ECE 4525

Lab 8 Report

Jena Francis

Mack Usmanova

Introduction

The purpose of this laboratory experiment is to design, simulate, and implement a synchronous 4-bit binary counter with 3-state outputs, functionally equivalent to the SN74ALS561 integrated circuit. This experiment reinforces the principles of synchronous sequential circuit design, clock signal synchronization, and FPGA-based hardware implementation. The lab focuses on developing VHDL code for a universal counter, integrating proper timing and control logic, and verifying the design through post-route simulations and hardware testing. A bounce-free manual clock circuit is also implemented to ensure reliable signal transitions. The completed design is tested on the Nexys A7 (Artix-7 FPGA) development board, with user inputs provided via DIP switches and output states visualized using LED indicators. This lab emphasizes understanding synchronous design methodologies and practical FPGA interfacing techniques for digital system prototyping.

Procedure

Task 1.1

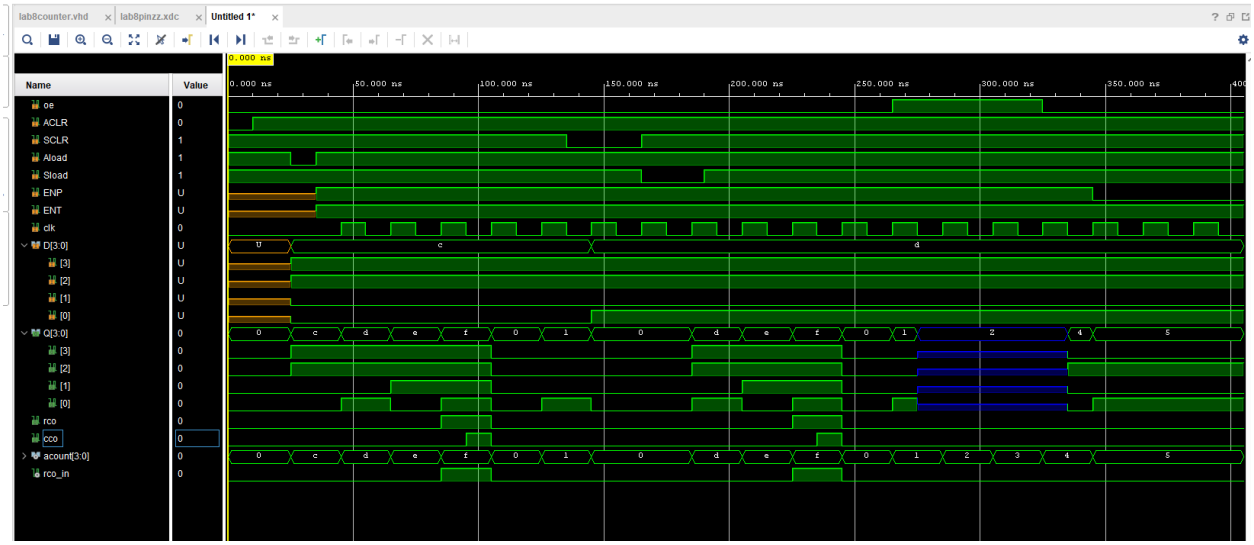

Figure 1: post route simulation following the circuit given in the datasheet.

Task 1.2

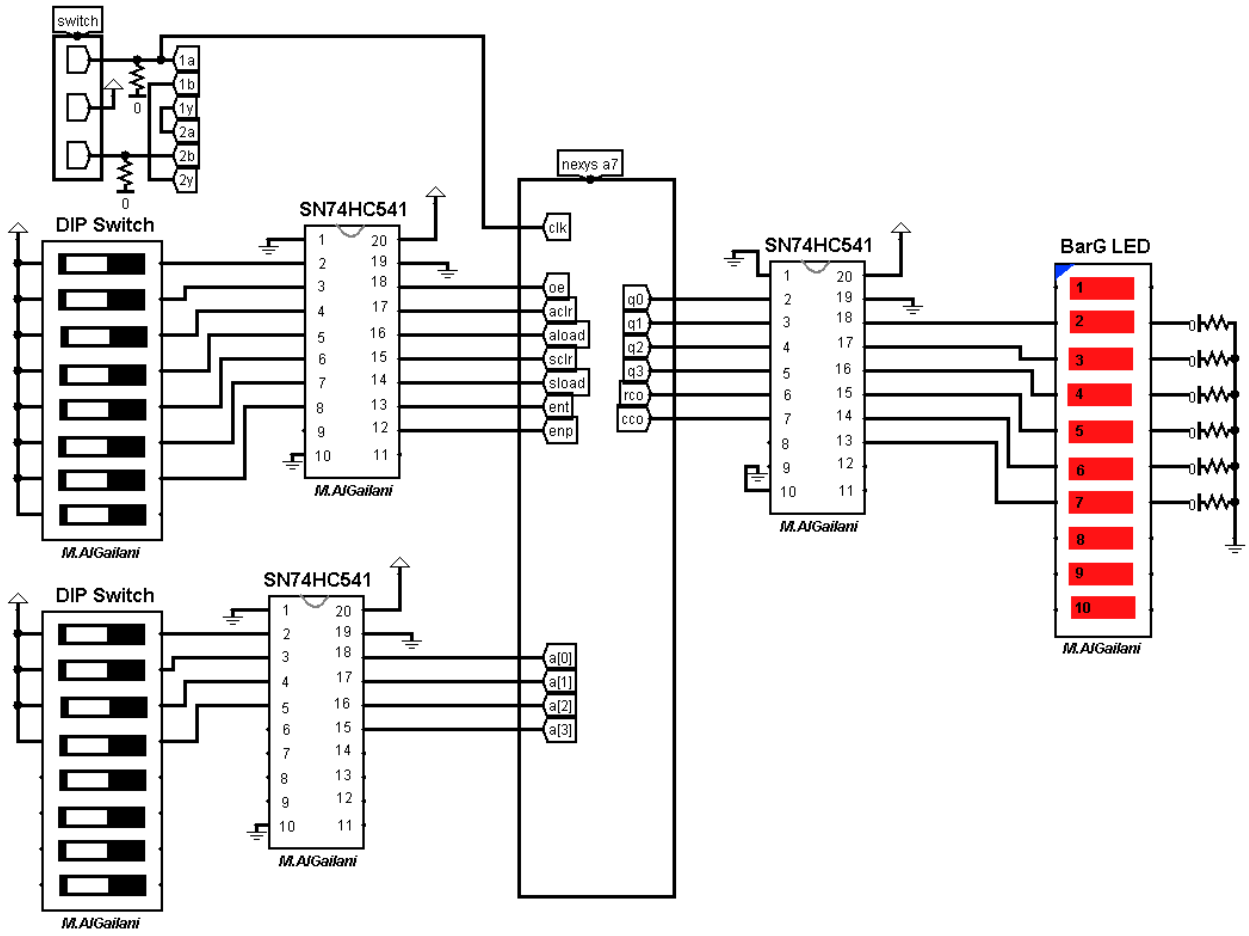

Figure 3: Breadboard schematic.

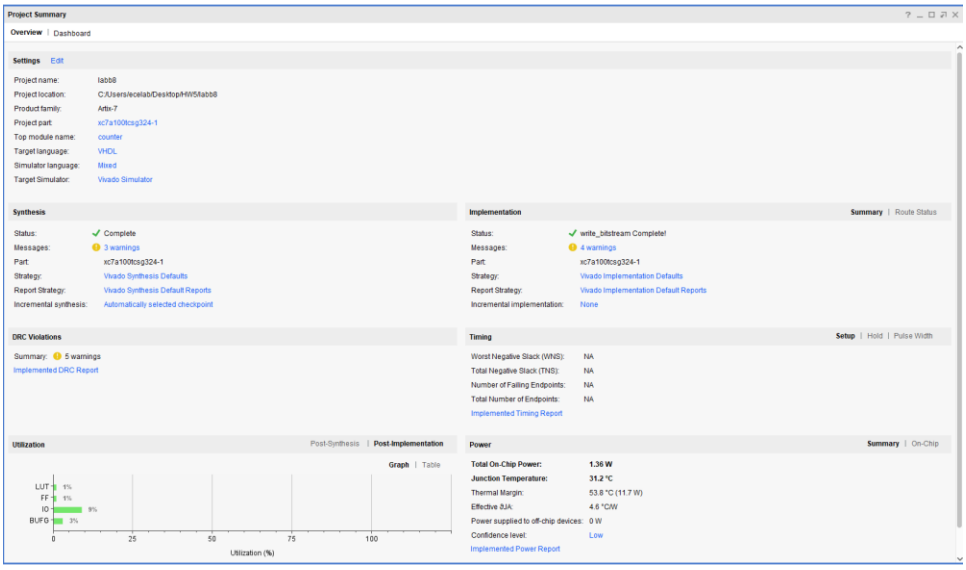

Figure 4: Project summery screenshot

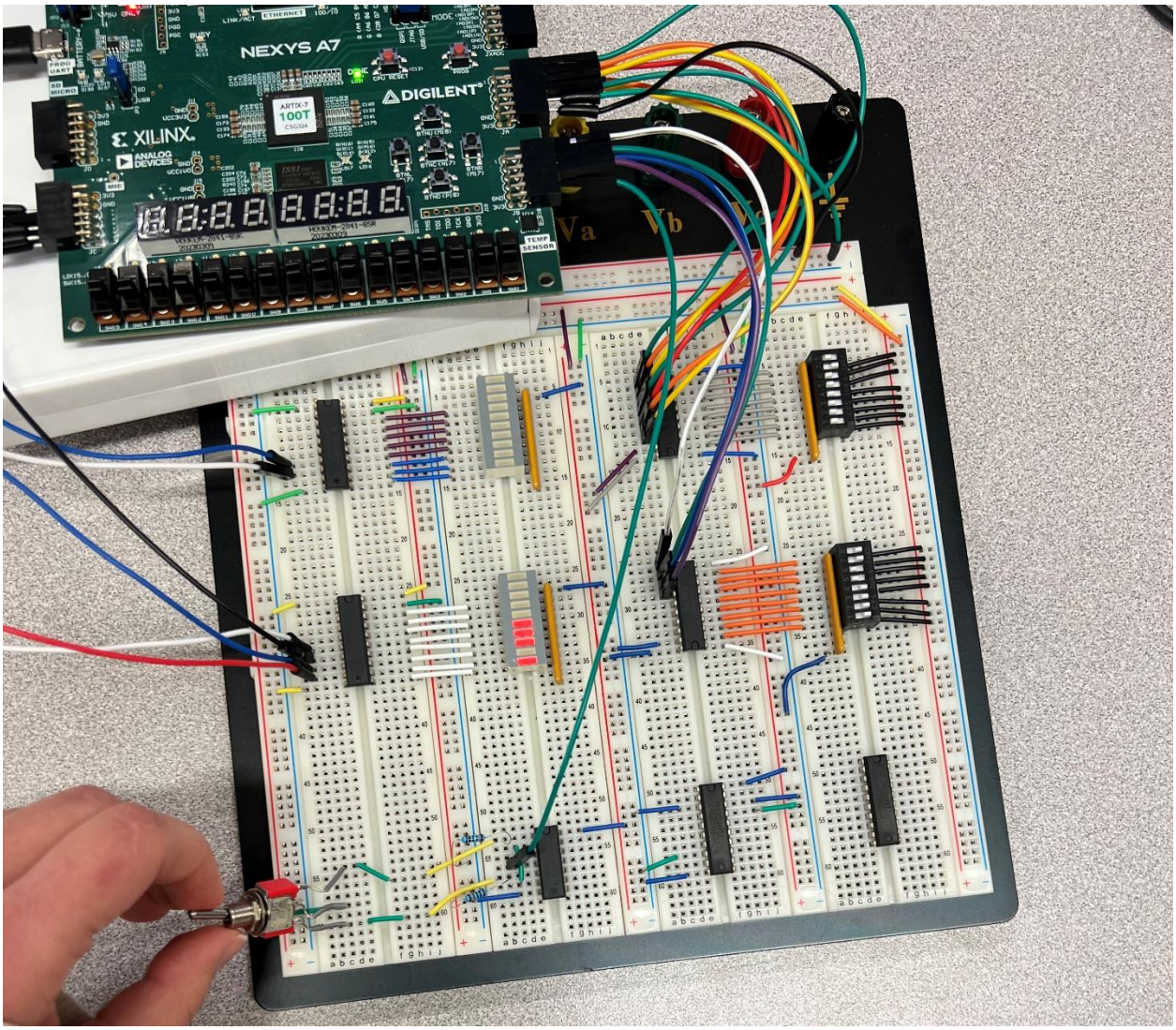

Figure 5: breadboard after loading and incurring a clock event.

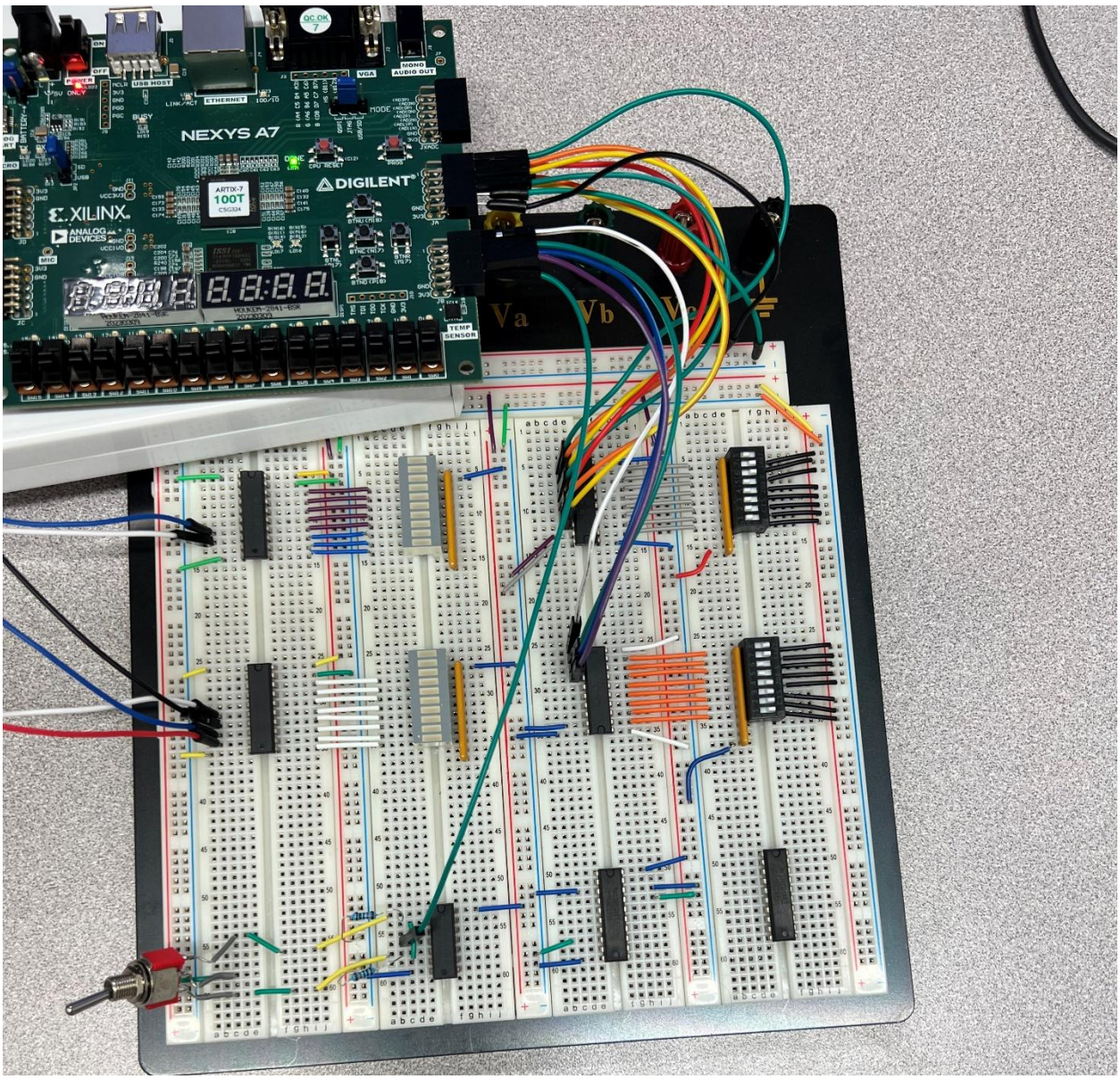

Figure 6: breadboard after aclr is active.

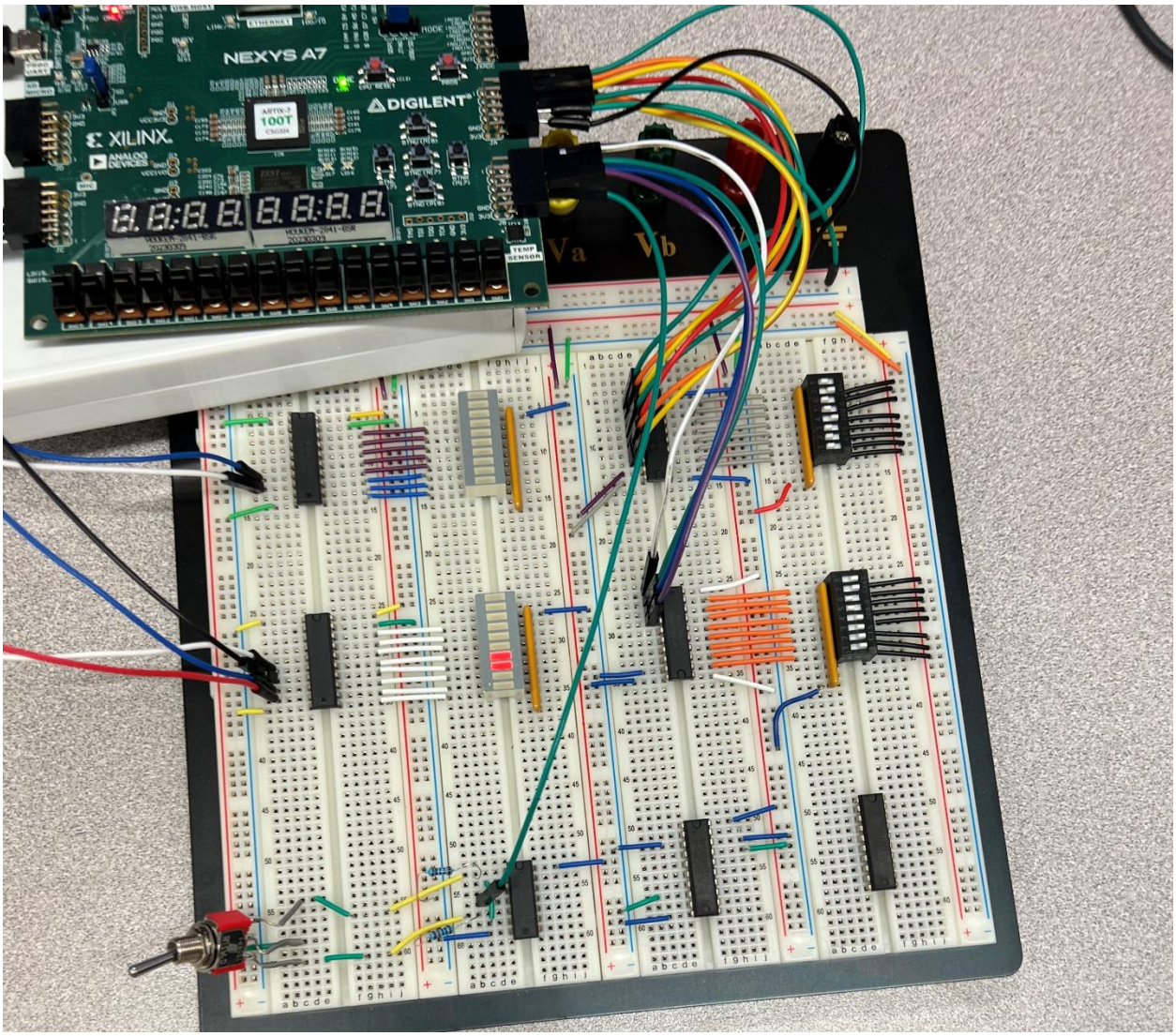

Figure 7: Breadboard after aload is active.

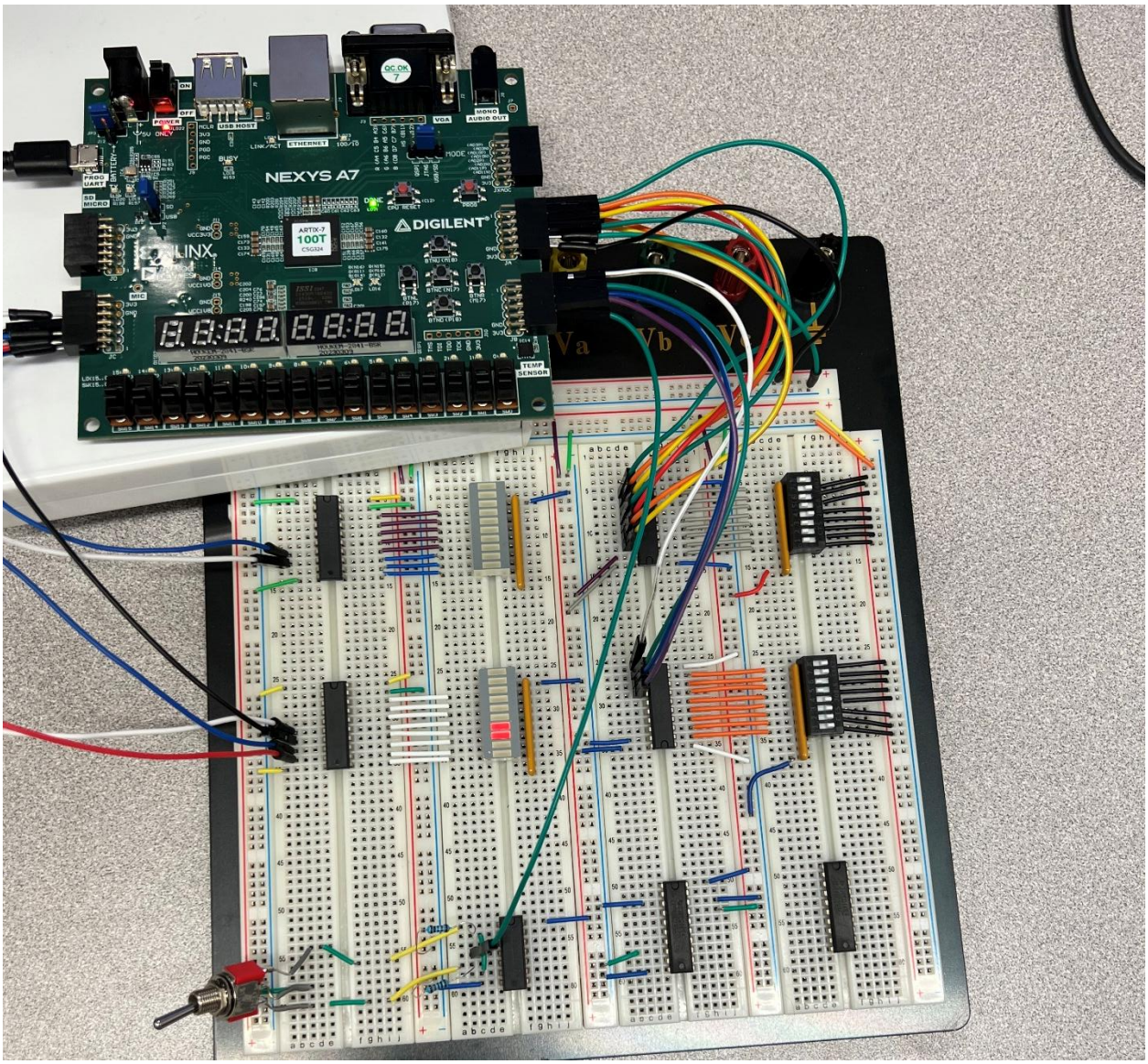

Figure 8: Breadboard after sload is active and clock event occurred.

Task 1.3

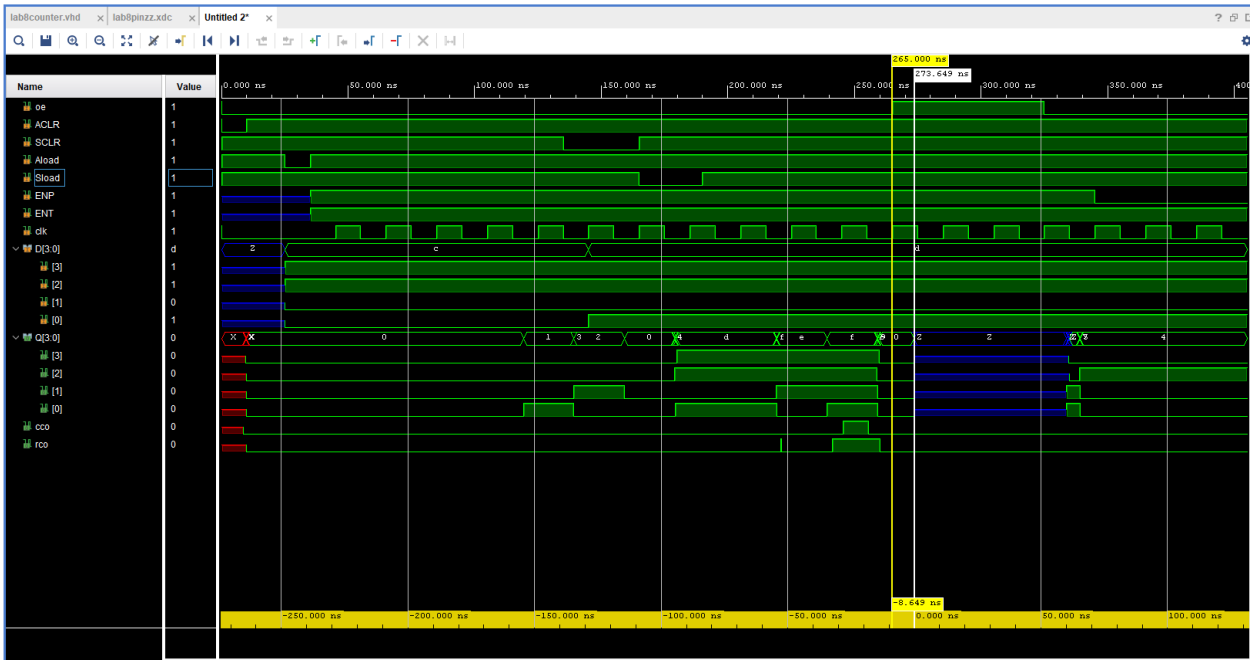

Figure 9: Post route implementation. The delay between output enable turning off, a clk rising edge til q outputs impedence state.

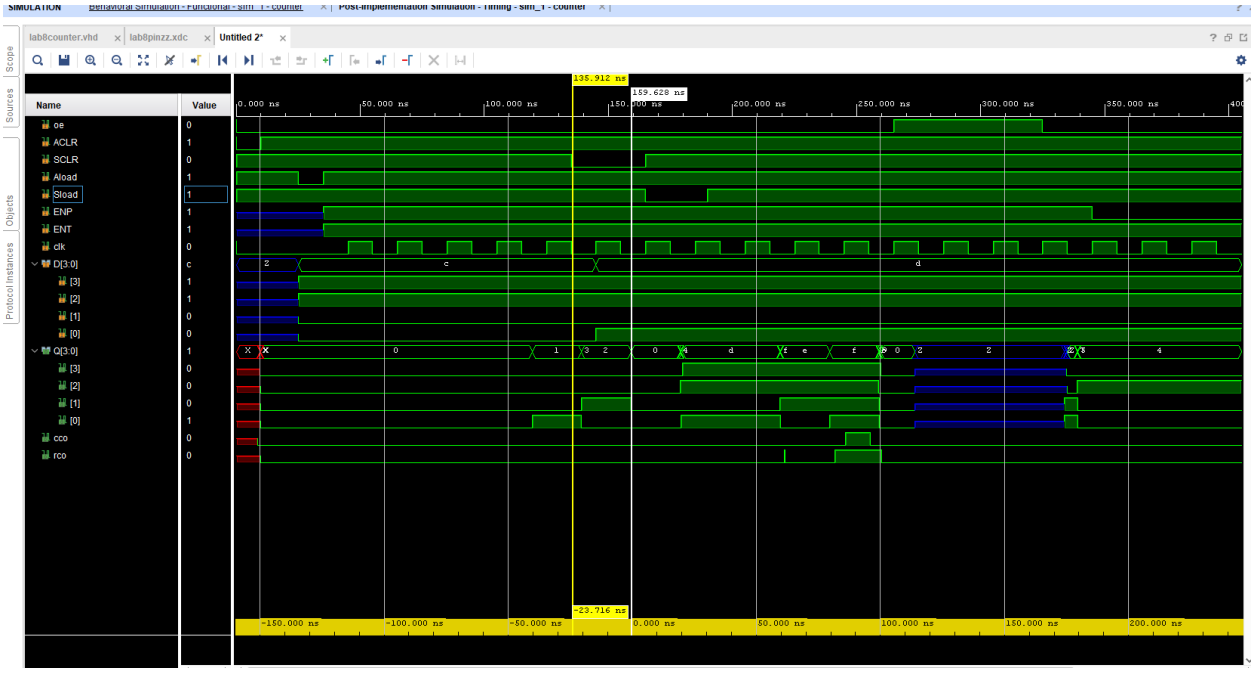

Figure 10: propagation delay between synch clear clearing q output is 23.7 ns.

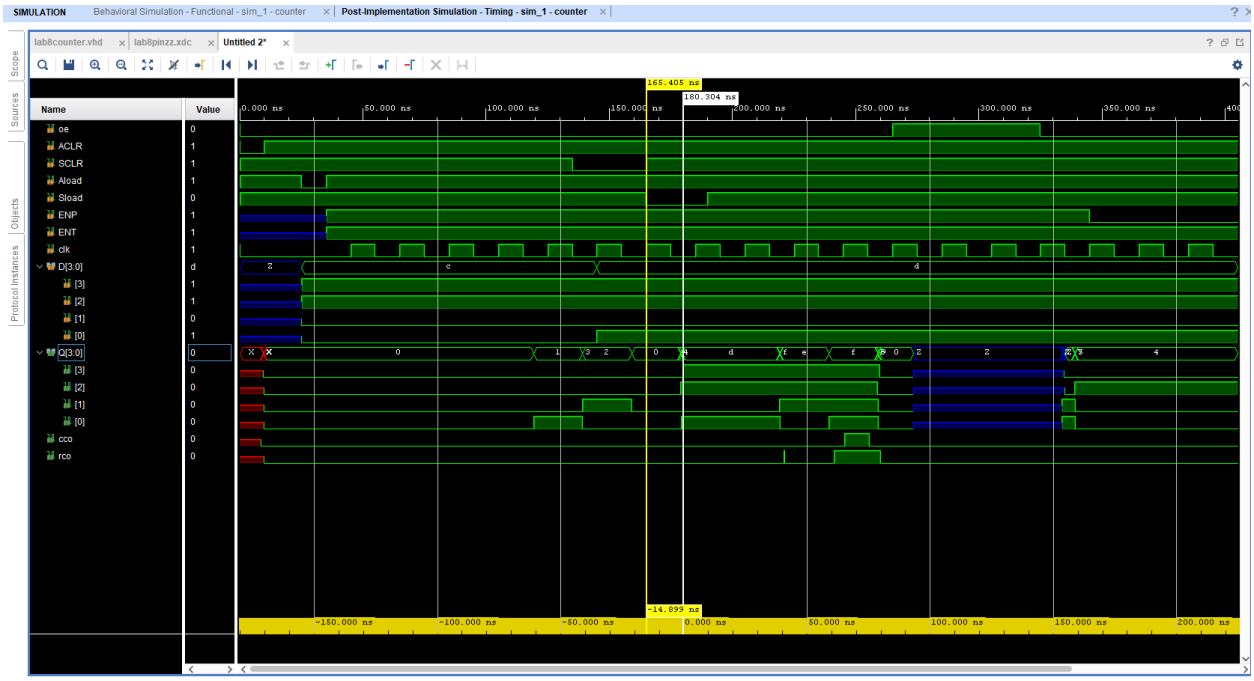

Figure 10: Post route implementation. The propogation synch load and rising edge til q load d input is 14.89ns.

Conclusion

In this lab, a synchronous 4-bit binary counter with 3-state outputs was successfully designed, simulated, and implemented on the Nexys A7 FPGA using VHDL. The project demonstrated how theoretical digital design concepts—such as synchronous clocking, propagation delay analysis, and tri-state output control—translate into real hardware functionality. Simulation results confirmed that the counter operated correctly according to the SN74ALS561 timing diagram, and hardware testing verified proper counting and output display. The inclusion of a bounce-free clock circuit ensured stable input transitions during manual testing. Overall, the experiment strengthened understanding of FPGA implementation, digital timing behavior, and sequential circuit design, while providing practical experience in integrating schematic design, VHDL coding, and post-route verification.

Appendix

Task 1

Vhdl

library IEEE; use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using -- arithmetic functions with Signed or Unsigned values --use IEEE.NUMERIC_STD.ALL; use IEEE.std_logic_unsigned.all; -- Uncomment the following library declaration if instantiating -- any Xilinx leaf cells in this code. --library UNISIM; --use UNISIM.VComponents.all;

entity counter is Port (ENP, ENT, clk, oe : in std_logic; ACLR, Aload, SCLR, Sload : in std_logic; D : in std_logic_vector(3 downto 0); rco, cco : out std_logic; Q : out std_logic_vector(3 downto 0));

end counter;

architecture Behavioral of counter is signal acount : std_logic_vector(3 downto 0);

signal rco_in : std_logic; begin

```
        process (ENT, ENP, ACLR, SCLR, Aload, Sload, acount, D, clk)
         begin

          if ACLR = '0' then
             acount <= "0000";

          elsif Aload = '0' then
             acount <= D;
          elsif rising_edge(clk) then
                if ENP = '1' and ENT = '1' and Sload = '1' and SCLR =
'1' and Aload = '1' and ACLR = '1' then
                     acount <= acount +1;

                 elsif SCLR = '0' then
                      acount <= "0000";
```

```
         elsif Sload = '0' then
            acount <= D;
         elsif ENP = '0' then
            acount <= acount;
            end if;
            end if;


      if oe = '0' then
        Q <= acount;
      else
        Q <= (others => 'Z');
        end if;


  if acount = 15 then
        rco <= '1';
        rco_in <= '1';
    else
        rco <= '0';
        rco_in <= '0';
        end if;


   end process;
```

process(clk) begin

```
       if rco_in = '1' and clk = '0' then
            cco <= '1';
        else
            cco <= '0';
             end if;
```

end process;

end Behavioral;

Tcl

```
restart

add_force oe {0 0ns} add_force ACLR {0 0ns} add_force SCLR {1 0ns} add_force Aload {1 0ns} add_force Sload {1 0ns} add_force clk {0 0ns}

run 10ns

add_force ACLR {1 0ns}

run 15ns

add_force Aload {0 0ns} add_force D {1100 0ns}

run 10ns add_force clk {0 0ns} {1 10ns} -repeat_every 20ns add_force Aload {1 0ns}

add_force ACLR {1 0ns} add_force ENP {1 0ns} add_force ENT {1 0ns}

run 100ns

add_force SCLR {0 0ns} run 10ns add_force D {1101 0ns} run 20ns

add_force SCLR {1 0ns} add_force Sload {0 0ns}

run 25ns

add_force Sload {1 0ns}

run 75ns

add_force oe {1 0ns}

run 60ns

add_force oe {0 0ns}

run 20ns

add_force ENP {0 0ns}

run 60ns
```

Xdc

```
#clk set_property -dict { PACKAGE_PIN H16 IOSTANDARD LVCMOS33 } [get_ports { clk }];
#IO_L13P_T2_MRCC_15 Sch=jb[10]

#data inputs

set_property -dict { PACKAGE_PIN D14 IOSTANDARD LVCMOS33 } [get_ports { D[0] }];
#IO_L1P_T0_AD0P_15 Sch=jb[1]

set_property -dict { PACKAGE_PIN F16 IOSTANDARD LVCMOS33 } [get_ports { D[1] }];
#IO_L14N_T2_SRCC_15 Sch=jb[2]

set_property -dict { PACKAGE_PIN G16 IOSTANDARD LVCMOS33 } [get_ports { D[2] }];
#IO_L13N_T2_MRCC_15 Sch=jb[3]

set_property -dict { PACKAGE_PIN H14 IOSTANDARD LVCMOS33 } [get_ports { D[3] }];
#IO_L15P_T2_DQS_15 Sch=jb[4]

#other inputs

set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { oe }];
#IO_L20N_T3_A19_15 Sch=ja[1]

set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { ACLR }];
#IO_L21N_T3_DQS_A18_15 Sch=ja[2]

set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { SCLR }];
#IO_L21P_T3_DQS_15 Sch=ja[3]

set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { Aload }];
#IO_L18N_T2_A23_15 Sch=ja[4]

set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { Sload }];
#IO_L16N_T2_A27_15 Sch=ja[7]

set_property -dict { PACKAGE_PIN E17 IOSTANDARD LVCMOS33 } [get_ports { ENP }];
#IO_L16P_T2_A28_15 Sch=ja[8]

set_property -dict { PACKAGE_PIN F18 IOSTANDARD LVCMOS33 } [get_ports { ENT }];
#IO_L22N_T3_A16_15 Sch=ja[9]

#outputs set_property -dict { PACKAGE_PIN K1 IOSTANDARD LVCMOS33 } [get_ports {
Q[0] }]; #IO_L23N_T3_35 Sch=jc[1]
```

```
set_property -dict { PACKAGE_PIN F6 IOSTANDARD LVCMOS33 } [get_ports { Q[1] }];
#IO_L19N_T3_VREF_35 Sch=jc[2]

set_property -dict { PACKAGE_PIN J2 IOSTANDARD LVCMOS33 } [get_ports { Q[2] }];
#IO_L22N_T3_35 Sch=jc[3]

set_property -dict { PACKAGE_PIN G6 IOSTANDARD LVCMOS33 } [get_ports { Q[3] }];
#IO_L19P_T3_35 Sch=jc[4]

set_property -dict { PACKAGE_PIN E7 IOSTANDARD LVCMOS33 } [get_ports { rco }];
#IO_L6P_T0_35 Sch=jc[7]

set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports { cco }];
#IO_L22P_T3_35 Sch=jc[8]
```