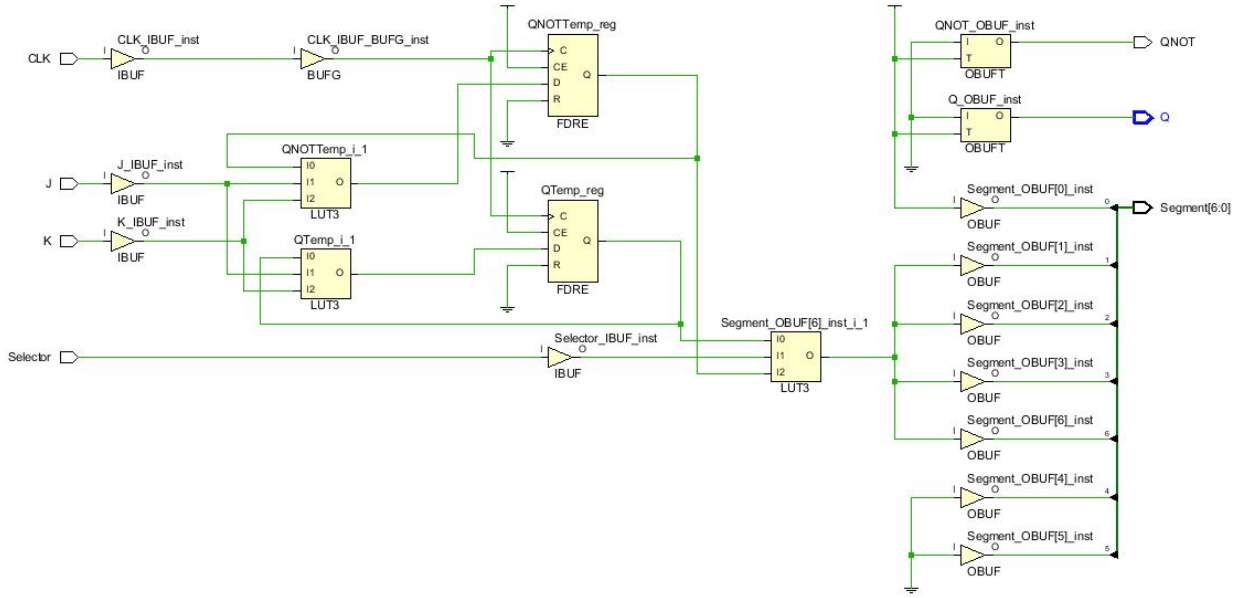
For lab 5, our focus turned to latches and flip flops. Beginning with J-K Flip Flops, which are the most versatile. Followed by D Flip Flops, which are used to store value from the data line. And finally, back to J-K Flip Flops-enhancing its functions allowing it to reset and transition from enable to disable.

Part 1: Create a J-K Flip Flop and demonstrate it using 7-Segment Display as an output.

Truth Table

J	**K**	**Selector**	**Q**	**QNOT**	**7-Seg Display**
0	0	0	Q	QNOT	QNOT
0	0	1	Q	QNOT	Q
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	Q	QNOT	Q
1	1	1	Q	QNOT	QNOT

Schematic



Pin Assignment

Segment	
6	U7
5	W6
4	U8
3	V8
2	U5
1	V5
0	W7

INPUTS	
CLK	W5
J	R2

K	T1
SELECTOR	U1

Result

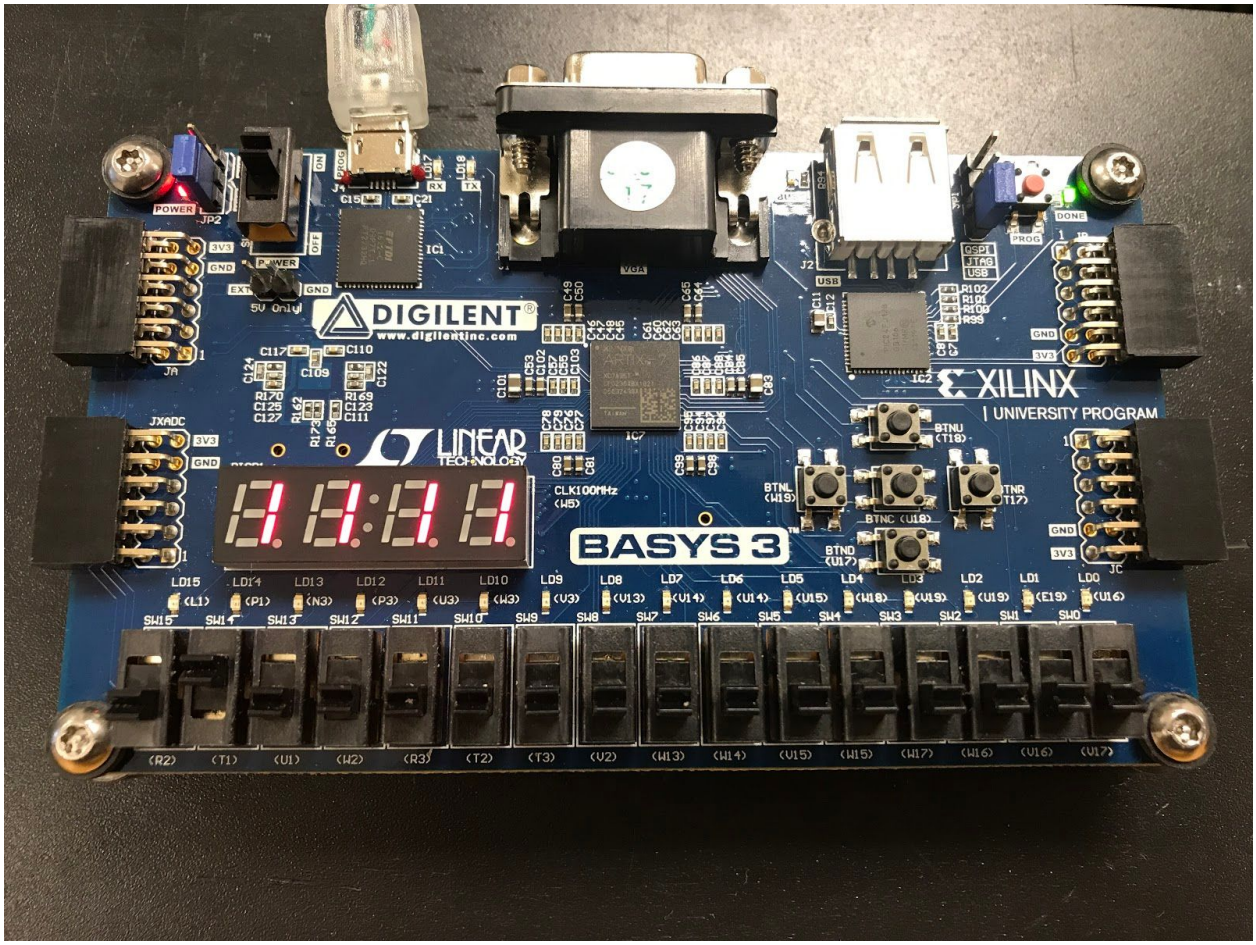


Figure 1.1) J = 0, K=1, Selector = 0, Segment Display = 1

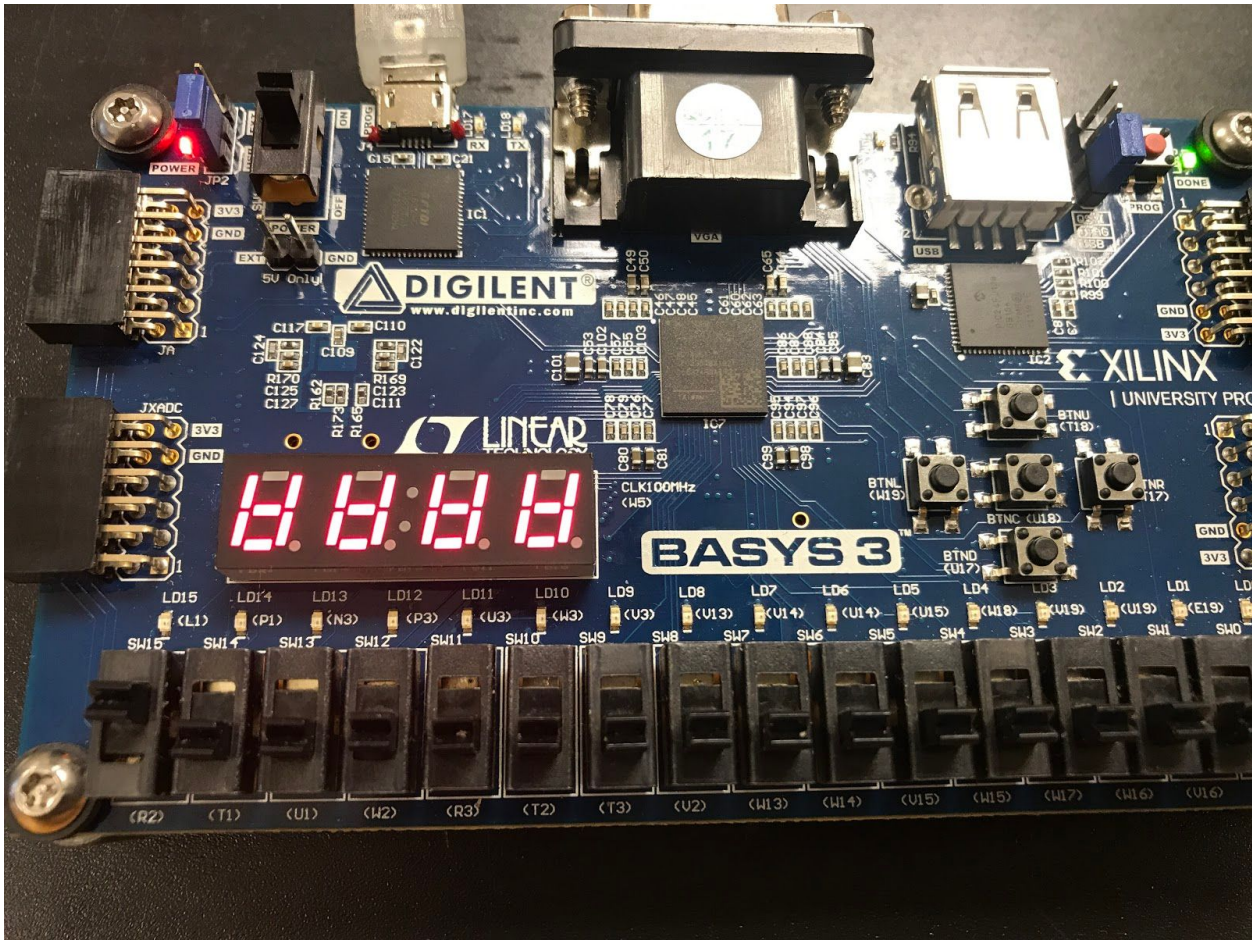


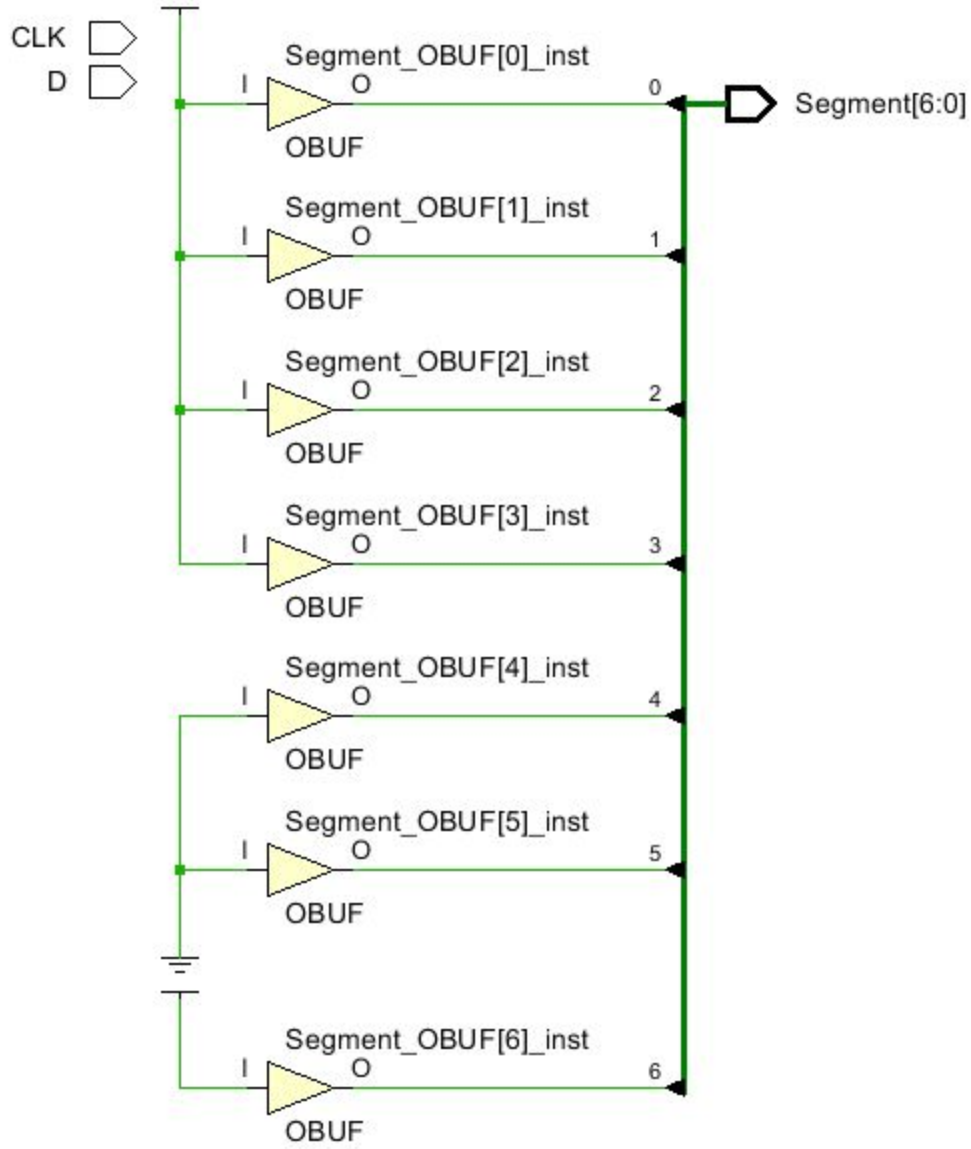
Figure 1.2) J = 1, K= 0, Selector = 0, Segment Display = 0

Part 2: Create a D Flip-Flop Using S-R Latch and Inverter as Components

Truth Table

D	CLK	Q	QNOT	**7-Seg Display**	STATE
?	0	Q	QNOT	QNOT	HOLD
0	1	0	1	0	RESET
1	1	1	0	1	SET

Schematic


CLK
D
Segment_OBUF[0]_inst
I
O
0
OBUF
Segment[6:0]
Segment_OBUF[1]_inst
I
O
1
OBUF
Segment_OBUF[2]_inst
I
O
2
OBUF
Segment_OBUF[3]_inst
I
O
3
OBUF
Segment_OBUF[4]_inst
I
O
4
OBUF
Segment_OBUF[5]_inst
I
O
5
OBUF
Segment_OBUF[6]_inst
I
O
6
OBUF

Pin Assignment

Segment	
6	U7
5	W6
4	U8
3	V8
2	U5
1	V5
0	W7

INPUTS	
CLK	W5
D	R2

Result

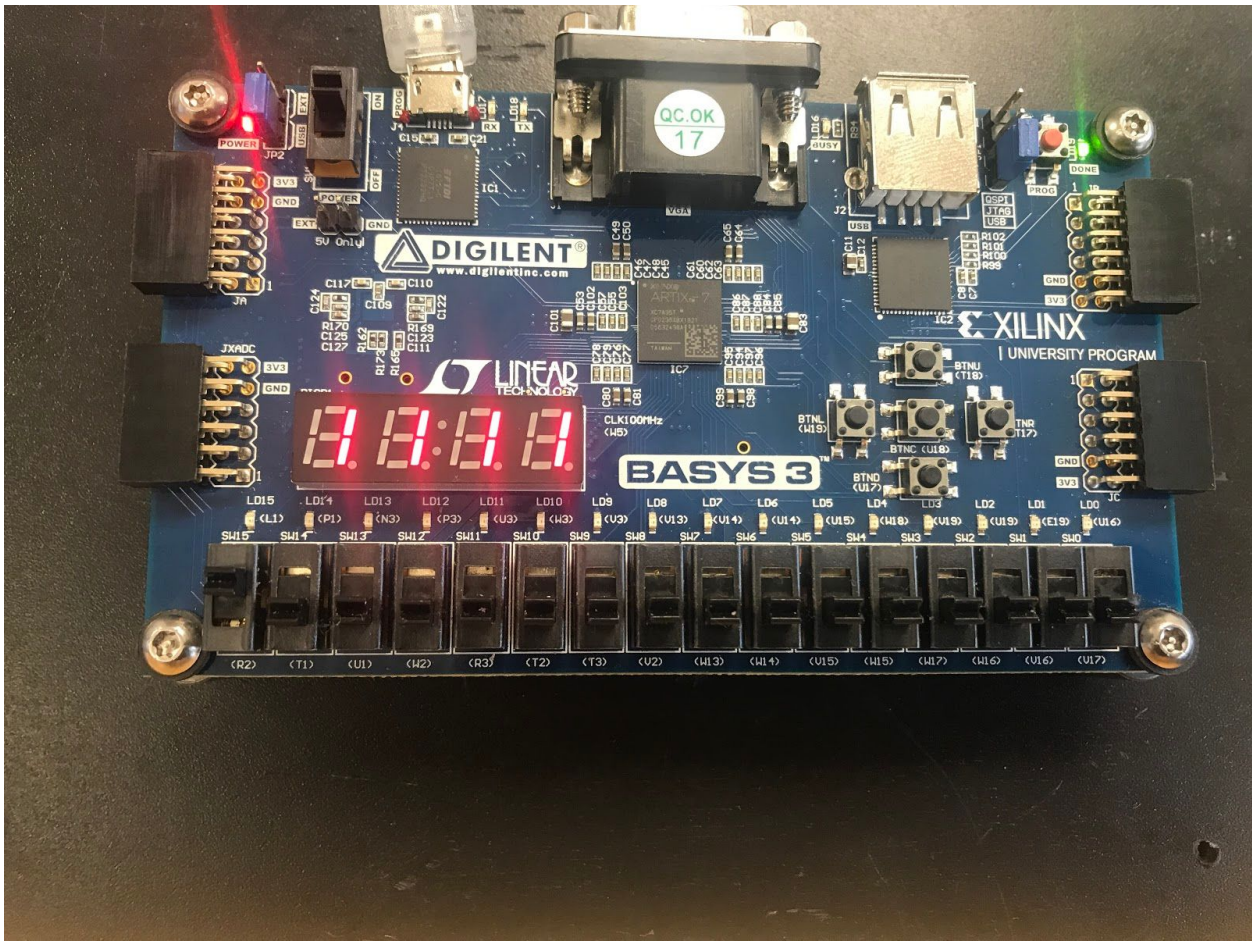


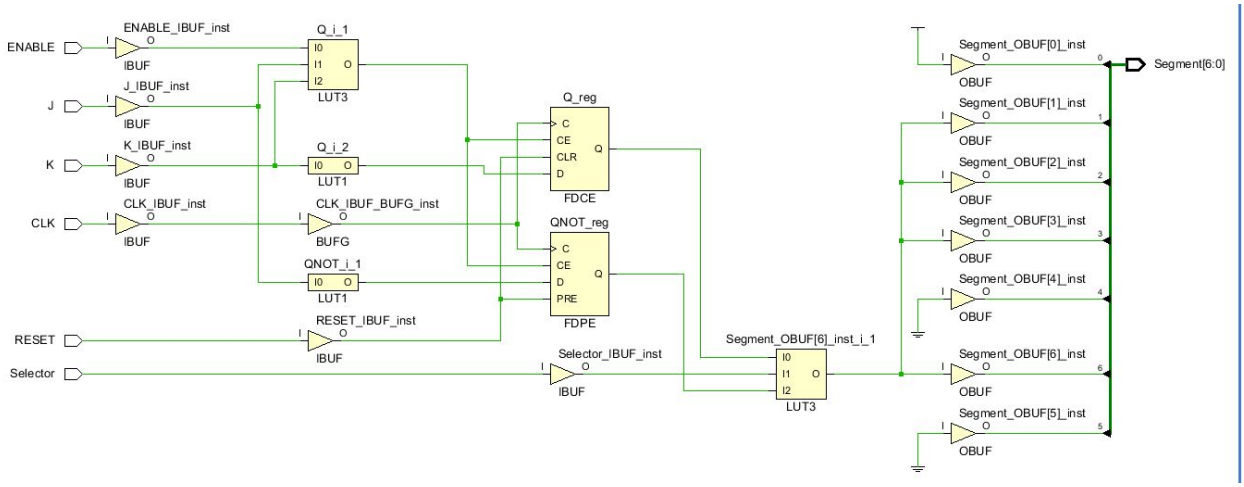
Figure 2.1) D= 1, CLK = 1.

Part 3: J-K Flip FLop with a Enable and Reset feature added to it

Truth Table

J	K	ENABLE	RESET		Selector 1	Selector 0
DC	**DC**	**DC**	1		0	1
DC	**DC**	0	0		Q	QNOT
0	0	1	0		Q	QNOT
0	1	1	0		0	1
1	0	1	0		1	0
1	1	1	0		**TOGGLE**	

Schematic



Pin Assignment

Segment Output	**PIN**
6	U7
5	W6
4	U8

3	V8
2	U5
1	V5
0	W7

INPUTS	**PIN**
CLK	W5
J	R2
K	T1
ENABLE	W13
RESET	W14
SELECTOR	V17

Result

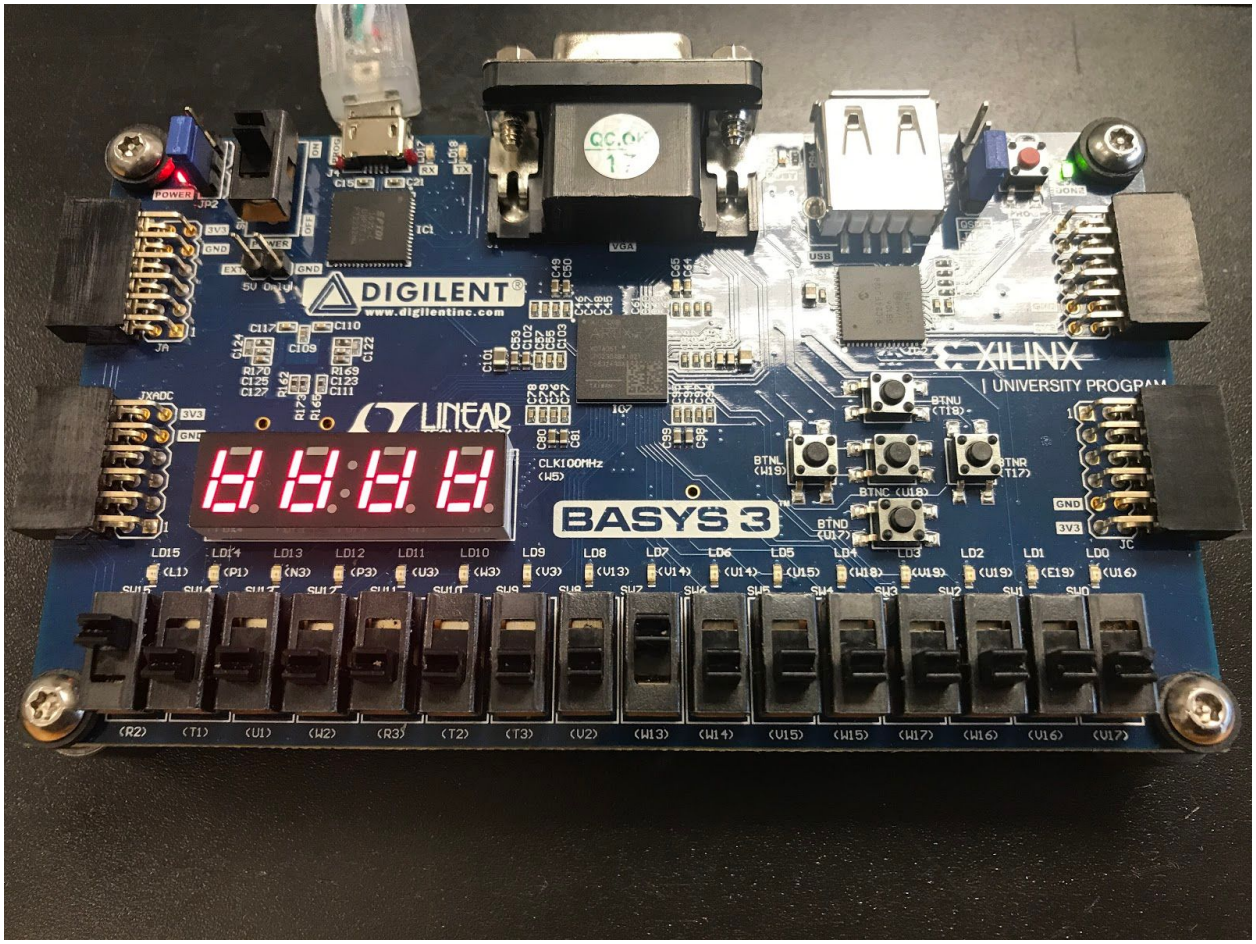


Figure 3.1) J = 1, K = 0, ENABLE = 1, RESET = 0, Selector = 0:Thus Segment Output = 0

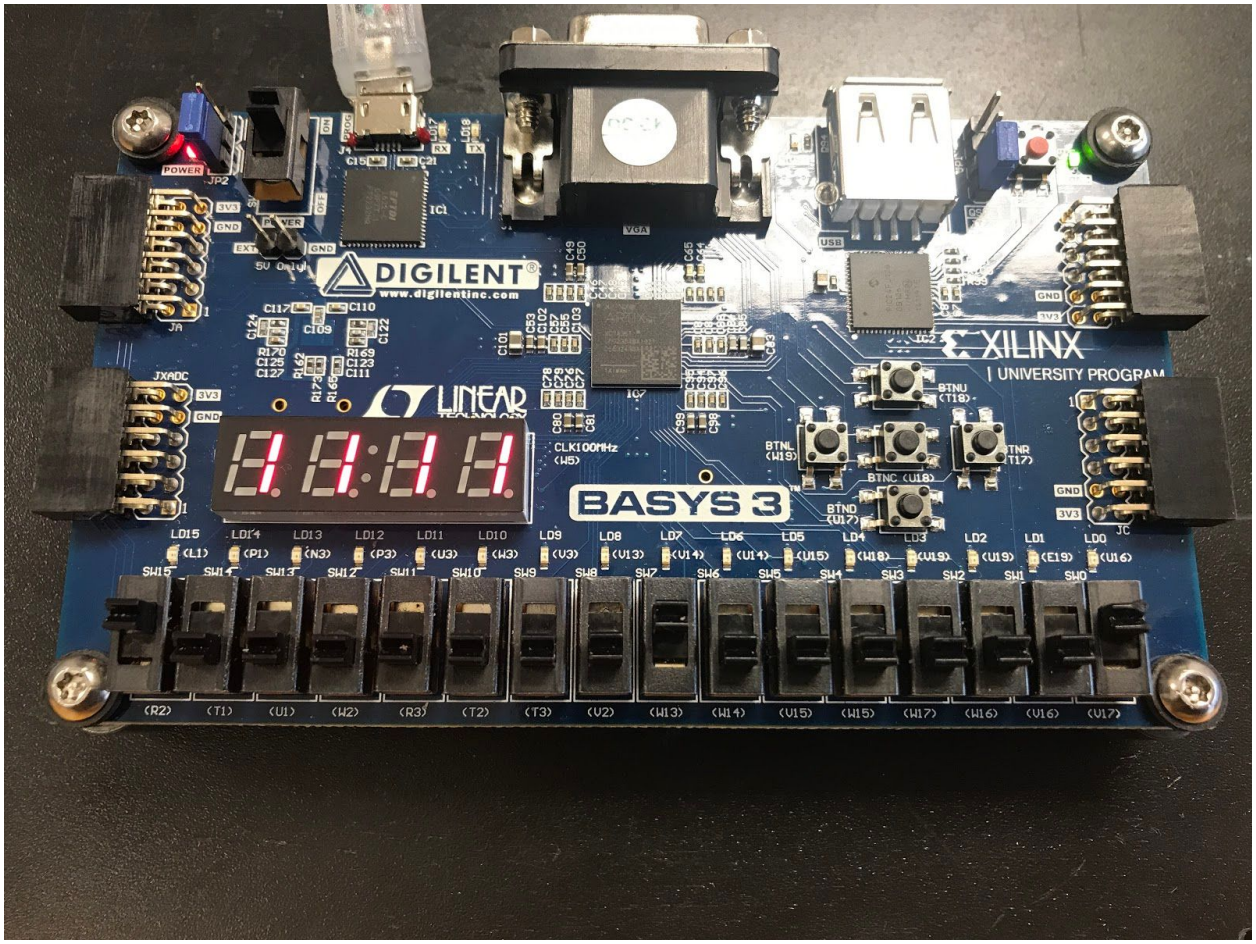


Figure 3.1) J = 1, K = 0, ENABLE = 1, RESET = 0, Selector = 1:Thus Segment Output = 1

Conclusion:

To conclude, we became a little comfortable with coding J-K Flip Flops and D Flip Flops in VHDL. We were able to practice making particular modifications to the flip flops and how those changes made a significant difference to the truth table or how the FF moves from state to state.

Appendices:

Part 1 VHDL CODE : JK FLIP FLOP

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Lab5Part1_JKFlipFlop_SourceFile is
    Port ( J,K,CLK,Selector : in STD_LOGIC;
           Q,QNOT : out STD_LOGIC;
           Segment : out STD_LOGIC_VECTOR(6 downto 0));
end Lab5Part1_JKFlipFlop_SourceFile;

architecture Behavioral of Lab5Part1_JKFlipFlop_SourceFile is
    signal QTemp, QNOTTemp, OutputTemp : STD_LOGIC;

        component TwotoOneMUX  --2:1 MUX
                port (A,B,S0 : in STD_LOGIC;
                        Z: out STD_LOGIC);
        end component;

        begin

            MUX : TwotoOneMUX PORT MAP (     A => QTemp,
                                             B => QNOTTemp,
                                             S0 => Selector,
                                             Z => OutputTemp);
            process(CLK)
               begin
                  if(rising_edge(CLK)) then --(clk'event and (clk = '1'))
                     if((J = '0') AND (K = '0')) then
                        --HOLD
                        QTemp <= QTemp;
                        QNOTTemp <= QNOTTemp;
                     elsif((J = '0') AND (K = '1')) then
                        QTemp <= '0';
                        QNOTTemp <= '1';
                     elsif((J = '1') AND (K = '0')) then
                        QTemp <= '1';
                        QNOTTemp <= '0';
                     else
                        QTemp <= NOT J;
                        QNOTTemp <= NOT K;
```

```
            end if;
        end if;
    end process;

    process(OutputTemp)
      begin
        if(OutputTemp = '0') then
            Segment <= "0000001";
        elsif(OutputTemp = '1') then
            Segment <= "1001111";
        end if;
    end process;
end Behavioral;
```

Part 1 Constraints

```
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports CLK]
set_property IOSTANDARD LVCMOS33 [get_ports J]
set_property IOSTANDARD LVCMOS33 [get_ports K]
set_property IOSTANDARD LVCMOS33 [get_ports Q]
set_property IOSTANDARD LVCMOS33 [get_ports QNOT]
set_property IOSTANDARD LVCMOS33 [get_ports Selector]
set_property PACKAGE_PIN R2 [get_ports J]
set_property PACKAGE_PIN U1 [get_ports Selector]
set_property PACKAGE_PIN T1 [get_ports K]
set_property PACKAGE_PIN W5 [get_ports CLK]
set_property PACKAGE_PIN V8 [get_ports {Segment[3]}]
set_property PACKAGE_PIN W7 [get_ports {Segment[0]}]
set_property PACKAGE_PIN V5 [get_ports {Segment[1]}]
set_property PACKAGE_PIN U5 [get_ports {Segment[2]}]
```

```
set_property PACKAGE_PIN U8 [get_ports {Segment[4]}]
set_property PACKAGE_PIN W6 [get_ports {Segment[5]}]
set_property PACKAGE_PIN U7 [get_ports {Segment[6]}]


set_property PACKAGE_PIN E19 [get_ports Q]
set_property PACKAGE_PIN U16 [get_ports QNOT]
```

Part 2 VHDL Code: D-Flip Flop

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity DFlipFlop is
    Port ( D, CLK : in STD_LOGIC;
           Segment : out STD_LOGIC_VECTOR(6 downto 0));
end DFlipFlop;

architecture Behavioral of DFlipFlop is
        component SRLatch is
                port(S,R, CLK : in STD_LOGIC;
                  Q, QNOT: out STD_LOGIC);
        end component;

        component INVERTER is
                port(Input : in STD_LOGIC;
                   Output : out STD_LOGIC);
        end component;

    signal DTemp, DNOTTemp : STD_LOGIC;--D DNOT for the D FLip FLop
    signal SRQTemp, SRQNOTTemp : STD_LOGIC; --The Q and QNOT output for the SR
Latch
    signal OutputTemp : STD_LOGIC;--Output
       begin
          LATCH: SRLATCH PORT MAP( D, DNOTTemp, CLK, SRQTemp, SRQNOTTemp);
          INVERT : INVERTER PORT MAP(D, DNOTTemp);
```

```
        process(CLK)
           begin
                if(rising_edge(CLK)) then
                        if (SRQTemp = '1') then
                           Segment <= "1001111";
                        elsif(SRQNOTTemp = '0') then
                           Segment <= "0000001";
                        end if;
                    end if;

        end process;
end Behavioral;
```

Part 3 VHDL Code JK Flip Flop with Reset and Enable

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Lab5Part1_JKFlipFlop_SourceFile is
   Port ( J,K,CLK,Selector : in STD_LOGIC;
          Q,QNOT : out STD_LOGIC;
          Segment : out STD_LOGIC_VECTOR(6 downto 0));
end Lab5Part1_JKFlipFlop_SourceFile;

architecture Behavioral of Lab5Part1_JKFlipFlop_SourceFile is
   signal QTemp, QNOTTemp, OutputTemp : STD_LOGIC;

       component TwotoOneMUX  --2:1 MUX
                port (A,B,S0 : in STD_LOGIC;
                      Z: out STD_LOGIC);
       end component;

       begin
```

```
        MUX : TwotoOneMUX PORT MAP (     A => QTemp,
                                         B => QNOTTemp,
                                         S0 => Selector,
                                         Z => OutputTemp);
        process(CLK)
           begin
              if(rising_edge(CLK)) then --(clk'event and (clk = '1'))
                 if((J = '0') AND (K = '0')) then
                    --HOLD
                    QTemp <= QTemp;
                    QNOTTemp <= QNOTTemp;
                 elsif((J = '0') AND (K = '1')) then
                    QTemp <= '0';
                    QNOTTemp <= '1';
                 elsif((J = '1') AND (K = '0')) then
                    QTemp <= '1';
                    QNOTTemp <= '0';
                 else
                    QTemp <= NOT J;
                    QNOTTemp <= NOT K;
                 end if;
              end if;
        end process;

        process(OutputTemp)
           begin
              if(OutputTemp = '0') then
                 Segment <= "0000001";
              elsif(OutputTemp = '1') then
                 Segment <= "1001111";
              end if;
        end process;
end Behavioral;
```

Part 3 Constraints)

```
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[0]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {Segment[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports CLK]
set_property IOSTANDARD LVCMOS33 [get_ports J]
set_property IOSTANDARD LVCMOS33 [get_ports K]
set_property IOSTANDARD LVCMOS33 [get_ports Q]
set_property IOSTANDARD LVCMOS33 [get_ports QNOT]
set_property IOSTANDARD LVCMOS33 [get_ports Selector]
set_property PACKAGE_PIN R2 [get_ports J]
set_property PACKAGE_PIN U1 [get_ports Selector]
set_property PACKAGE_PIN T1 [get_ports K]
set_property PACKAGE_PIN W5 [get_ports CLK]
set_property PACKAGE_PIN V8 [get_ports {Segment[3]}]
set_property PACKAGE_PIN W7 [get_ports {Segment[0]}]
set_property PACKAGE_PIN V5 [get_ports {Segment[1]}]
set_property PACKAGE_PIN U5 [get_ports {Segment[2]}]
set_property PACKAGE_PIN U8 [get_ports {Segment[4]}]
set_property PACKAGE_PIN W6 [get_ports {Segment[5]}]
set_property PACKAGE_PIN U7 [get_ports {Segment[6]}]


set_property PACKAGE_PIN E19 [get_ports Q]
set_property PACKAGE_PIN U16 [get_ports QNOT]
```