ECE 4525

Lab 10 Report

Jena Mae Francis

Mack Usmanova

Introduction

This lab focused on designing, implementing, and evaluating an asynchronous sequential circuit. To begin, we created a critical race-free state assignment, ensuring that no unintended transitions could happen, based on the provided reduced state table. From there we derived the next-state equations. These equations were then put into vhdl code and verified using post-route simulations. Hardware was also used to verify functionality and make sure it moved through the states properly.

Procedure

Task 1.1

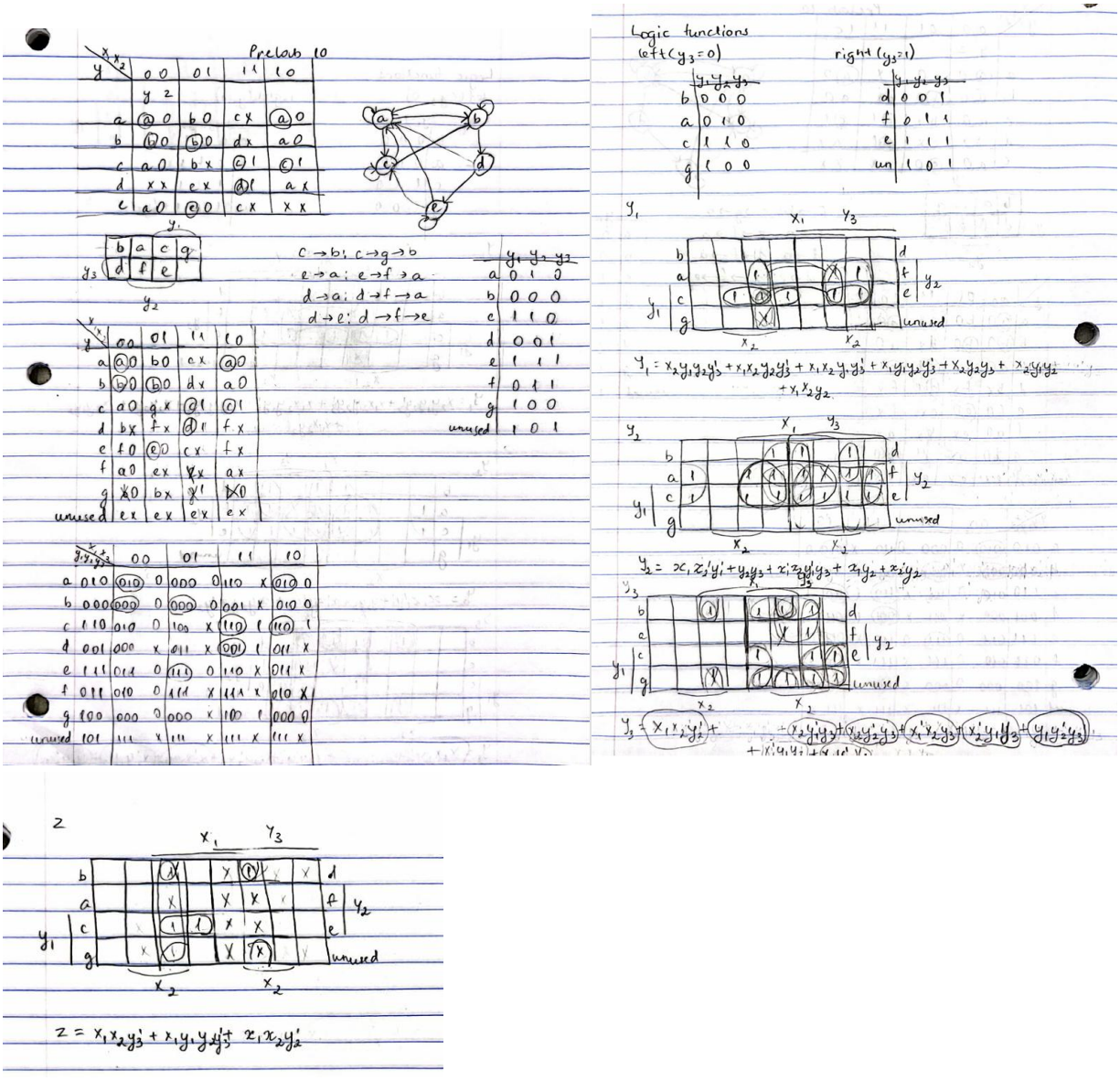

Figure 1: critical race free state assignment table and Kmap

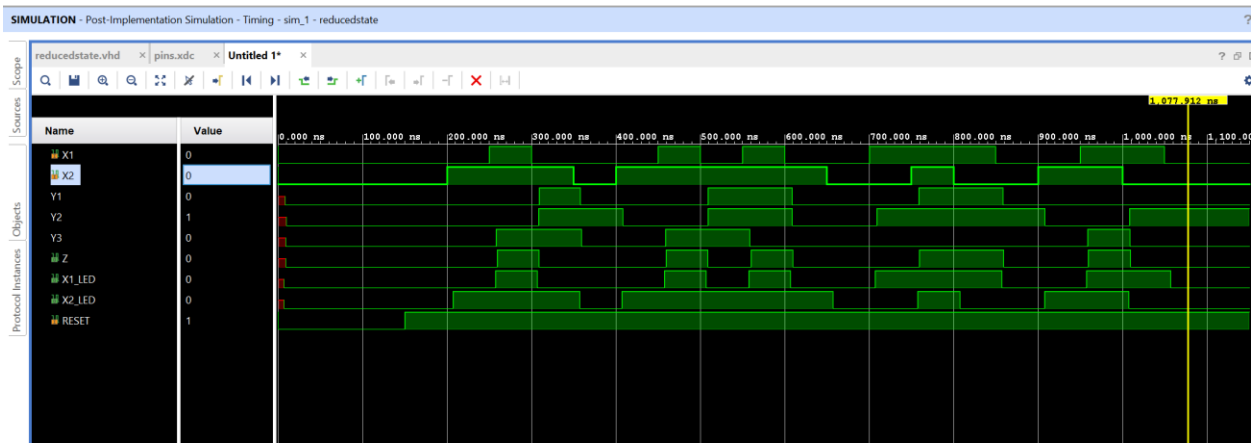

Figure 2: post route simulation verifying every transition

Task 1.2

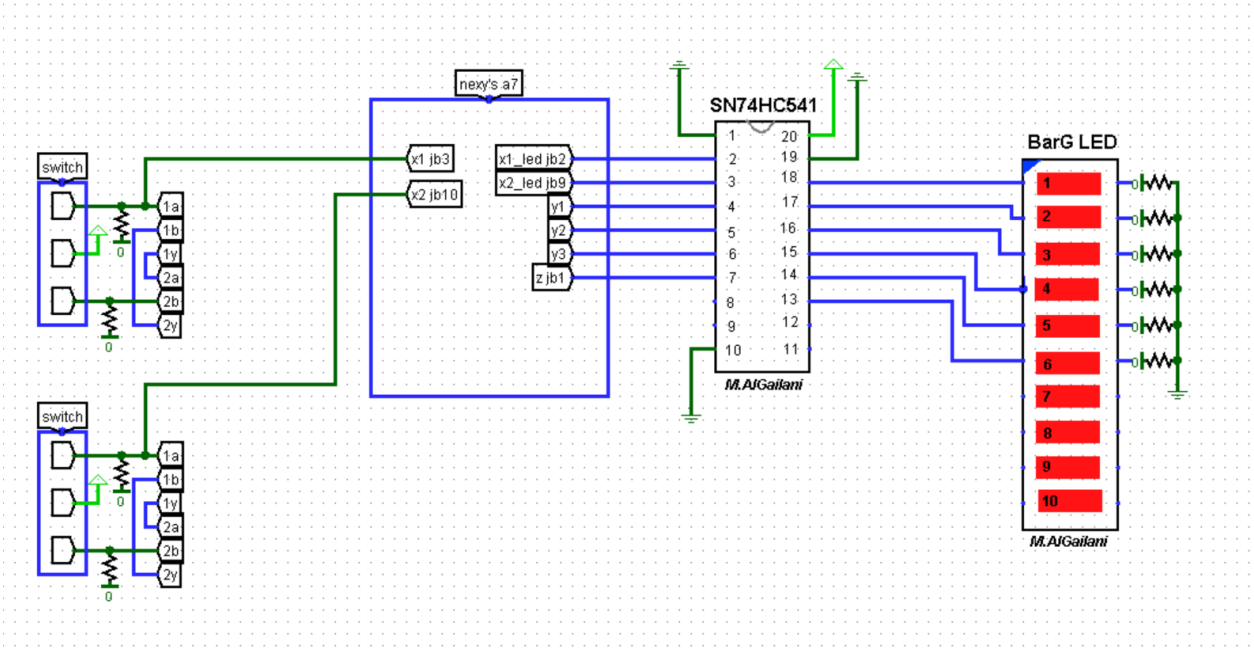

Figure 3: breadboard schematic

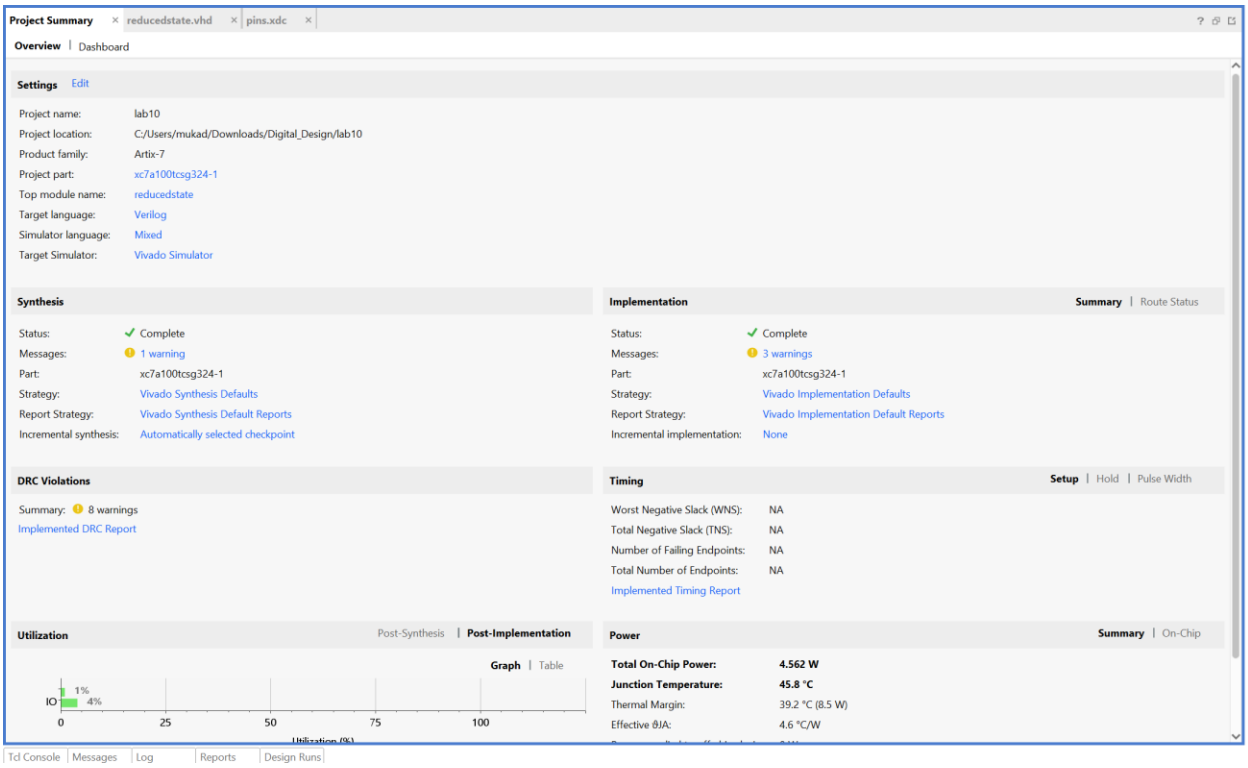

Figure 4: Project summary after running generate bitstream.

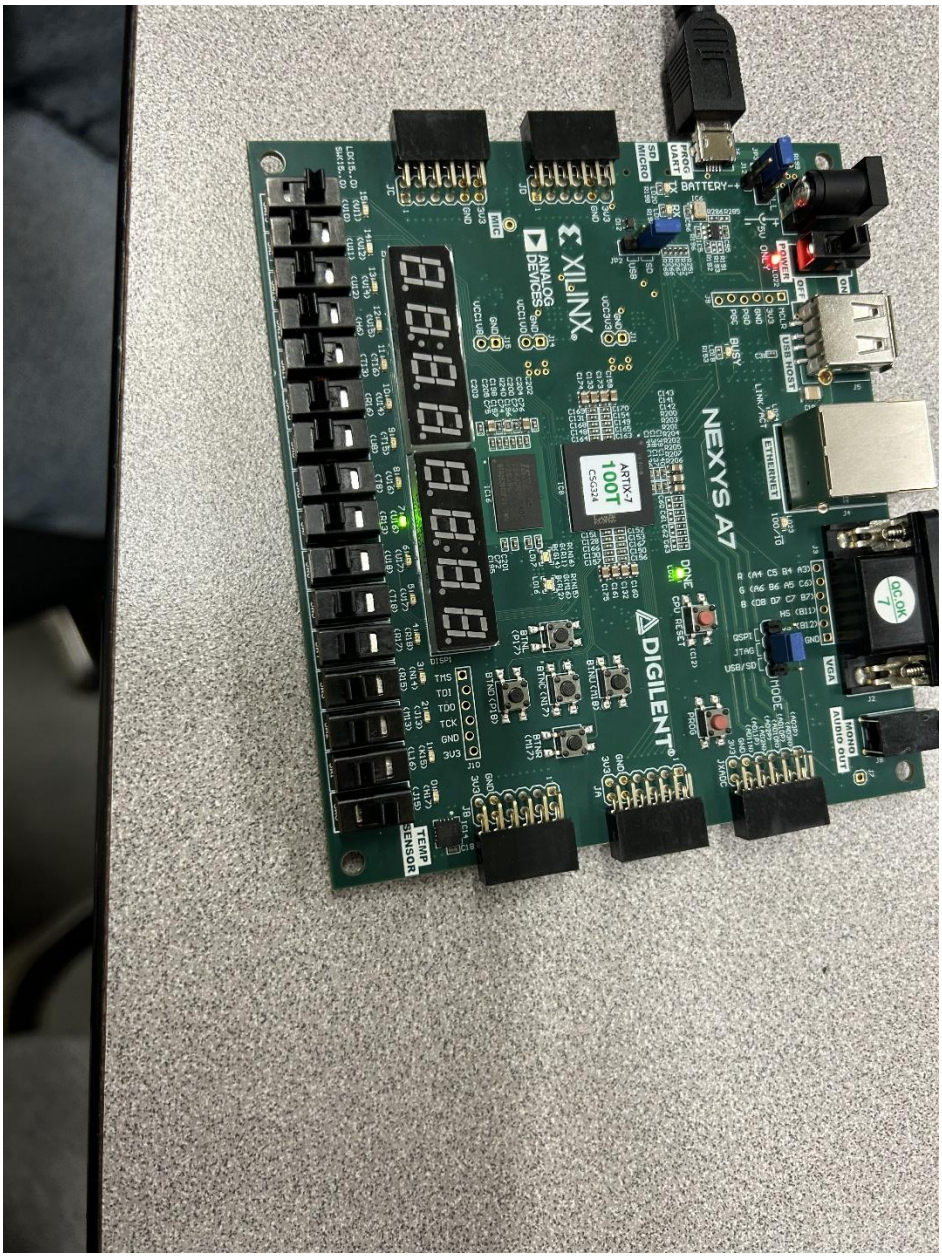

Figure 5: This shows state a. X1 and X2 are low, Y1 =0, Y2= 1, Y3=0, and Z = 0.

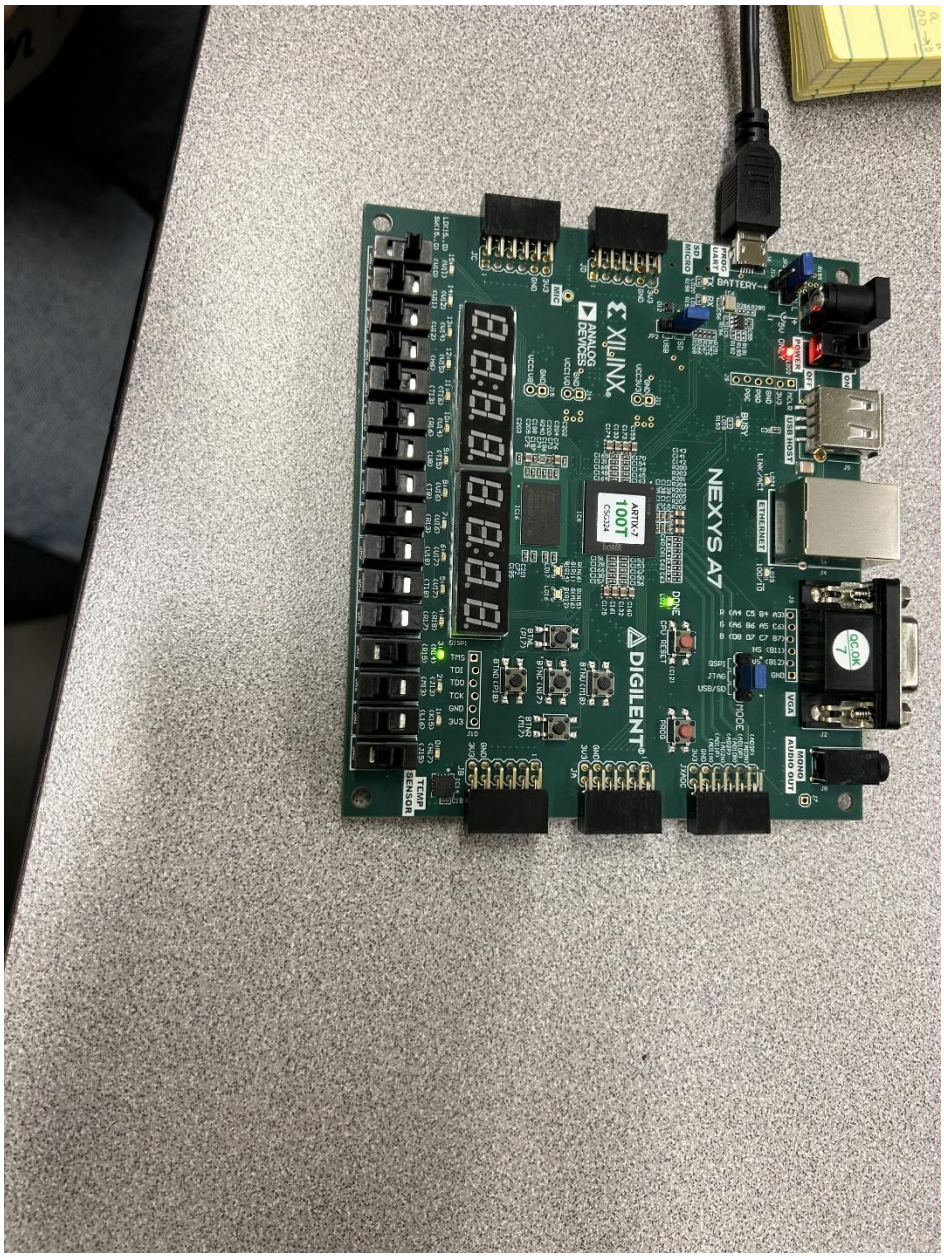

Figure 6: This shows state b. X1 = 0 and X2 = 1, Y1 =0, Y2= 0, Y3=0, and Z = 0.

Figure 7: This shows state c. X1 and X2 are high, Y1 =1, Y2= 1, Y3=0, and Z = 1.

Figure 8: This shows state d. X1 and X2 are high, Y1 =0, Y2= 0, Y3=1, and Z = 1.

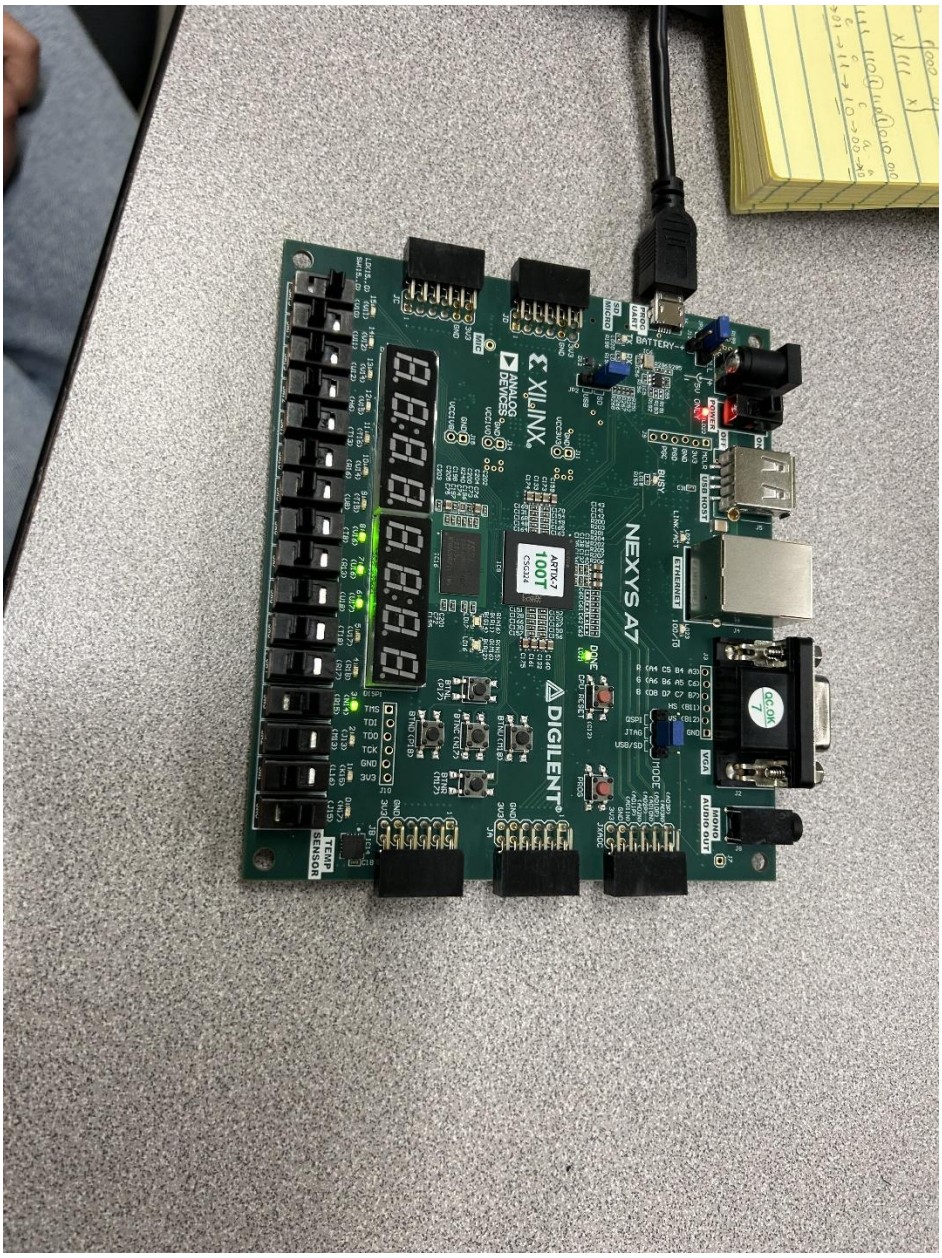

Figure 9: This shows state e. X1= 0and X2= 1, Y1 =1, Y2= 1, Y3=1, and Z = 0.

Conclusion

This lab provided hands-on experience with designing and analyzing asynchronous sequential circuits. By beginning with a reduced state table and developing a critical race-free assignment, we ensured stable and predictable state transitions. The next-state equations were derived and implemented through vhdl. Post-route simulations and hardware verified the functionality of the circuit.

Appendix

Vhd

```
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;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity reducedstate is
  Port (
        X1, X2, RESET: in std_logic;
        Y1, Y2, Y3: inout std_logic;
        X1_LED, X2_LED, Z: out std_logic
  );
end reducedstate;

architecture Behavioral of reducedstate is

begin

        Y1 <= ((x2 and y1 and y2 and not(y3))
                or (x1 and x2 and y2 and not(y3))
                or (x1 and x2 and y1 and not(y3))
                or (x1 and y1 and y2 and not(y3))
                or (x2 and y2 and y3)
                or (x2 and y1 and y2)
                or (x1 and x2 and y2)) and reset;

        Y2 <= ((x1 and not(x2) and not(y1))
                or (not(x1) and x2 and not(y1) and y3)
                or (y2 and y3)
                or (x1 and y2)
                or (not(x2) and y2)) and reset;

         Y3 <= ((x1 and x2 and not(y2))
                or (x2 and not(y1) and y3)
                or (x2 and not(y2) and y3)
                or (not(x1) and x2 and y3)
                or (not(x2) and y1 and y3)
                or (y1 and not(y2) and y3)
                or (not(x1) and y1 and y3)
                or (x1 and not(y2) and y3)) and reset;

        Z <= (x1 and x2 and not(y3))
                or (x1 and y1 and y2 and not(y3))
                or (x1 and x2 and not(y2));

        x1_led <= x1;
        x2_led <= x2;

end Behavioral;
```

Xdc

```
##Switches
set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { RESET }];
#IO_L24N_T3_RS0_15 Sch=sw[0]
```

```
set_property -dict { PACKAGE_PIN R15 IOSTANDARD LVCMOS33 } [get_ports { X2 }];
#IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17 IOSTANDARD LVCMOS33 } [get_ports { X1 }];
#IO_L12N_T1_MRCC_14 Sch=sw[4]

set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { Z }];
#IO_L18P_T2_A24_15 Sch=led[0]
#set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { LED[1] }];
#IO_L24P_T3_RS1_15 Sch=led[1]
#set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { X2_LED }];
#IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33 } [get_ports { X2_LED }];
#IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports { X1_LED }];
#IO_L7P_T1_D09_14 Sch=led[4]
#set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports { y3led }];
#IO_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 } [get_ports { Y3 }];
#IO_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 } [get_ports { Y2 }];
#IO_L18P_T2_A12_D28_14 Sch=led[7]
set_property -dict { PACKAGE_PIN V16 IOSTANDARD LVCMOS33 } [get_ports { Y1 }];
#IO_L16N_T2_A15_D31_14 Sch=led[8]

set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets Y1*];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets Y2*];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets Y3*];
```

Tcl

```
#inputs:
# X1, X2, RESET STD_LOGIC

#outputs:
# Y1, Y2, Y3, Z, X1_LED, X2_LED STD_LOGIC

restart

# apply active low RESET
add_force RESET {0 0ns}

# initialize inputs
add_force X1 {0 0ns}
add_force X2 {0 0ns}

# wait out 150ns
run 150ns

# set active low RESET back to inactive state
add_force RESET {1 0ns}
run 50ns

# go to b
add_force X2 {1 0ns}
run 50ns

# go to d
add_force X1 {1 0ns}
run 50ns

# go to e
add_force X1 {0 0ns}
run 50ns

# go to a
add_force X2 {0 0ns}
run 50ns

# again b
add_force X2 {1 0ns}
```

```
run 50ns

# go to d
add_force X1 {1 0ns}
run 50ns

# go to e
add_force X1 {0 0ns}
run 50ns

# go to c
add_force X1 {1 0ns}
run 50ns

# go to b
add_force X1 {0 0ns}
run 50ns

# go to b
add_force X2 {0 0ns}
run 50ns

# go to a
add_force X1 {1 0ns}
run 50ns

# go to c
add_force X2 {1 0ns}
run 50ns

# go to c
add_force X2 {0 0ns}
run 50ns

# go to a
add_force X1 {0 0ns}
run 50ns

# go to b
add_force X2 {1 0ns}
run 50ns

# go to d
add_force X1 {1 0ns}
run 50ns

# go to a
add_force X2 {0 0ns}
run 50ns

# go to a
add_force X1 {0 0ns}
run 100ns
```