Alexander Martinez

EGCP281

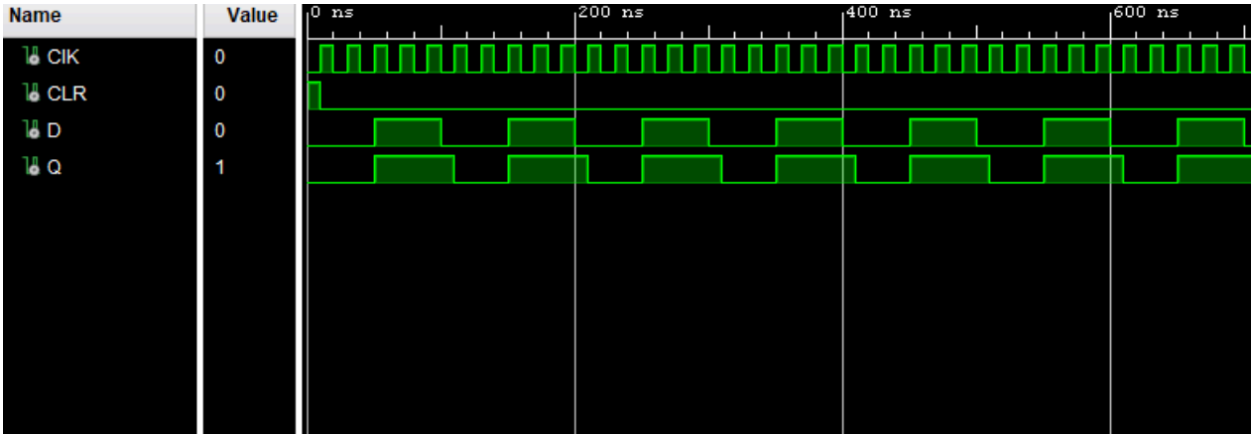
D Latch / Flip Flop

Part A

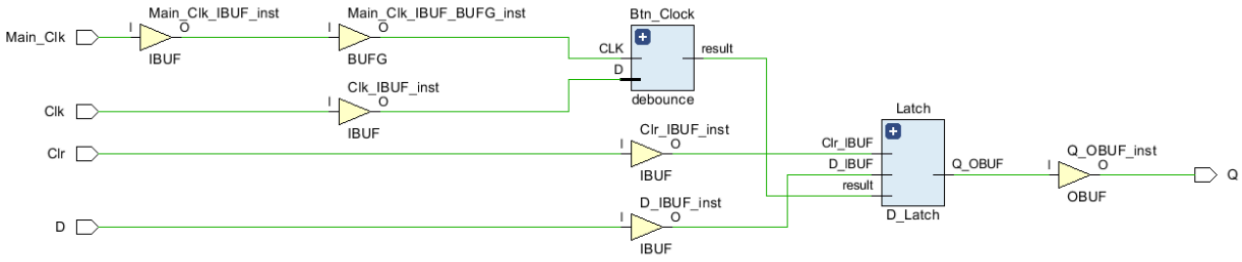
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity Top is
    Port ( Main_Clk, Clk, Clr, D : in STD_LOGIC;
           Q : out STD_LOGIC);
end Top;

architecture Behavioral of Top is
component debounce IS
  GENERIC(
    counter_size  :  INTEGER := 19); --counter size (19 bits gives 10.5ms with 50MHz clock)
  PORT(
    clk     : IN  STD_LOGIC;  --input clock
    button  : IN  STD_LOGIC;  --input signal to be debounced
    result  : OUT STD_LOGIC); --debounced signal
END component;

component D_Latch is
    Port ( Clk, Clr : in STD_LOGIC;
           D : in STD_LOGIC;
           Q : out STD_LOGIC);
end component;

signal N_Clk: Std_logic:='0';

begin
Btn_Clock: debounce port map (Clk => Main_Clk, Button => Clk, Result => N_Clk);
Latch: D_latch port map (Clk => N_Clk, Clr => Clr, D => D, Q=> Q);


```

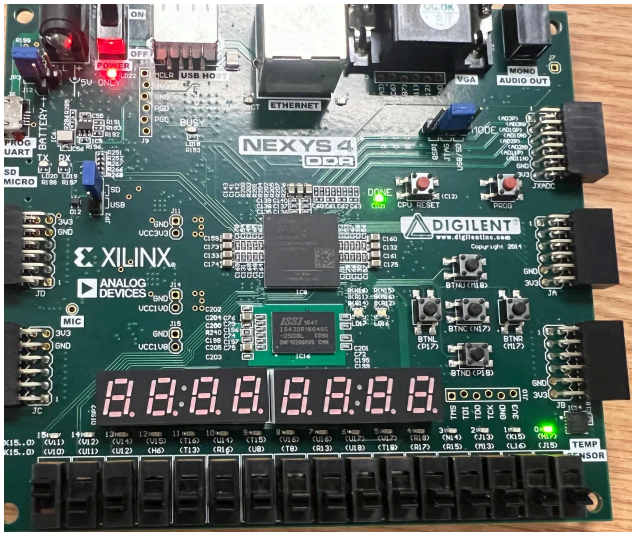
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity D_Latch is
    Port ( Clk, Clr : in STD_LOGIC;
           D : in STD_LOGIC;
           Q : out STD_LOGIC);
end D_Latch;

architecture Behavioral of D_Latch is

begin
-- Write your code here
process(Clr,Clk, D)
begin
if(Clr='1') then
Q <= '0';
elsif(Clk='1') then
Q<= D;
end if;
end process;



--- Code ends here
end Behavioral;
```

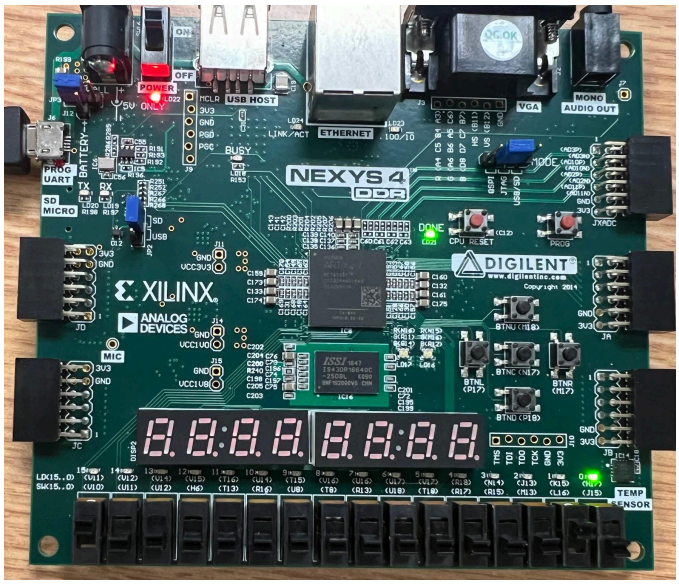
```vhdl
entity D_Latch_TB is
--  Port ( );
end D_Latch_TB;

architecture Behavioral of D_Latch_TB is
component D_Latch is
    Port ( Clk, Clr : in STD_LOGIC;
           D : in STD_LOGIC;
           Q : out STD_LOGIC);
end component;
signal ClK, CLR, D: std_logic:='0';
signal Q: std_logic;
begin

UUT: D_Latch port map (Clk => Clk, CLR=> CLR, D=> D, Q => Q);

D_Proc: process
begin
D <= '0';
wait for 50 ns;
D <= '1';
wait for 50 ns;

end process;

Clr_Proc: process
begin
Clr <= '1';
wait for 10 ns;
Clr <= '0';
wait;
end process;


Clk_Proc: process
begin
Clk <= '0';
wait for 10 ns;
Clk <= '1';
wait for 10 ns;
end process;

end Behavioral;
```

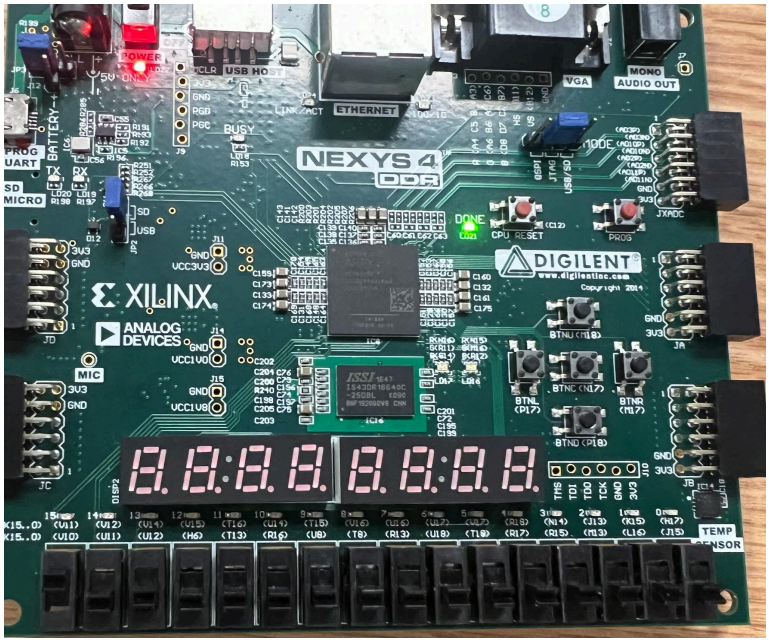


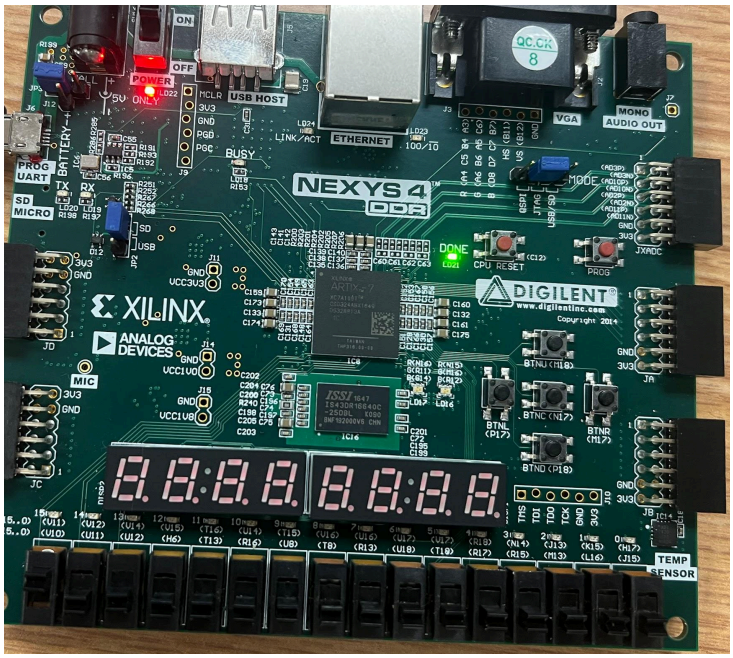

```
set_property IOSTANDARD LVCMOS18 [get_ports Clk]
set_property PACKAGE_PIN J15 [get_ports Clr]
set_property PACKAGE_PIN N17 [get_ports Clk]
set_property PACKAGE_PIN L16 [get_ports D]
set_property PACKAGE_PIN E3 [get_ports Main_Clk]
set_property PACKAGE_PIN H17 [get_ports Q]
set_property IOSTANDARD LVCMOS18 [get_ports Q]
set_property IOSTANDARD LVCMOS18 [get_ports Main_Clk]
set_property IOSTANDARD LVCMOS18 [get_ports D]
set_property IOSTANDARD LVCMOS18 [get_ports Clr]
```

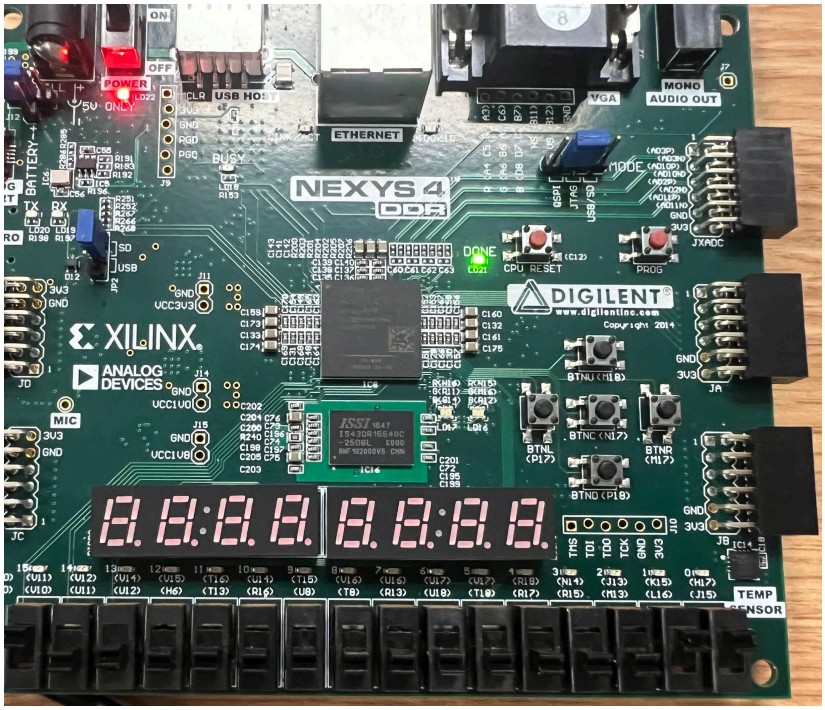



NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
TEMP SENSOR

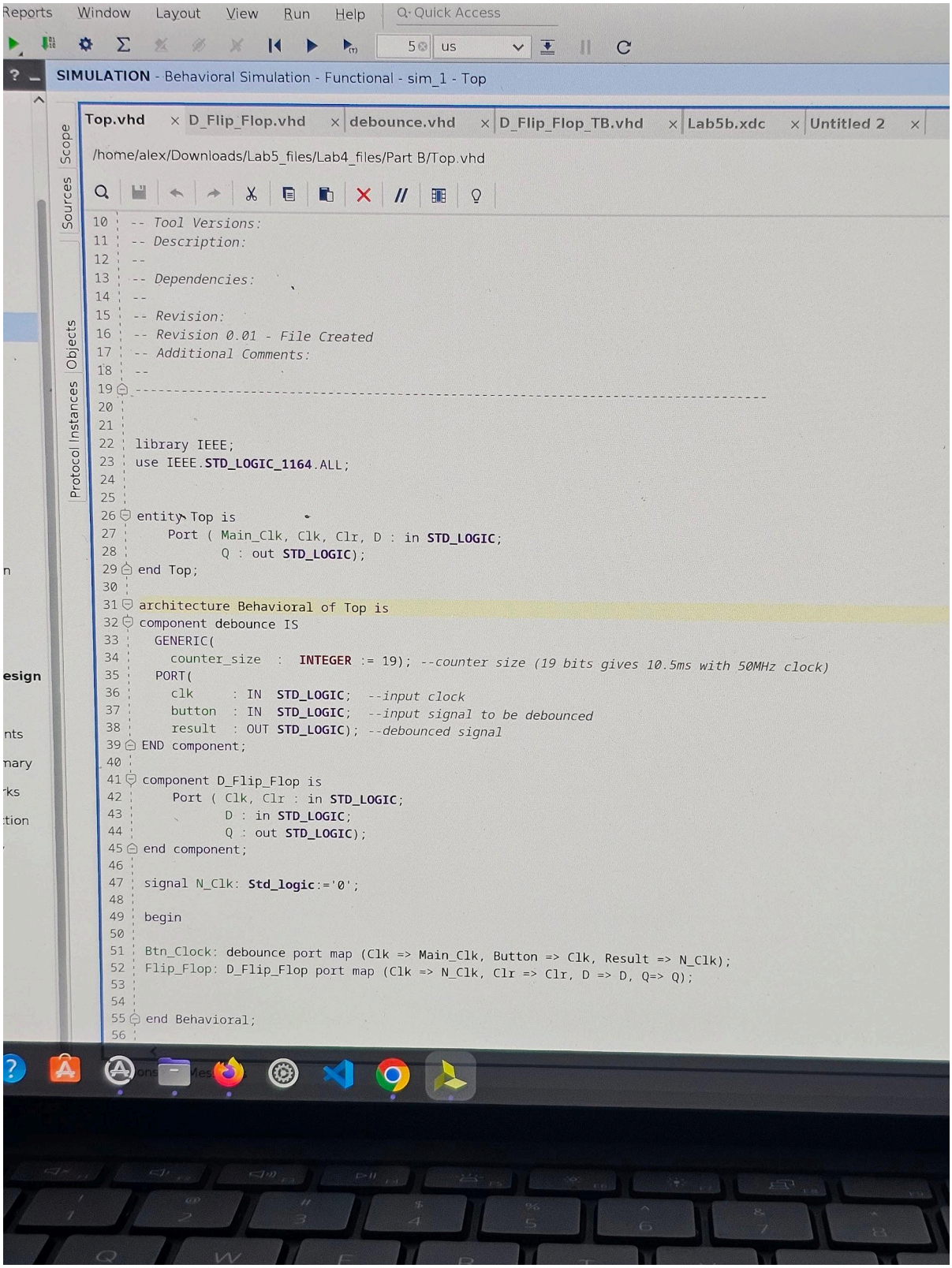

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
TEMP SENSOR

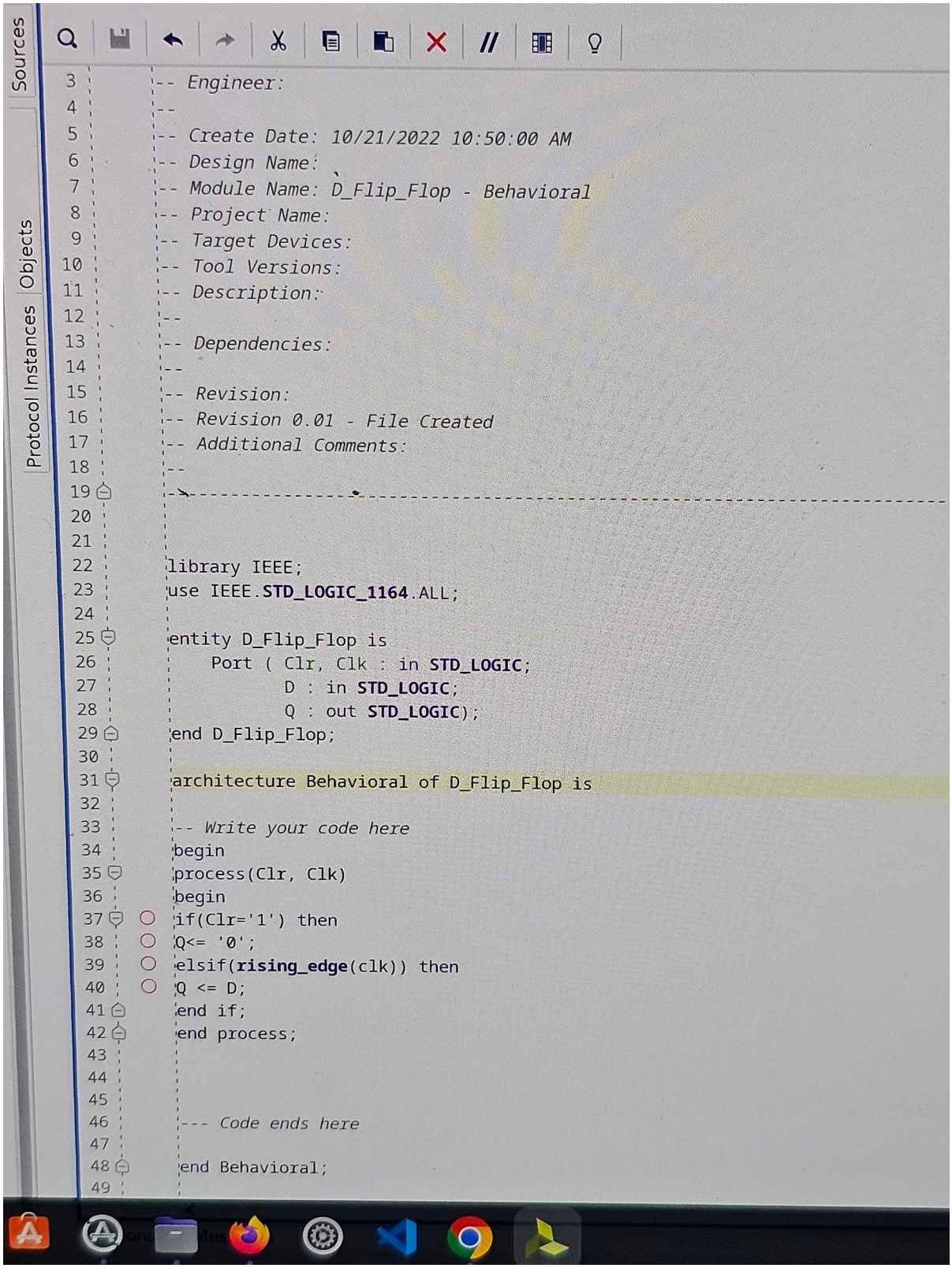

NEXYS 4
DDR
XILINX
DIGILENT
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
PROG
MIC
TEMP SENSOR


NEXYS 4
DDR
XILINX
DIGILENT
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
PROG
MIC
TEMP SENSOR


NEXYS 4
DDR
DIGILENT
www.digilentinc.com
Copyright 2014
XILINX
ANALOG DEVICES
ETHERNET
VGA
MONO AUDIO OUT
USB HOST
POWER
CPU RESET
PROG
DONE
MODE
JXADC
JA
JB
JC
JD
MIC
BTNU (M18)
BTNL (P17)
BTNC (N17)
BTNR (M17)
BTND (P18)
TEMP SENSOR

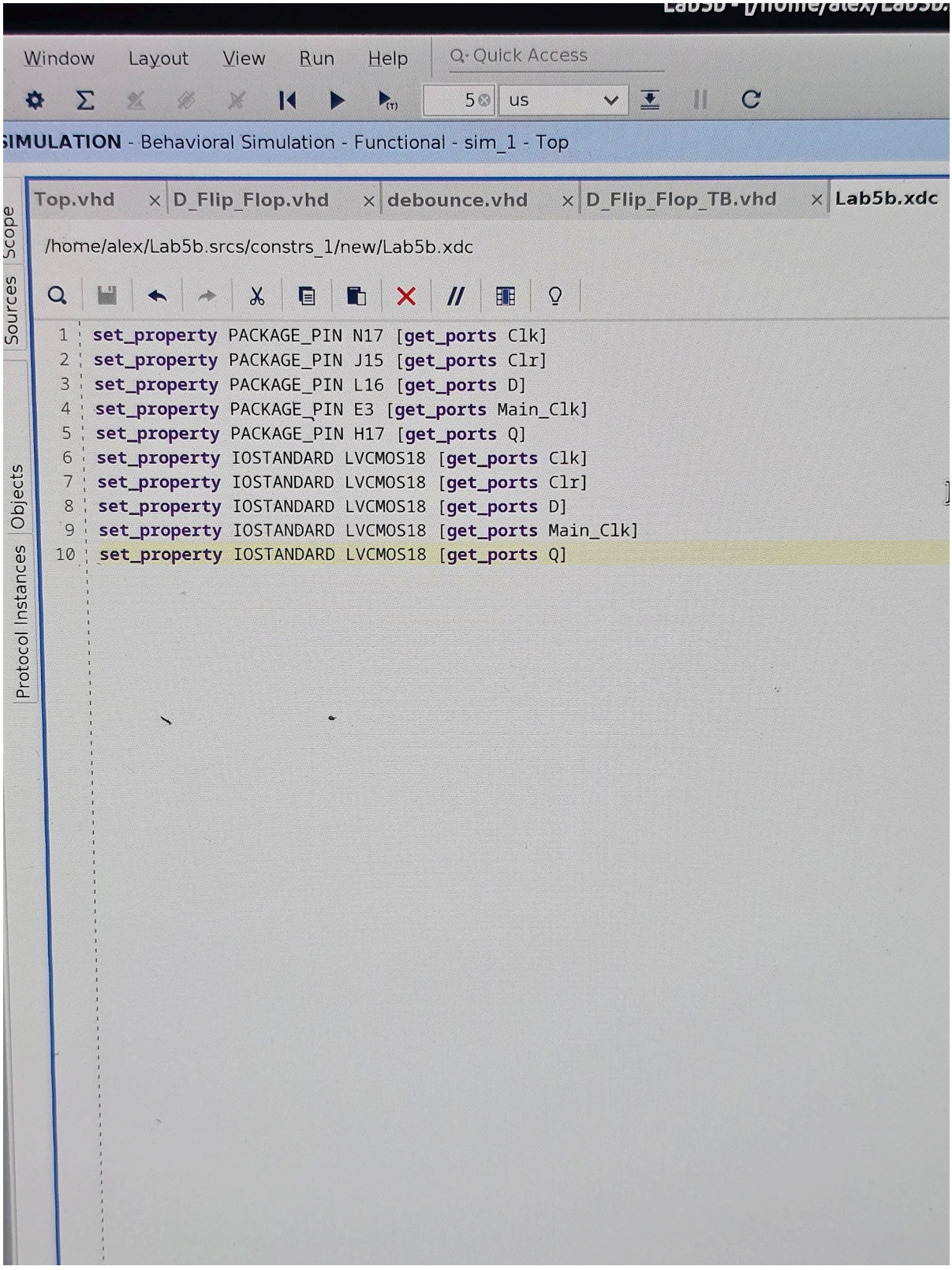
Part B

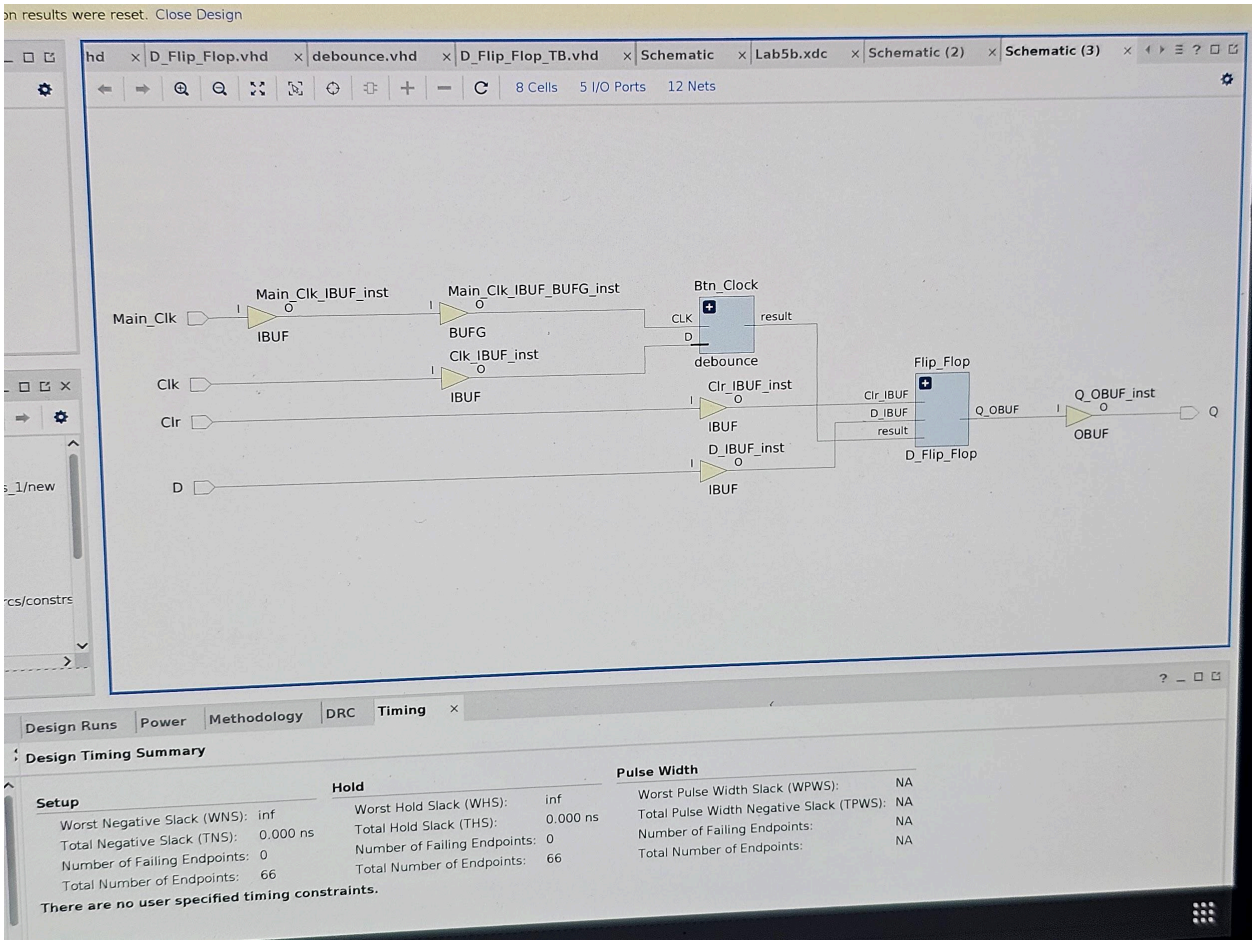

Reports Window Layout View Run Help Quick Access
5 us
SIMULATION - Behavioral Simulation - Functional - sim_1 - Top
Top.vhd D_Flip_Flop.vhd debounce.vhd D_Flip_Flop_TB.vhd Lab5b.xdc Untitled 2
/home/alex/Downloads/Lab5_files/Lab4_files/Part B/Top.vhd
Sources Scope Objects Protocol Instances
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Top is
Port (Main_Clk, Clk, Clr, D : in STD_LOGIC;
Q : out STD_LOGIC);
end Top;
architecture Behavioral of Top is
component debounce IS
GENERIC(
counter_size : INTEGER := 19); --counter size (19 bits gives 10.5ms with 50MHz clock)
PORT(
clk : IN STD_LOGIC; --input clock
button : IN STD_LOGIC; --input signal to be debounced
result : OUT STD_LOGIC); --debounced signal
END component;
component D_Flip_Flop is
Port (Clk, Clr : in STD_LOGIC;
D : in STD_LOGIC;
Q : out STD_LOGIC);
end component;
signal N_Clk: Std_logic:='0';
begin
Btn_Clock: debounce port map (Clk => Main_Clk, Button => Clk, Result => N_Clk);
Flip_Flop: D_Flip_Flop port map (Clk => N_Clk, Clr => Clr, D => D, Q=> Q);
end Behavioral;

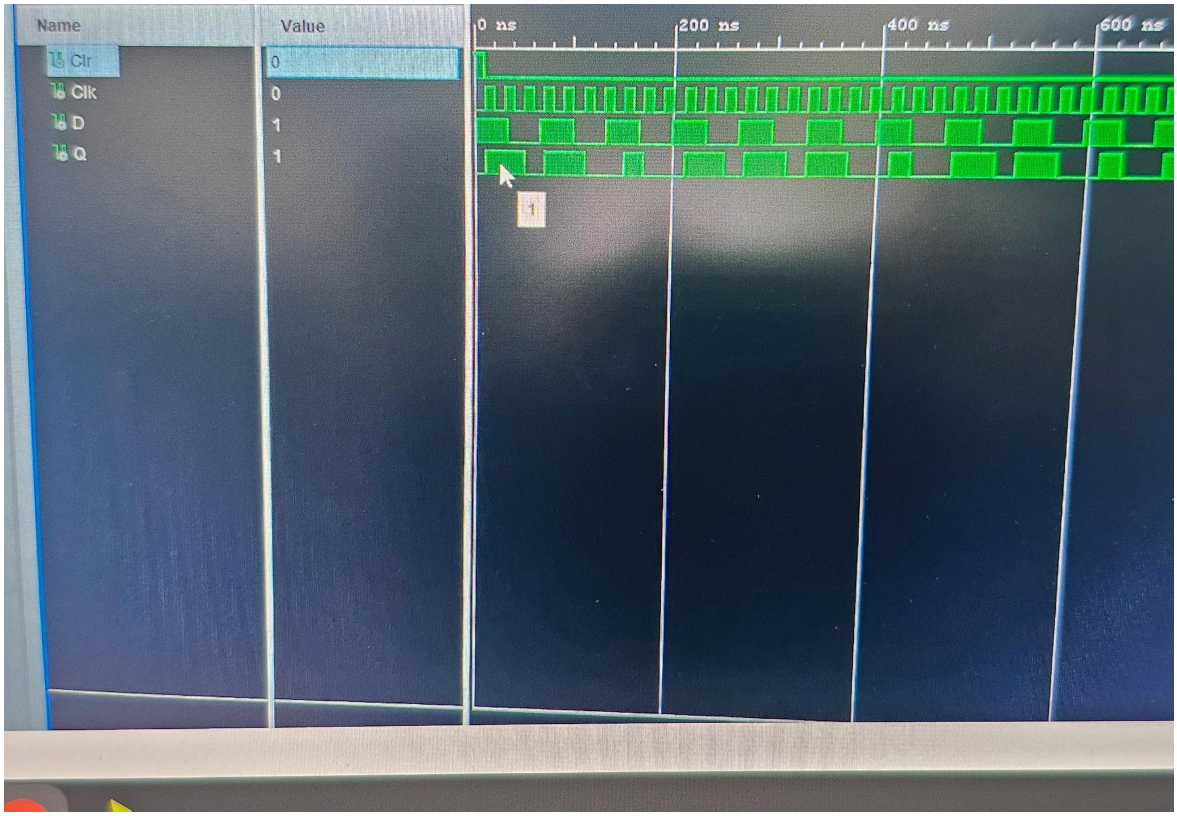

Sources
Objects
Protocol Instances
3 -- Engineer:
4 --
5 -- Create Date: 10/21/2022 10:50:00 AM
6 -- Design Name:
7 -- Module Name: D_Flip_Flop - Behavioral
8 -- Project Name:
9 -- Target Devices:
10 -- Tool Versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 ----------
20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 entity D_Flip_Flop is
26 Port (Clr, Clk : in STD_LOGIC;
27 D : in STD_LOGIC;
28 Q : out STD_LOGIC);
29 end D_Flip_Flop;
30
31 architecture Behavioral of D_Flip_Flop is
32
33 -- Write your code here
34 begin
35 process(Clr, Clk)
36 begin
37 if(Clr='1') then
38 Q<= '0';
39 elsif(rising_edge(clk)) then
40 Q <= D;
41 end if;
42 end process;
43
44
45
46 --- Code ends here
47
48 end Behavioral;
49

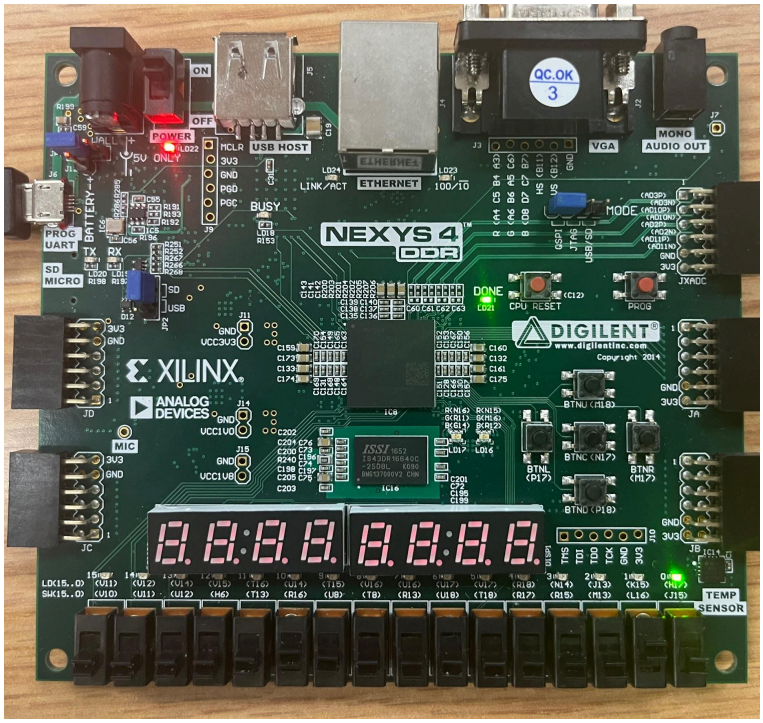
```
--   PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES, ANY CLAIMS
--   BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF),
--   ANY CLAIMS FOR INDEMNITY OR CONTRIBUTION, OR OTHER SIMILAR COSTS.
--
--   Version History
--   Version 1.0 3/26/2012 Scott Larson
--     Initial Public Release
--
----------------------------------------------------------------------------------


LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY debounce IS
  GENERIC(
    counter_size  :  INTEGER := 19); --counter size (19 bits gives 10.5ms with 5
  PORT(
    clk     : IN  STD_LOGIC;  --input clock
    button  : IN  STD_LOGIC;  --input signal to be debounced
    result  : OUT STD_LOGIC); --debounced signal
END debounce;

ARCHITECTURE logic OF debounce IS
  SIGNAL flipflops   : STD_LOGIC_VECTOR(1 DOWNTO 0); --input flip flops
  SIGNAL counter_set : STD_LOGIC;                    --sync reset to zero
  SIGNAL counter_out : STD_LOGIC_VECTOR(counter_size DOWNTO 0) := (OTHERS => '0')
BEGIN

  counter_set <= flipflops(0) xor flipflops(1);   --determine when to start/reset

  PROCESS(clk)
  BEGIN
    IF(clk'EVENT and clk = '1') THEN
      flipflops(0) <= button;
      flipflops(1) <= flipflops(0);
      If(counter_set = '1') THEN                  --reset counter because input i
        counter_out <= (OTHERS => '0');
      ELSIF(counter_out(counter_size) = '0') THEN --stable input time is not yet m
        counter_out <= counter_out + 1;
      ELSE                                        --stable input time is met
        result <= flipflops(1);
      END IF;
    END IF;
  END PROCESS;
END logic;
```

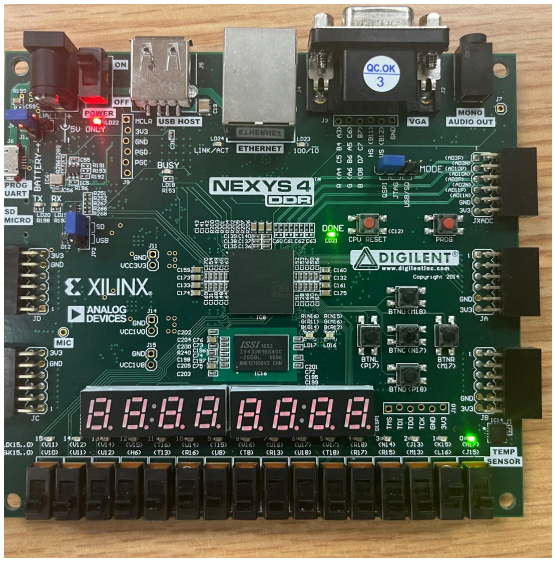
```
32  --use UNISIM.VComponents.all;
33
34  entity D_Flip_Flop_TB is
35      --  Port ( );
36  end D_Flip_Flop_TB;
37
38  architecture Behavioral of D_Flip_Flop_TB is
39
40  component D_Flip_Flop is
41      Port ( Clr, Clk : in STD_LOGIC;
42             D : in STD_LOGIC;
43             Q : out STD_LOGIC);
44  end component;
45
46  signal Clr, Clk, D: std_logic:='0';
47  Signal Q: std_logic;
48  begin
49  UUT: D_Flip_Flop port map(Clr=> Clr, Clk => Clk, D=> D, Q => Q);
50
51  D_Proc: process
52  begin
53  D <= '1';
54  wait for 33 ns;
55  D <= '0';
56  wait for 33 ns;
57
58  end process;
59
60  Clr_Proc: process
61  begin
62  Clr <= '1';
63  wait for 10 ns;
64  Clr <= '0';
65  wait;
66  end process;
67
68
69  Clk_Proc: process
70  begin
71  Clk <= '0';
72  wait for 10 ns;
73  Clk <= '1';
74  wait for 10 ns;
75  end process;
76
77  end Behavioral;
78
```

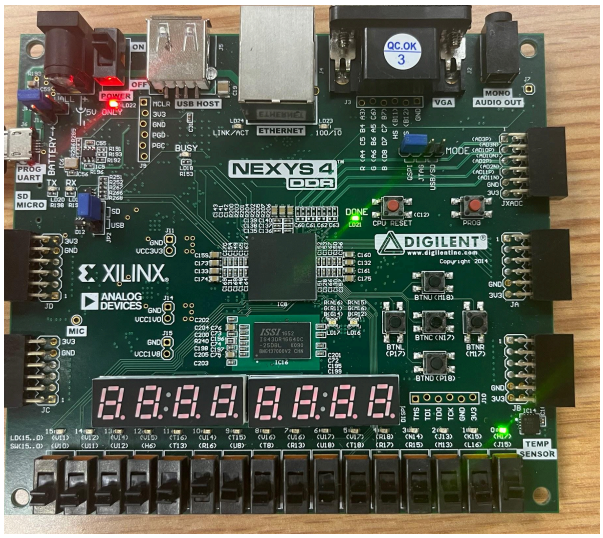

Window
Layout
View
Run
Help
Quick Access
5
us
SIMULATION - Behavioral Simulation - Functional - sim_1 - Top
Top.vhd
D_Flip_Flop.vhd
debounce.vhd
D_Flip_Flop_TB.vhd
Lab5b.xdc
/home/alex/Lab5b.srcs/constrs_1/new/Lab5b.xdc
Scope
Sources
Objects
Protocol Instances
1 set_property PACKAGE_PIN N17 [get_ports Clk]
2 set_property PACKAGE_PIN J15 [get_ports Clr]
3 set_property PACKAGE_PIN L16 [get_ports D]
4 set_property PACKAGE_PIN E3 [get_ports Main_Clk]
5 set_property PACKAGE_PIN H17 [get_ports Q]
6 set_property IOSTANDARD LVCMOS18 [get_ports Clk]
7 set_property IOSTANDARD LVCMOS18 [get_ports Clr]
8 set_property IOSTANDARD LVCMOS18 [get_ports D]
9 set_property IOSTANDARD LVCMOS18 [get_ports Main_Clk]
10 set_property IOSTANDARD LVCMOS18 [get_ports Q]

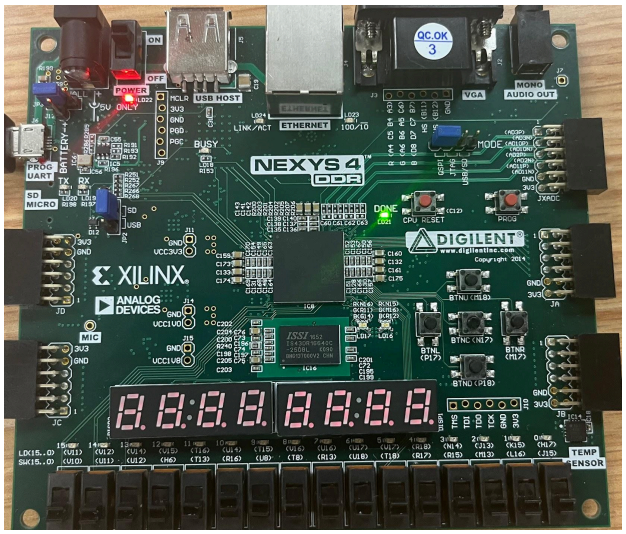

on results were reset. Close Design
hd
D_Flip_Flop.vhd
debounce.vhd
D_Flip_Flop_TB.vhd
Schematic
Lab5b.xdc
Schematic (2)
Schematic (3)
8 Cells
5 I/O Ports
12 Nets
Main_Clk
Main_Clk_IBUF_inst
IBUF
Main_Clk_IBUF_BUFG_inst
BUFG
Btn_Clock
CLK
D
result
debounce
Clk
Clk_IBUF_inst
IBUF
Clr
Clr_IBUF_inst
IBUF
D
D_IBUF_inst
IBUF
Flip_Flop
Clr_IBUF
D_IBUF
result
Q_OBUF
D_Flip_Flop
Q_OBUF_inst
OBUF
Q
Design Runs
Power
Methodology
DRC
Timing
Design Timing Summary
Setup
Worst Negative Slack (WNS): inf
Total Negative Slack (TNS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 66
Hold
Worst Hold Slack (WHS): inf
Total Hold Slack (THS): 0.000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 66
Pulse Width
Worst Pulse Width Slack (WPWS): NA
Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: NA
Total Number of Endpoints: NA
There are no user specified timing constraints.

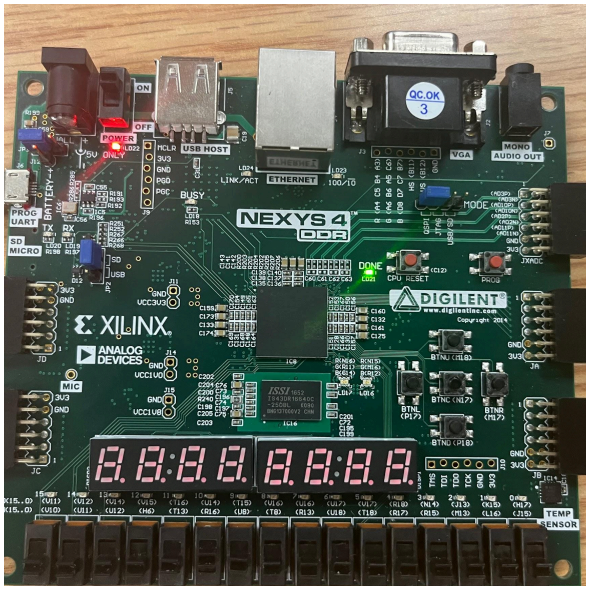

Name
Value
Clr
Clk
D
Q
0
0
1
1
0 ns
200 ns
400 ns
600 ns

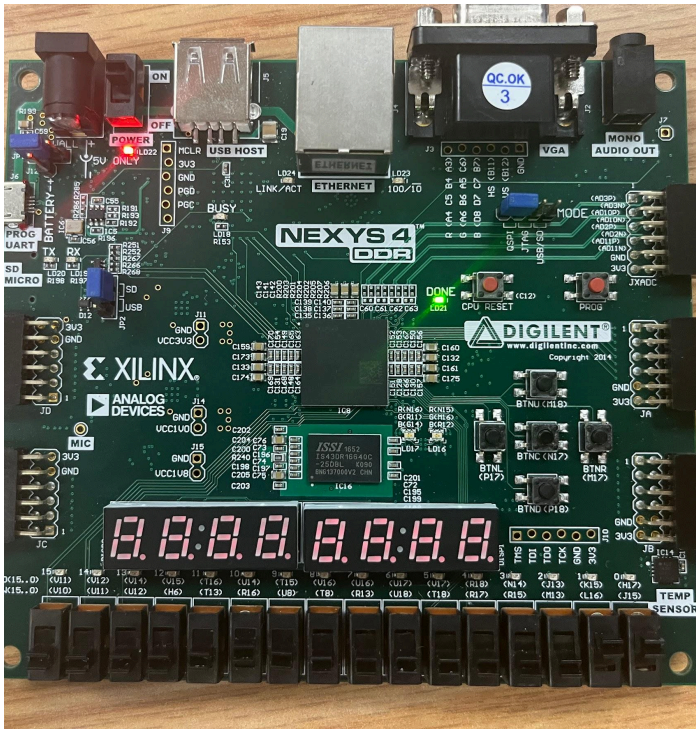

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
USB HOST
ETHERNET
VGA
MONO AUDIO OUT
CPU RESET
PROG
MODE
JXADC
JA
JB
JC
JD
TEMP SENSOR


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
POWER
PROG UART
SD MICRO
MIC
TEMP SENSOR
CPU RESET
PROG
MODE
BUSY
DONE


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
POWER
PROG UART
SD MICRO
MIC
TEMP SENSOR
CPU RESET
PROG
MODE
BUSY
DONE


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
POWER
PROG UART
SD MICRO
MIC
TEMP SENSOR
CPU RESET
PROG
MODE
BUSY
DONE

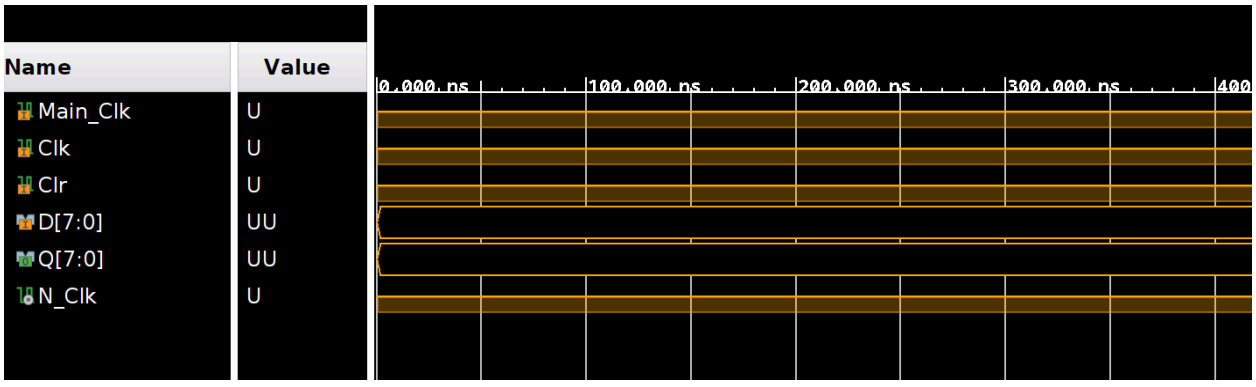

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
CPU RESET
PROG
TEMP SENSOR


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
CPU RESET
PROG
TEMP SENSOR

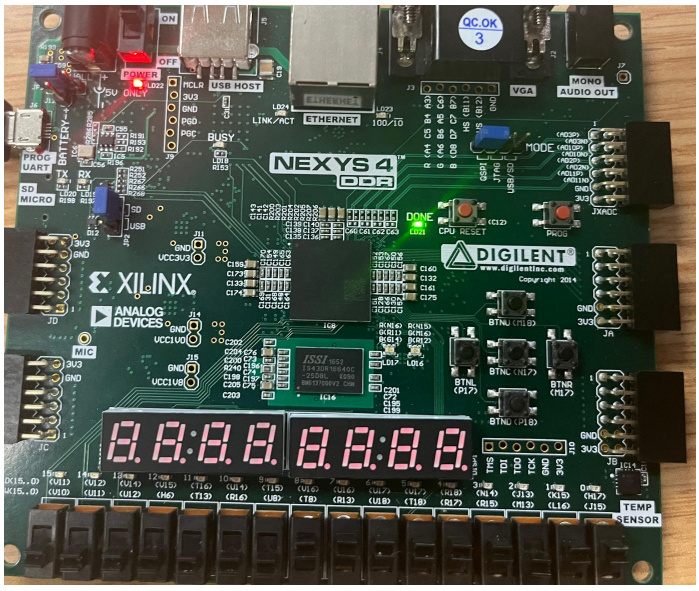
Part c

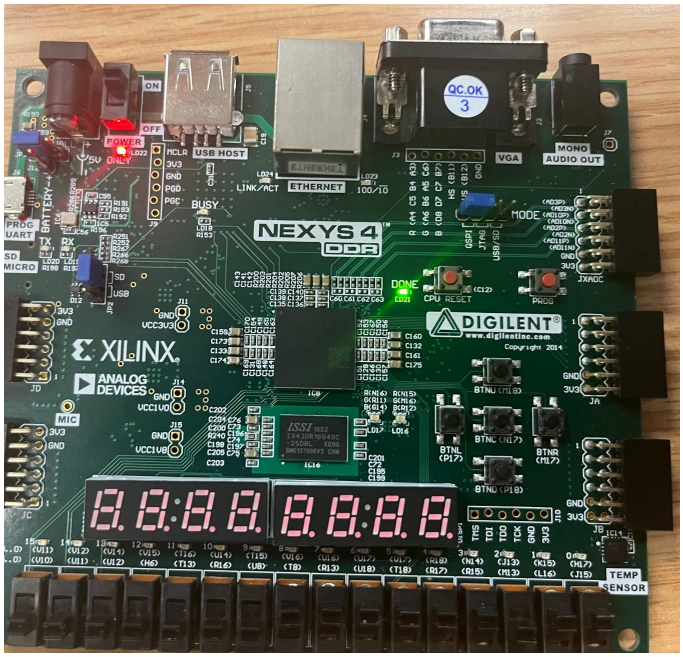
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity Top is
    Port ( Main_Clk, Clk, Clr: in STD_LOGIC;
           D: in std_logic_vector(7 downto 0);
           Q : out std_logic_vector(7 downto 0));
end Top;

architecture Behavioral of Top is
component debounce IS
  GENERIC(
    counter_size  :  INTEGER := 19); --counter size (19 bits gives 10.5ms with 50MHz clock)
  PORT(
    clk     : IN  STD_LOGIC;  --input clock
    button  : IN  STD_LOGIC;  --input signal to be debounced
    result  : OUT STD_LOGIC); --debounced signal
END component;

component Register_8B is
    Port ( Clk, Clr : in STD_LOGIC;
           D : in STD_LOGIC_VECTOR (7 downto 0);
           Q : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal N_Clk: Std_logic:='0';

begin

Btn_Clock: debounce port map (Clk => Main_Clk, Button => Clk, Result => N_Clk);
Reg: Register_8B port map (Clk => N_Clk, Clr => Clr, D => D, Q=> Q);


end Behavioral;
```

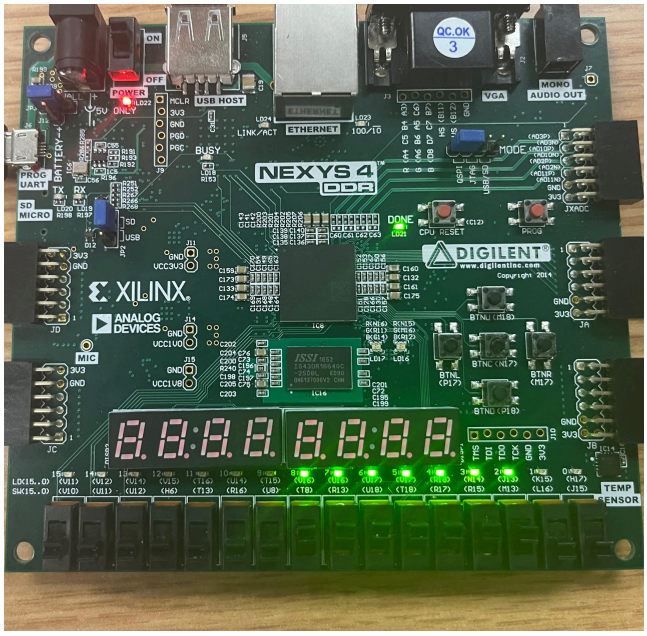
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY debounce IS
  GENERIC(
    counter_size  :  INTEGER := 19); --counter size (19 bits gives 10.5ms with 50MHz clock)
  PORT(
    clk     : IN  STD_LOGIC;  --input clock
    button  : IN  STD_LOGIC;  --input signal to be debounced
    result  : OUT STD_LOGIC); --debounced signal
END debounce;

ARCHITECTURE logic OF debounce IS
  SIGNAL flipflops   : STD_LOGIC_VECTOR(1 DOWNTO 0); --input flip flops
  SIGNAL counter_set : STD_LOGIC;                    --sync reset to zero
  SIGNAL counter_out : STD_LOGIC_VECTOR(counter_size DOWNTO 0) := (OTHERS => '0'); --counter output
BEGIN

  counter_set <= flipflops(0) xor flipflops(1);   --determine when to start/reset counter

  PROCESS(clk)
  BEGIN
    IF(clk'EVENT and clk = '1') THEN
      flipflops(0) <= button;
      flipflops(1) <= flipflops(0);
      If(counter_set = '1') THEN                  --reset counter because input is changing
        counter_out <= (OTHERS => '0');
      ELSIF(counter_out(counter_size) = '0') THEN --stable input time is not yet met
        counter_out <= counter_out + 1;
      ELSE                                        --stable input time is met
        result <= flipflops(1);
      END IF;
    END IF;
  END PROCESS;
END logic;
```

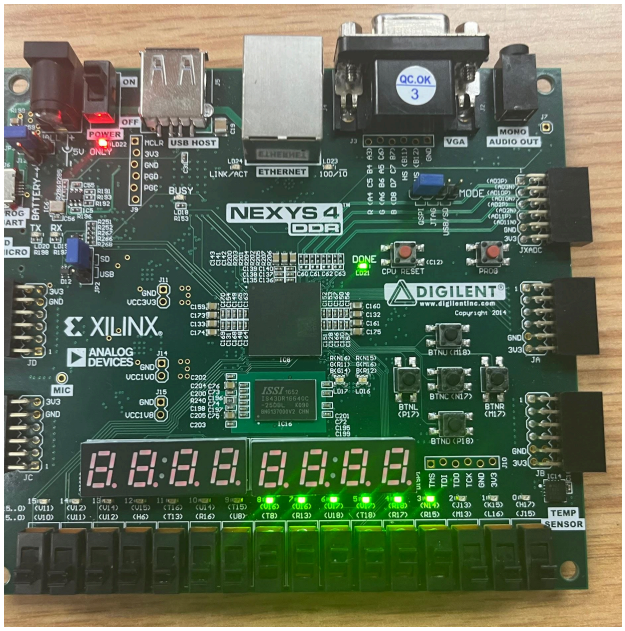


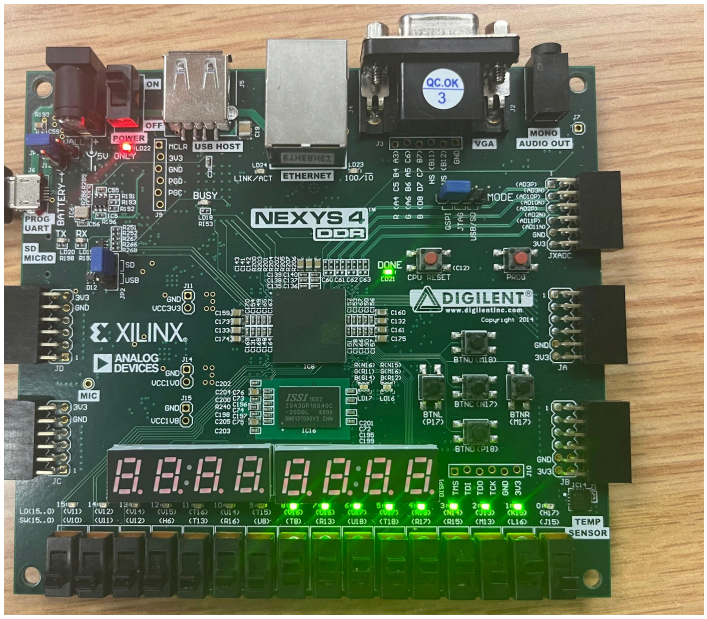
```
set_property PACKAGE_PIN N17 [get_ports Clk]
set_property PACKAGE_PIN J15 [get_ports Clr]
set_property PACKAGE_PIN E3 [get_ports Main_Clk]
set_property IOSTANDARD LVCMOS18 [get_ports Clk]
set_property IOSTANDARD LVCMOS18 [get_ports Clr]
set_property IOSTANDARD LVCMOS18 [get_ports {D[7]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[6]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[5]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[4]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {D[0]}]
set_property IOSTANDARD LVCMOS18 [get_ports Main_Clk]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[7]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[6]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[5]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[4]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[3]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[2]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[1]}]
set_property IOSTANDARD LVCMOS18 [get_ports {Q[0]}]

set_property PACKAGE_PIN L16 [get_ports {D[0]}]
set_property PACKAGE_PIN M13 [get_ports {D[1]}]
set_property PACKAGE_PIN R15 [get_ports {D[2]}]
set_property PACKAGE_PIN R17 [get_ports {D[3]}]
set_property PACKAGE_PIN T18 [get_ports {D[4]}]
set_property PACKAGE_PIN U18 [get_ports {D[5]}]
set_property PACKAGE_PIN R13 [get_ports {D[6]}]
set_property PACKAGE_PIN T8 [get_ports {D[7]}]
set_property PACKAGE_PIN K15 [get_ports {Q[0]}]
set_property PACKAGE_PIN J13 [get_ports {Q[1]}]
set_property PACKAGE_PIN N14 [get_ports {Q[2]}]
set_property PACKAGE_PIN R18 [get_ports {Q[3]}]
set_property PACKAGE_PIN V17 [get_ports {Q[4]}]
set_property PACKAGE_PIN U17 [get_ports {Q[5]}]
set_property PACKAGE_PIN V16 [get_ports {Q[7]}]
set_property PACKAGE_PIN U16 [get_ports {Q[6]}]
```

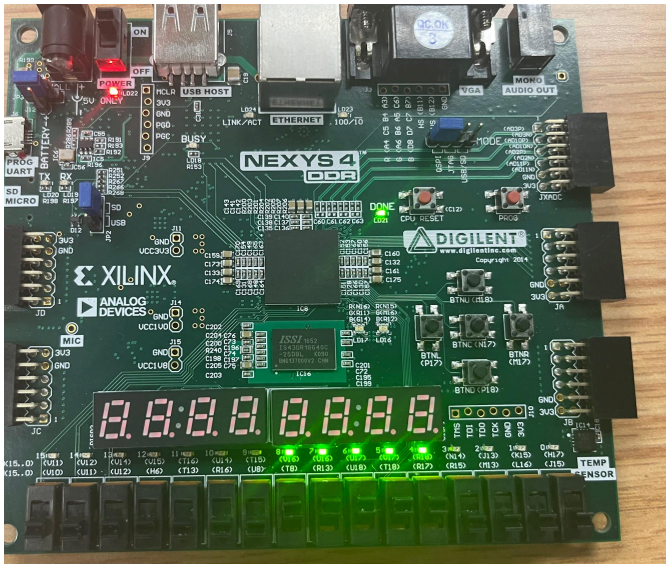


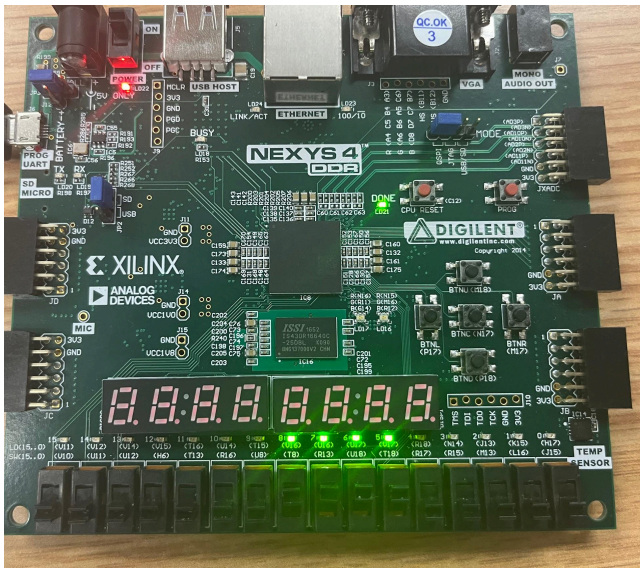

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
POWER
USB HOST
ETHERNET
VGA
MONO AUDIO OUT
CPU RESET
PROG
MODE
TEMP SENSOR

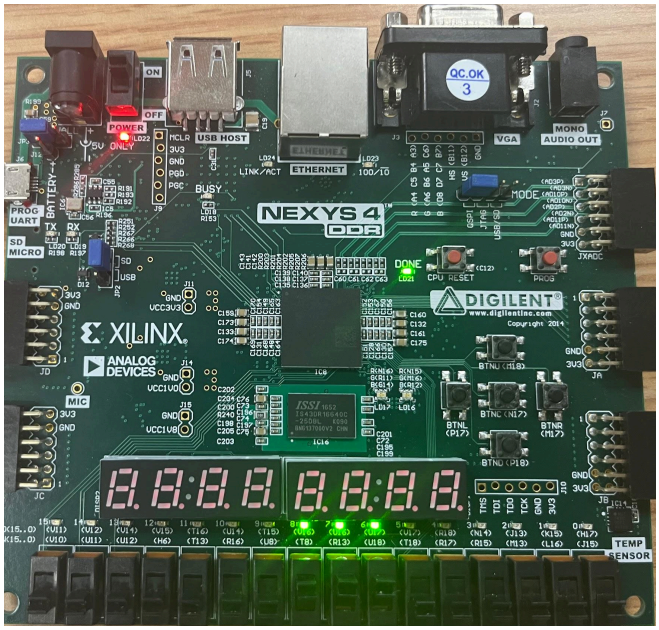

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
POWER
USB HOST
ETHERNET
VGA
MONO AUDIO OUT
CPU RESET
PROG
MODE
TEMP SENSOR

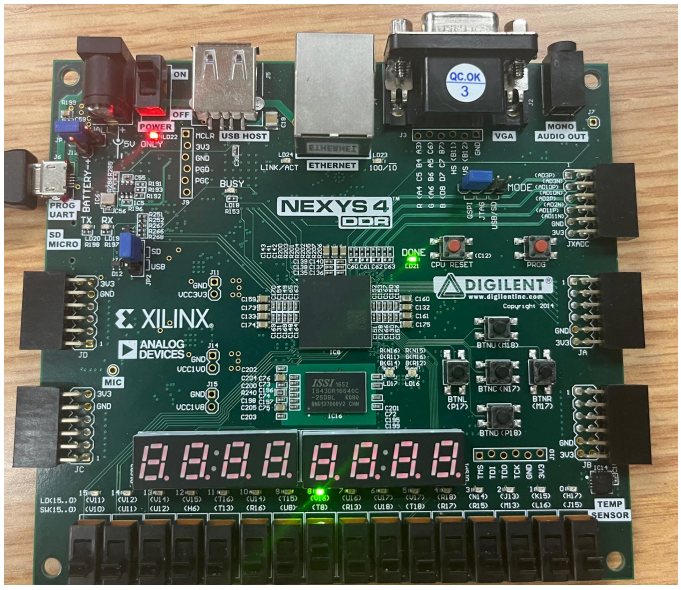

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
ETHERNET
USB HOST
VGA
AUDIO OUT
CPU RESET
TEMP SENSOR

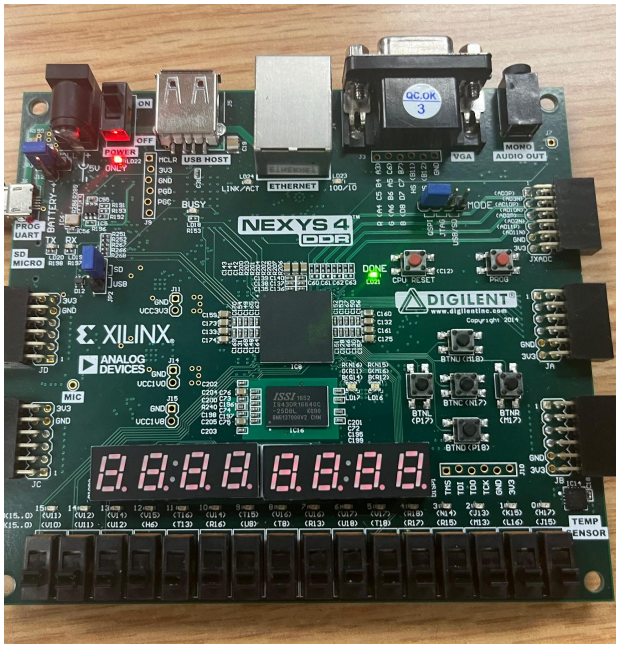

NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
ETHERNET
USB HOST
VGA
AUDIO OUT
CPU RESET
TEMP SENSOR

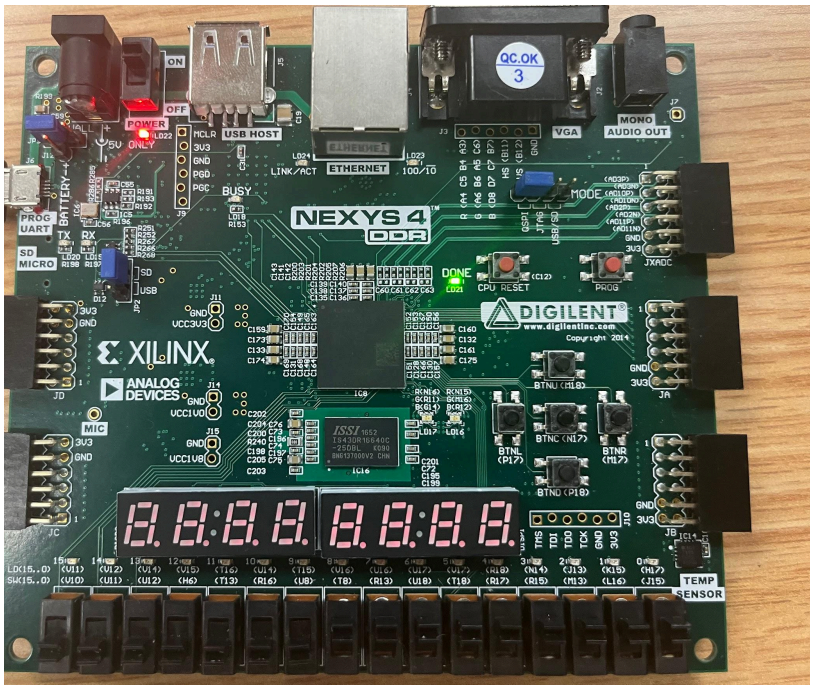

NEXYS 4
DDR
XILINX
DIGILENT
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
PROG
TEMP SENSOR


NEXYS 4
DDR
XILINX
DIGILENT
ANALOG DEVICES
ETHERNET
USB HOST
VGA
MONO AUDIO OUT
CPU RESET
PROG
TEMP SENSOR


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
CPU RESET
PROG
BTNU (M18)
BTNC (N17)
BTNL (P17)
BTNR (M17)
BTND (P18)
TEMP SENSOR


NEXYS 4
DDR
DIGILENT
XILINX
ANALOG DEVICES
QC.OK
3
VGA
MONO AUDIO OUT
USB HOST
ETHERNET
CPU RESET
PROG
BTNU (M18)
BTNC (N17)
BTNL (P17)
BTNR (M17)
BTND (P18)
TEMP SENSOR





So the work completed was heaving to implement the code for the D latch with the Clr input which was easy because I only had to write a little bit of code. Then after I did the same thing but for the D flip flop with the clear input and I just also wrote a little bit of code but observed the results we had. Then at last I completed the 8 bit register only observing the results, but I am not sure if my waveform is completely correct.