ECE 4525

Lab 11 Report

Jena Mae Francis

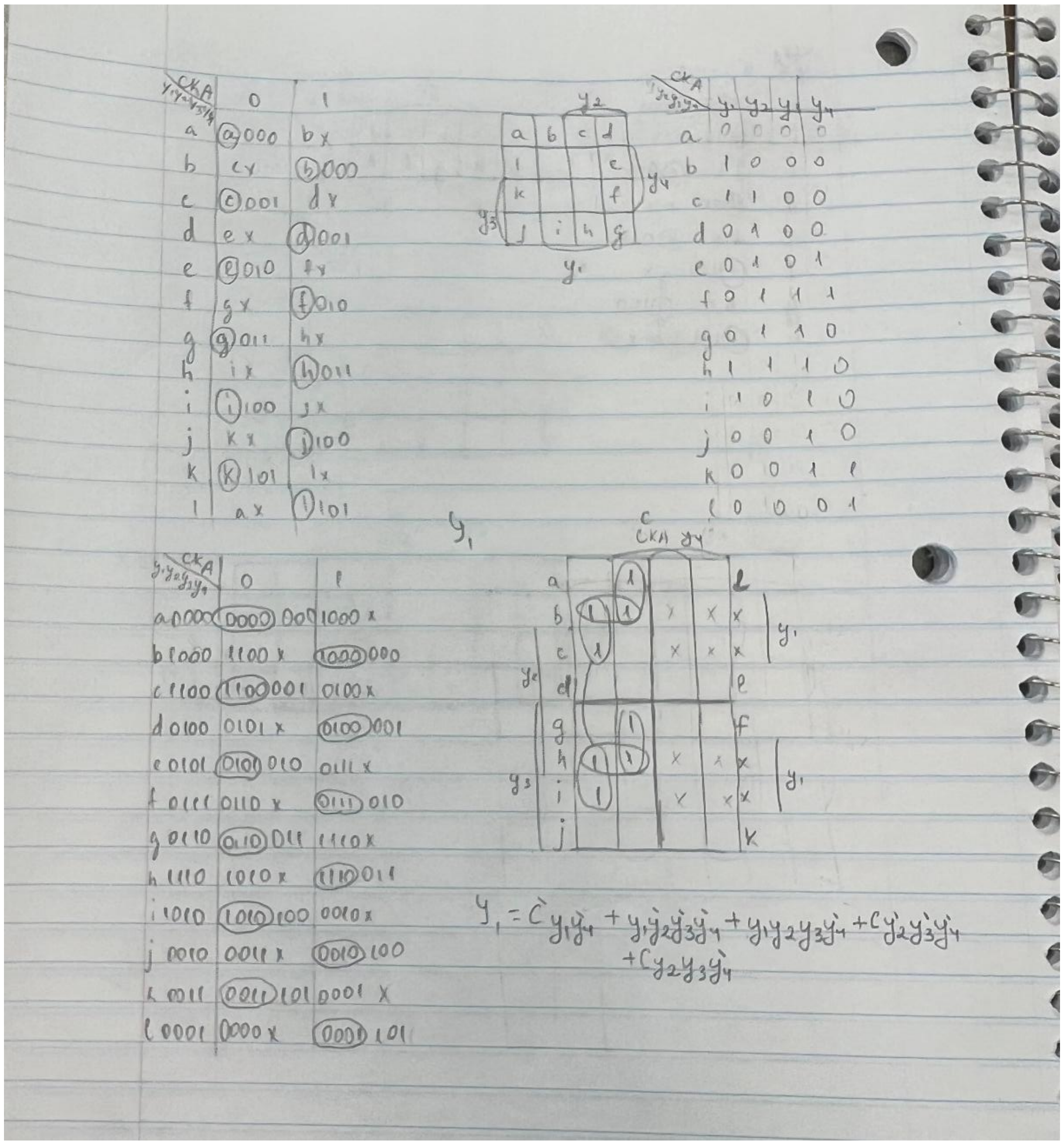
Mack Usmanova

Table of Contents

Introduction

This lab focused on designing and implementing an asynchronous sequential counter based on a modified version of the SN7490A decade counter. The main objective of this lab was to develop a functional asynchronous counter using VHDL that fits these new specifications. The critical race free state assignment was made as well as an assigned state table. Hardware was implemented to verify the counter worked properly and that it moved through the correct states.

Procedure



$y_1y_2y_3y_4$ \ CKA	0	1
a	a 000	b x
b	c x	b 000
c	c 001	d x
d	e x	d 001
e	e 010	f x
f	g x	f 010
g	g 011	h x
h	i x	h 011
i	i 100	j x
j	k x	j 100
k	k 101	l x
l	a x	l 101

a	b	c	d
l			e
k			f
j	i	h	g

y_2
y_4
y_3
y_1

$y_1y_2y_3y_4$ \ CKA	y_1	y_2	y_3	y_4
a	0	0	0	0
b	1	0	0	0
c	1	1	0	0
d	0	1	0	0
e	0	1	0	1
f	0	1	1	1
g	0	1	1	0
h	1	1	1	0
i	1	0	1	0
j	0	0	1	0
k	0	0	1	1
l	0	0	0	1

Y_1

a		1			l
b	1	1	x	x	x
c	1		x	x	x
d					e
g		1			f
h	1	1	x	x	x
i	1		x	x	x
j					k

C
CKA y_4
y_2
y_3
y_1
y_1

$y_1y_2y_3y_4$ \ CKA	0	1
a 0000	0000 000	1000 x
b 1000	1100 x	1000 000
c 1100	1100 001	0100 x
d 0100	0101 x	0100 001
e 0101	0101 010	0111 x
f 0111	0110 x	0111 010
g 0110	0110 011	1110 x
h 1110	1010 x	1110 011
i 1010	1010 100	0010 x
j 0010	0011 x	0010 100
k 0011	0011 101	0001 x
l 0001	0000 x	0001 101

$$Y_1 = C' y_1 y_4' + y_1 y_2' y_3' y_4' + y_1 y_2 y_3 y_4' + C y_2' y_3' y_4' + C y_2 y_3 y_4'$$

Fig 1: Critical race free state assignment and assigned state table.

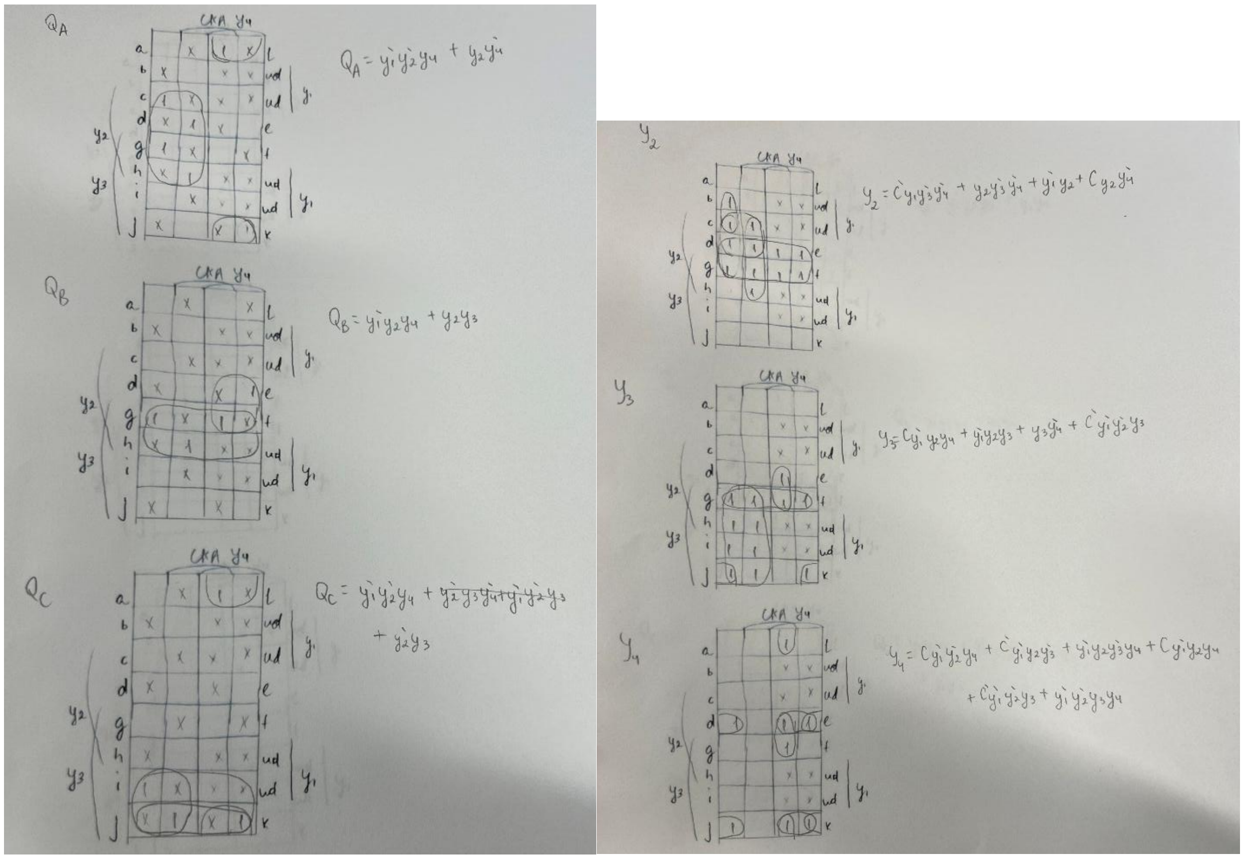
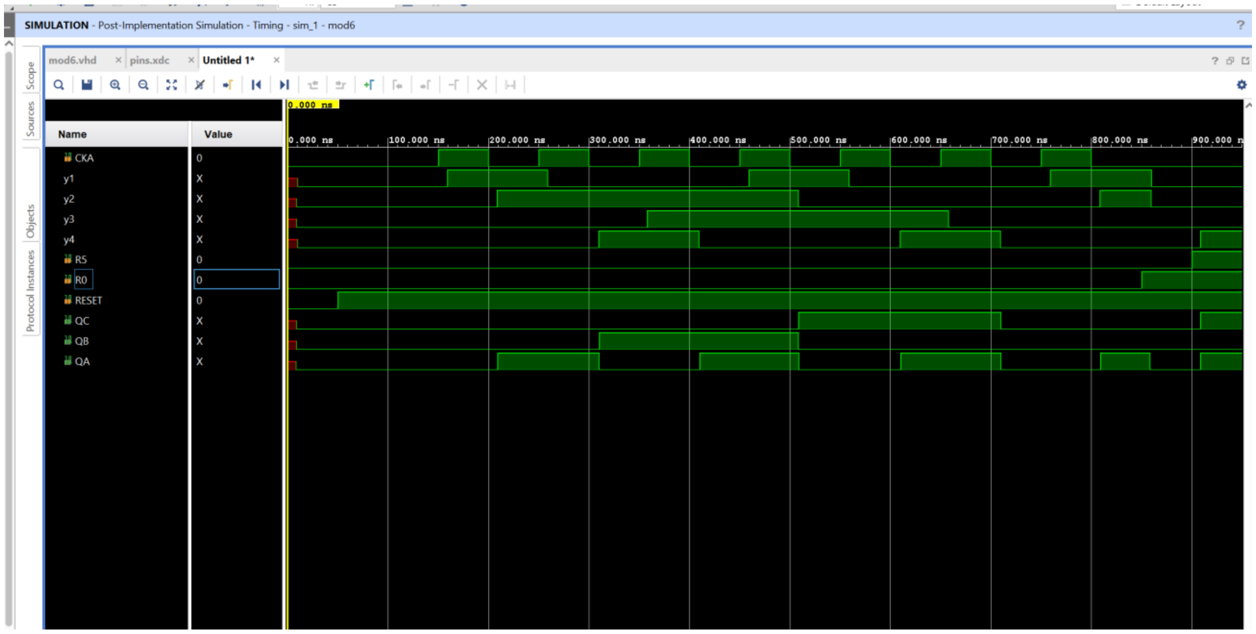

Q_A
CKA y_4
$Q_A = y_1' y_2' y_4 + y_2 y_4'$
Q_B
$Q_B = y_1' y_2 y_4 + y_2 y_3$
Q_C
$Q_C = y_1' y_2' y_4 + ~~y_2' y_3 y_4' + y_1' y_2' y_3~~ + y_2' y_3$
y_2
$y_2 = C' y_1 y_3' y_4' + y_2 y_3' y_4' + y_1' y_2 + C y_2 y_4'$
y_3
$y_3 = C y_1' y_2 y_4 + y_1' y_2 y_3 + y_3 y_4' + C' y_1' y_2' y_3$
y_4
$y_4 = C y_1' y_2' y_4 + C' y_1' y_2 y_3' + y_1' y_2 y_3' y_4 + C y_1' y_2 y_4 + C' y_1' y_2' y_3 + y_1' y_2' y_3 y_4$

Fig 2: rest of the state assignments



SIMULATION - Post-Implementation Simulation - Timing - sim_1 - mod6

mod6.vhd | pins.xdc | Untitled 1*

Name	Value
CKA	0
y1	X
y2	X
y3	X
y4	X
R5	0
R0	0
RESET	0
QC	X
QB	X
QA	X

Fig 3: Post-route simulation showing a 0-5 loop then showing r0 resetting q and y outputs and then r5 having priority over it.

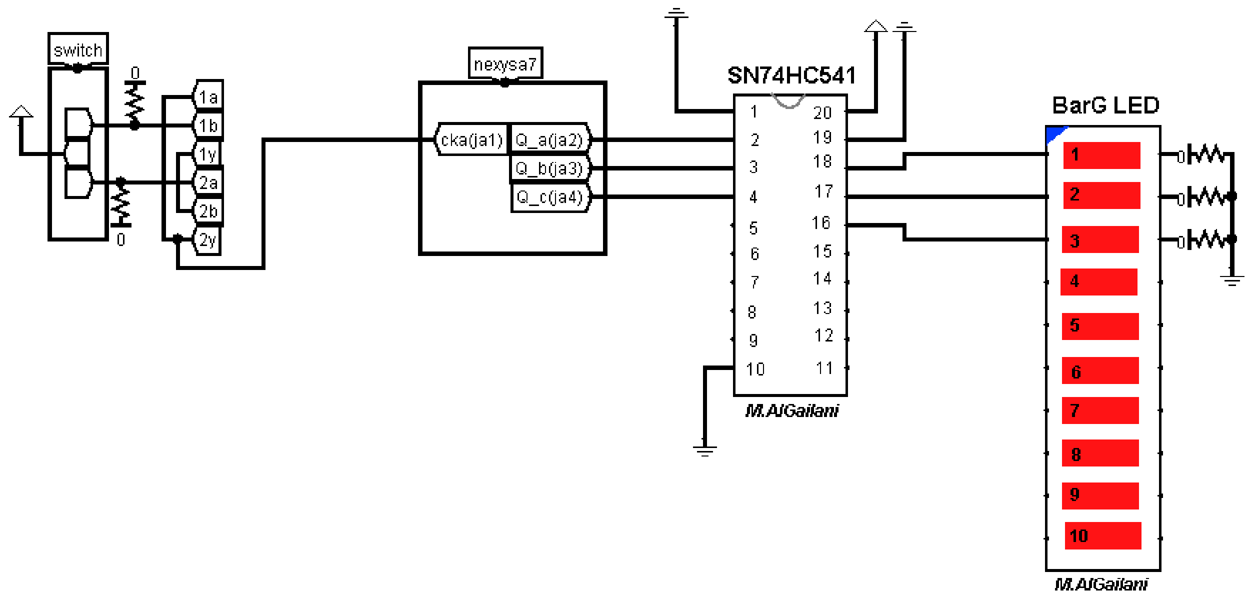
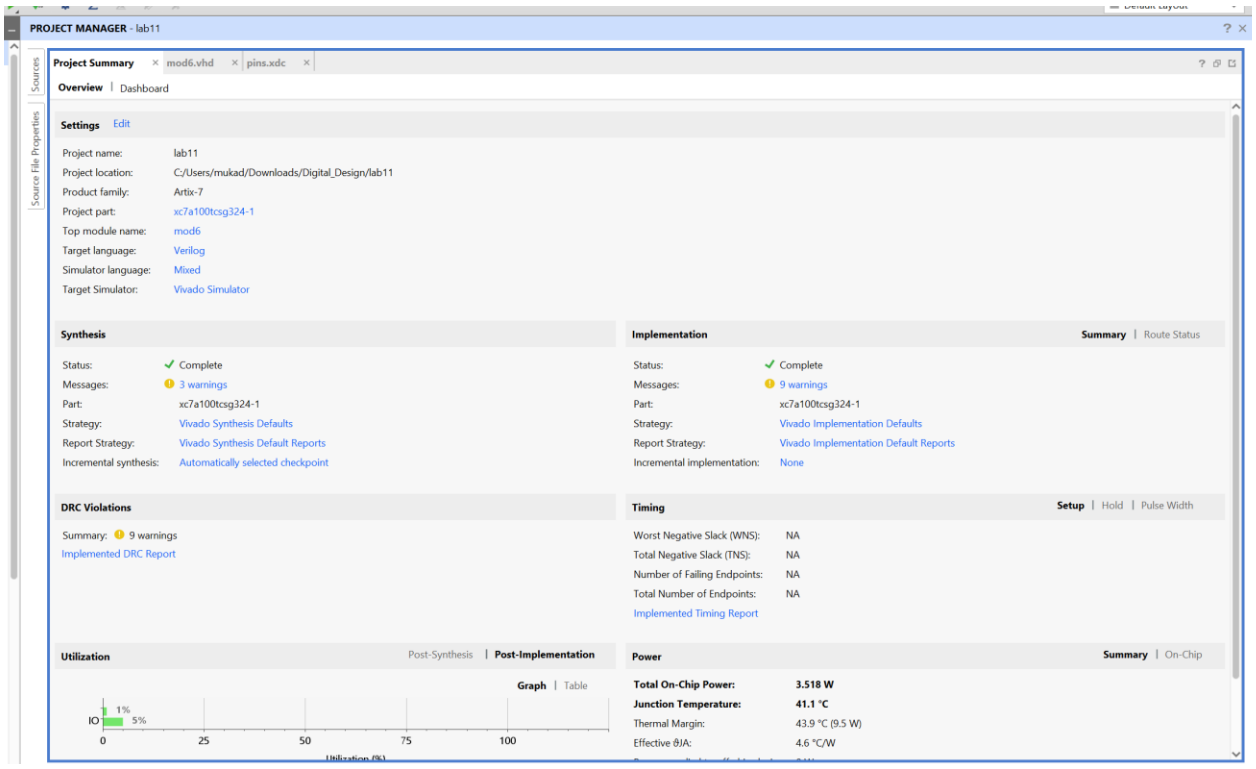


Fig 4:Breadboard schematic


PROJECT MANAGER - lab11

Project Summary | mod6.vhd | pins.xdc

Overview | Dashboard

Settings Edit

Project name:	lab11
Project location:	C:/Users/mukad/Downloads/Digital_Design/lab11
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	mod6
Target language:	Verilog
Simulator language:	Mixed
Target Simulator:	Vivado Simulator

Synthesis

Status:	✓ Complete
Messages:	3 warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Synthesis Defaults
Report Strategy:	Vivado Synthesis Default Reports
Incremental synthesis:	Automatically selected checkpoint

Implementation Summary | Route Status

Status:	✓ Complete
Messages:	9 warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

DRC Violations

Summary: 9 warnings

Implemented DRC Report

Timing Setup | Hold | Pulse Width

Worst Negative Slack (WNS):	NA
Total Negative Slack (TNS):	NA
Number of Failing Endpoints:	NA
Total Number of Endpoints:	NA

Implemented Timing Report

Utilization Post-Synthesis | Post-Implementation

Graph | Table

IO 1% 5%

Power Summary | On-Chip

Total On-Chip Power:	3.518 W
Junction Temperature:	41.1 °C
Thermal Margin:	43.9 °C (9.5 W)
Effective θJA:	4.6 °C/W

Fig 5: Project summary page

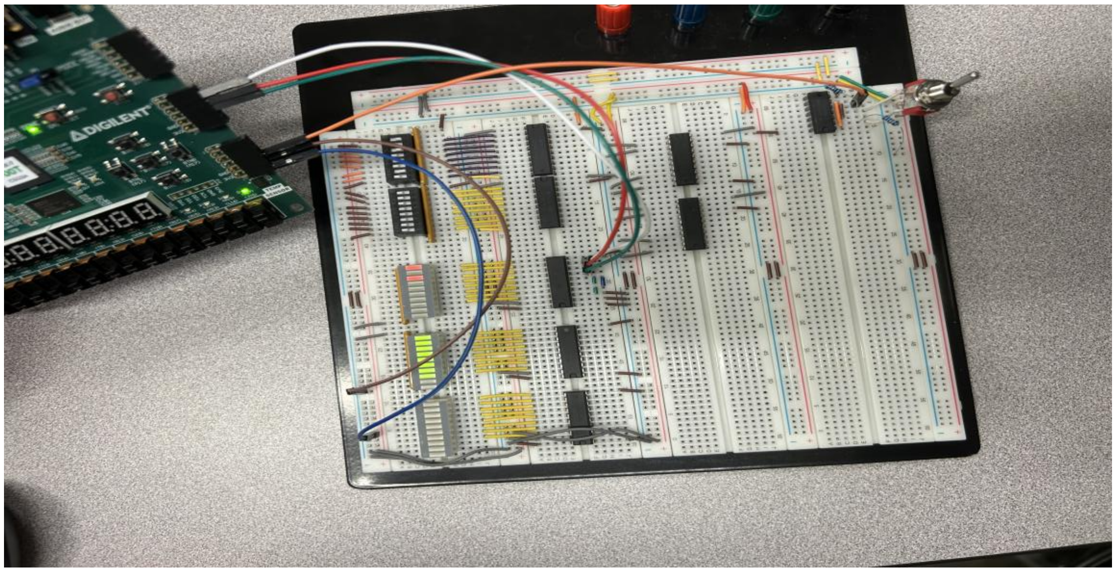


Fig 6: This shows when the R5 switch on the FPGA board is on the output 5.

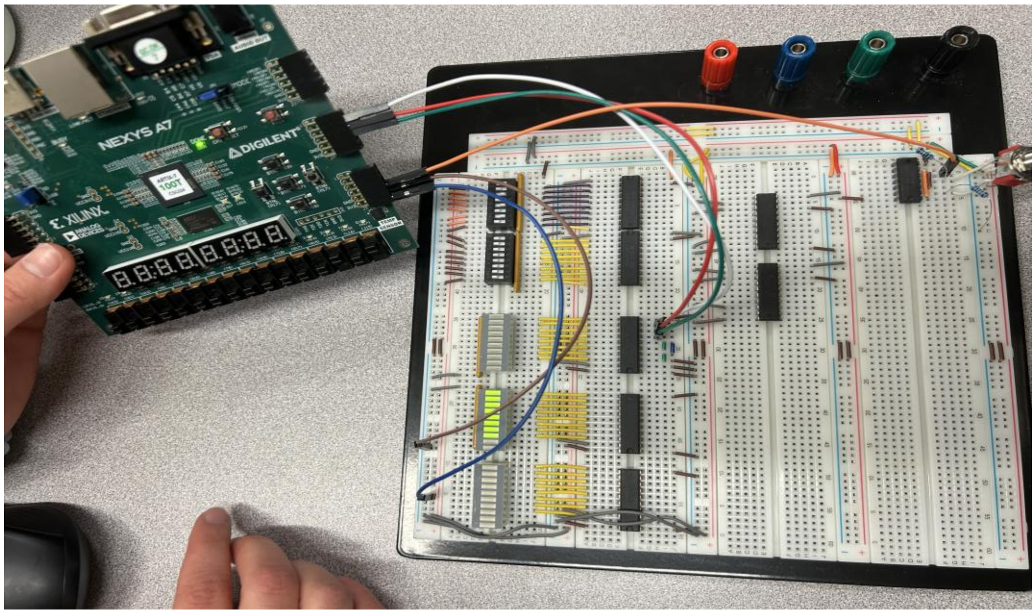


Fig 7: This shows when the R0 switch on the FPGA board is on the output 0.

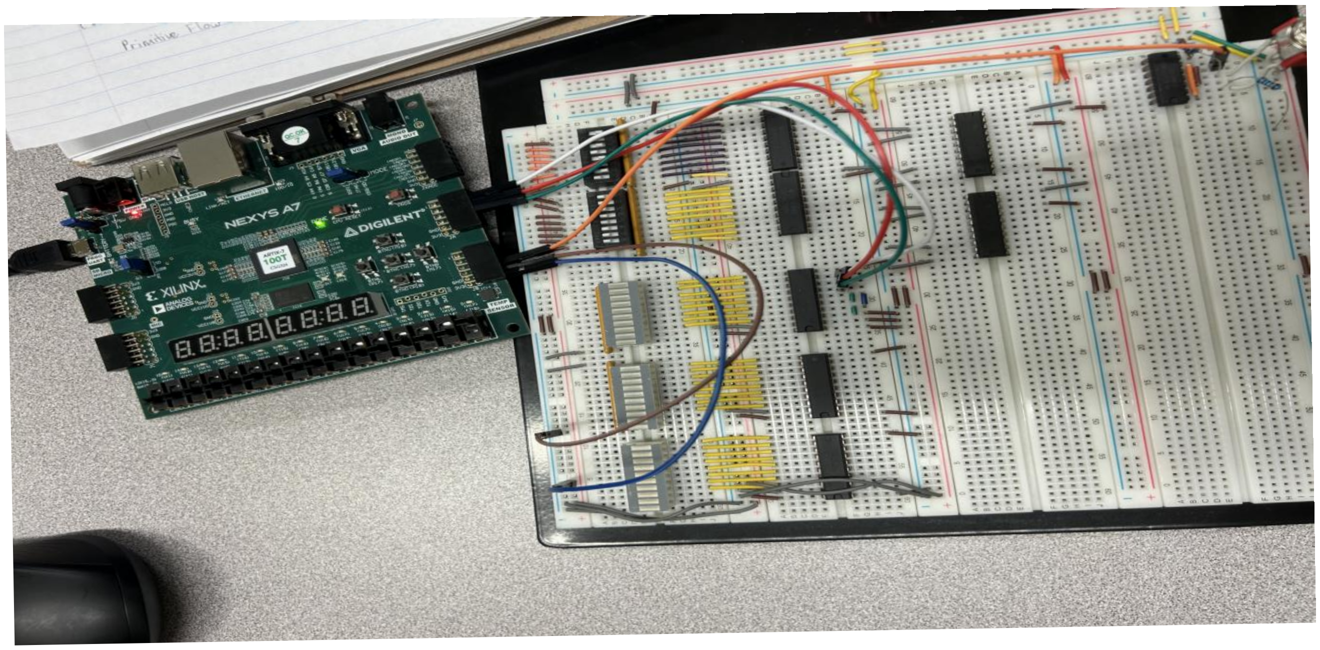


Fig 8: This shows the system in state A where the count is zero and none of the Y LEDS are on.

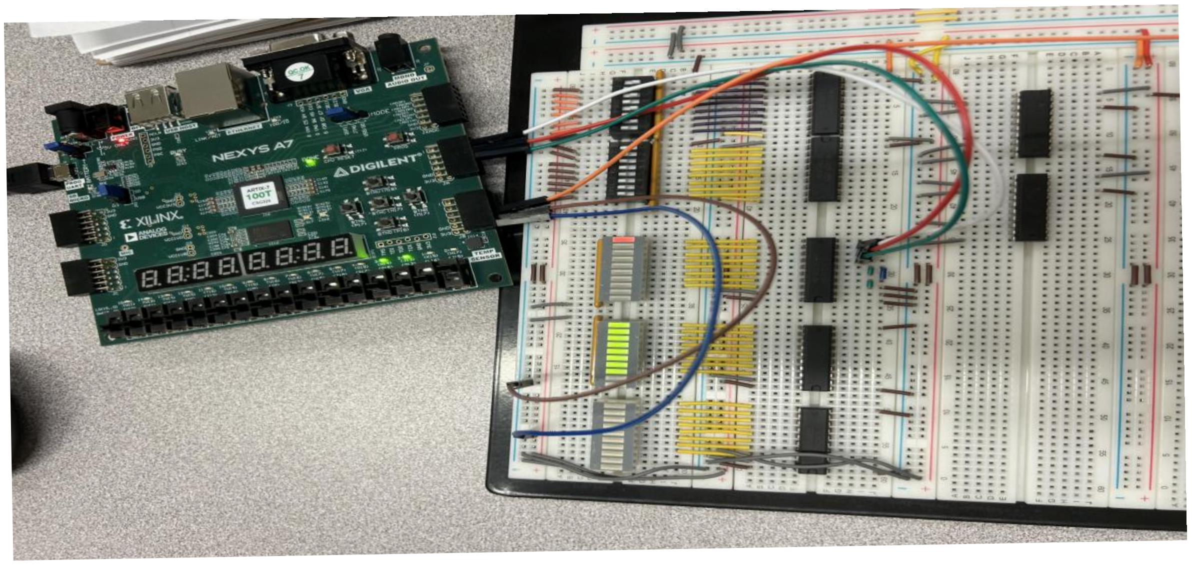


Fig 9: This shows the system in state C where the count is one and Y1 and Y2 LEDS are on.

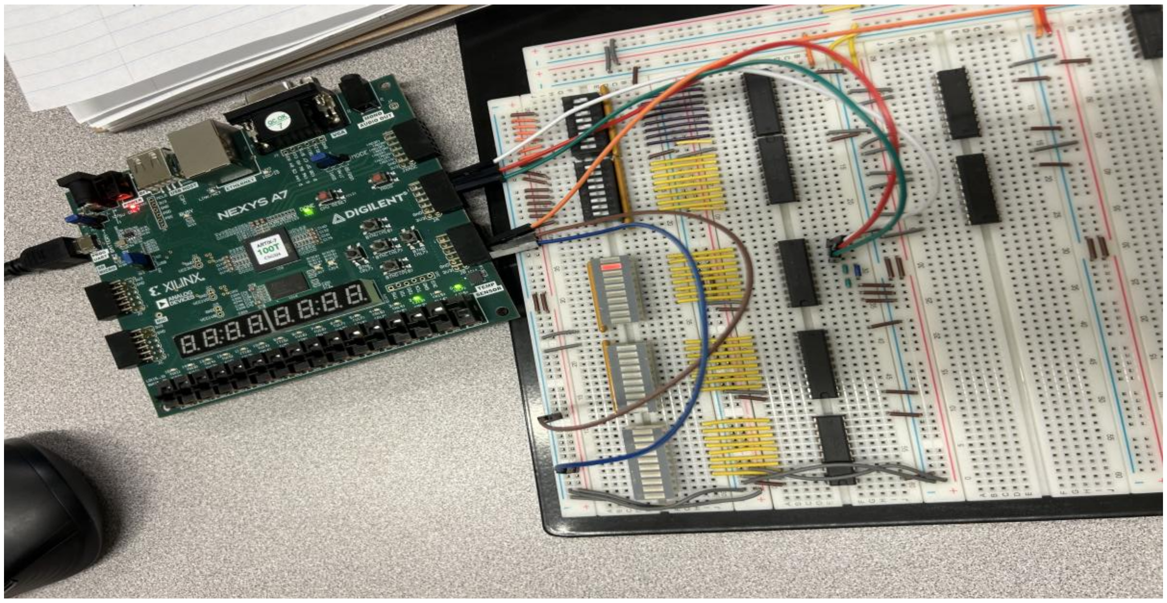


Fig 10: This shows the system in state E where the count is two and Y2 and Y4 LEDS are on.

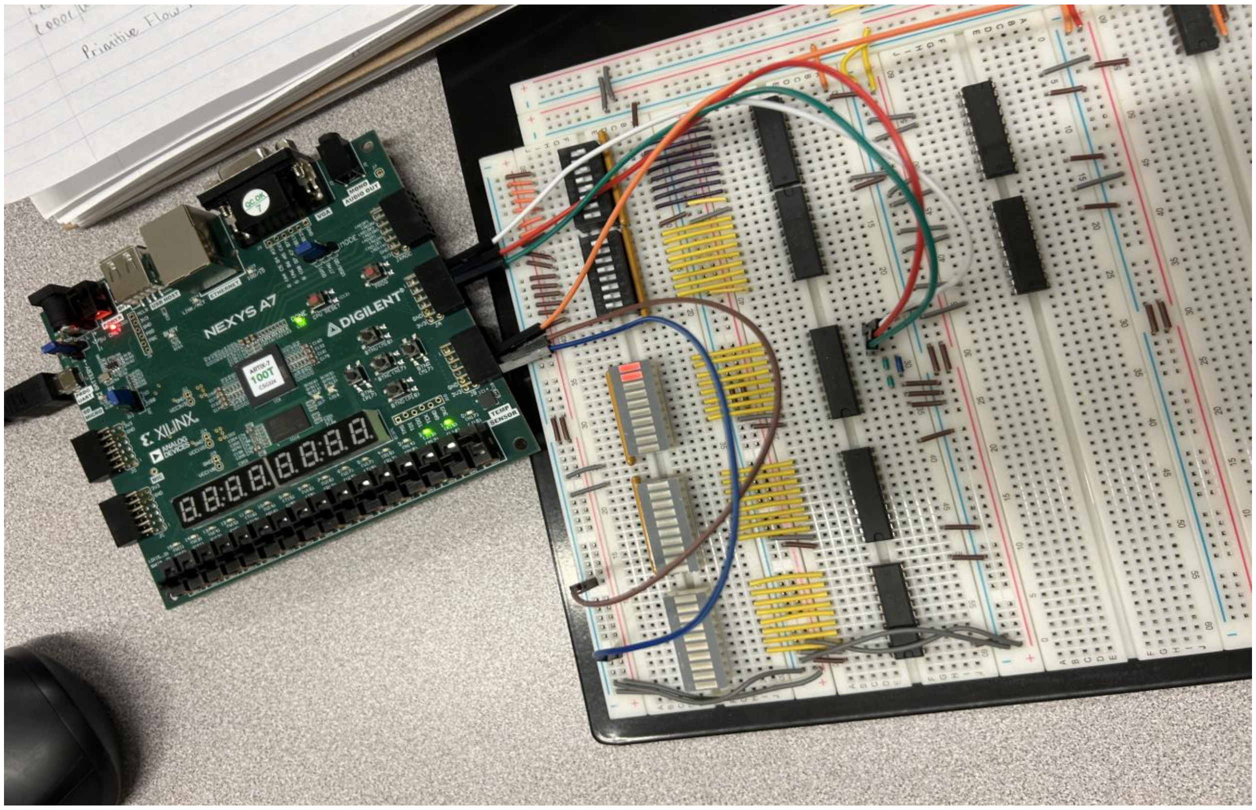


Fig 11: This shows the system in state F where the count is three and Y2 and Y3 LEDS are on.

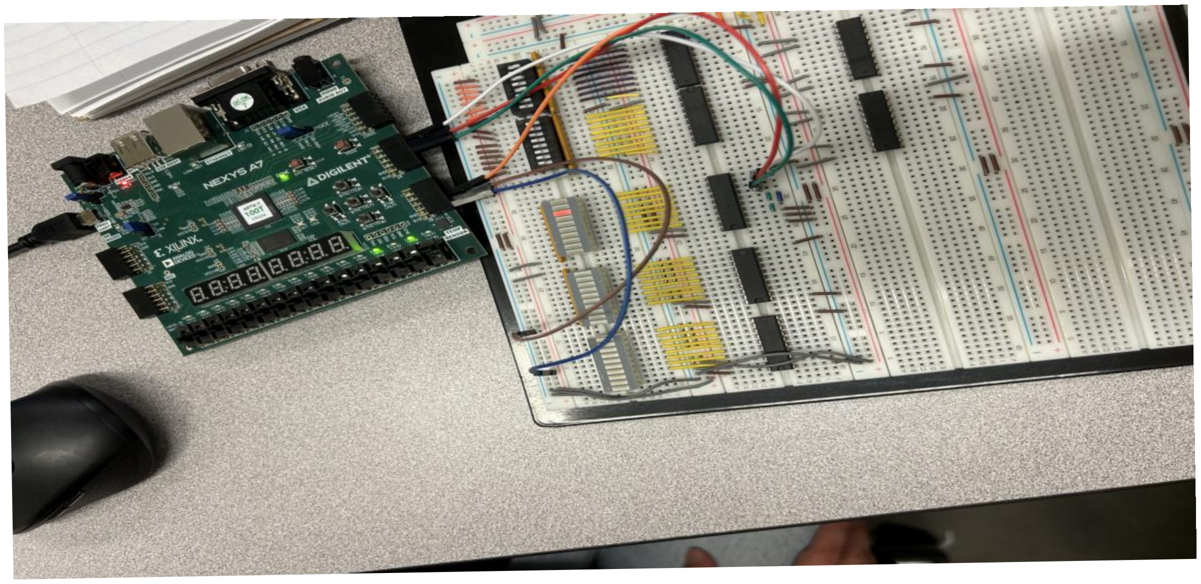


Fig 12: This shows the system in state G where the count is four and Y1 and Y3 LEDS are on

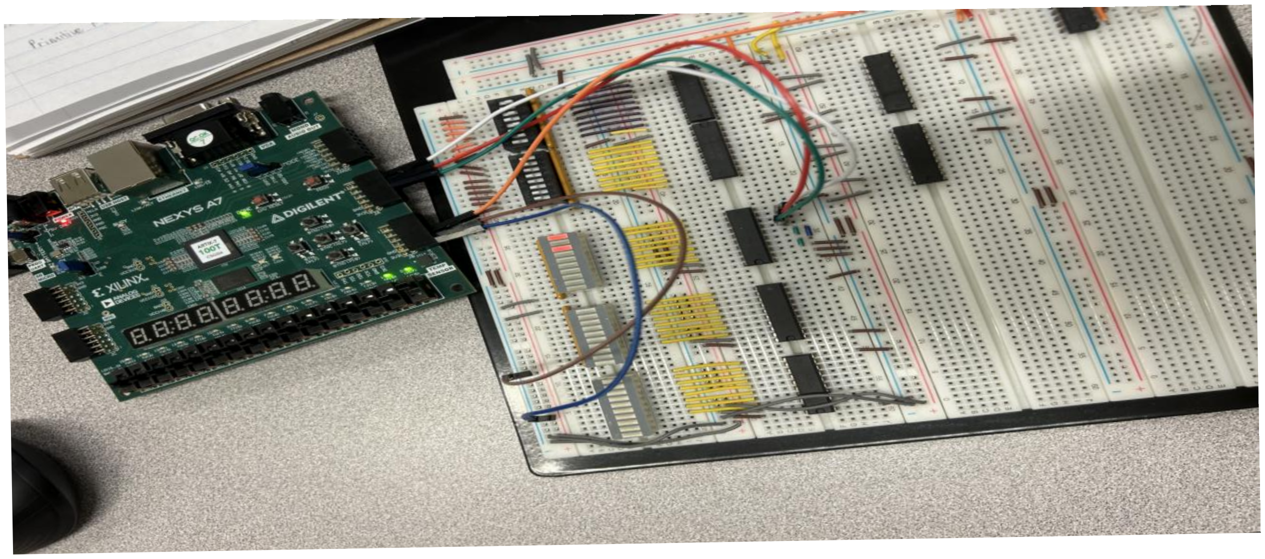


Fig 13: This shows the system in state I where the count is five and Y3 and Y4 LEDS are on

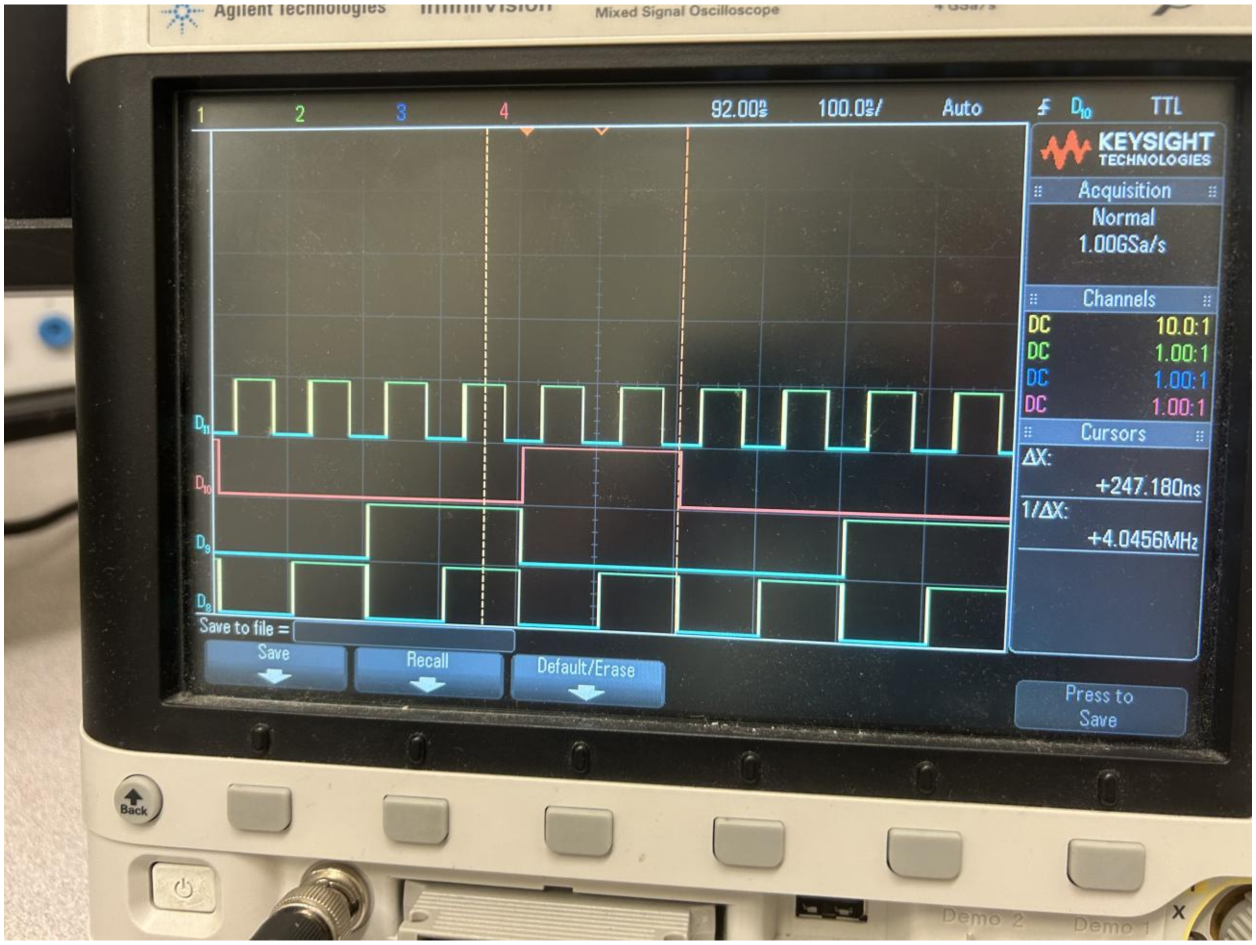


Fig 14: This shows the oscilloscope view. Channel D8 is Qa, D9 is Qb, D10 is Qc, and D11 is CKA (the photo we saved onto the flash drive saved as a text file for some reason so the only picture we have is of the oscilloscope sorry)

When we change the CKA to a button it immediately goes from state a to state e then from e to k. So, without using a bounce free switch it jumps states randomly.

Conclusion

This lab successfully demonstrated the design flow for implementing an asynchronous sequential counter based on the modified specifications of the SN7490A decade counter. Post-route simulations confirmed correct timing and correct state changes, verifying that the circuit operated as intended. Hardware was also used to verifiy the functionality of the circuit. Using DIP switches, bounce-free switches, push buttons, and LEDs, the real-time behavior of the counter was observed and verified. Overall, this lab provided valuable hands-on experience.

Appendix

Task 1

Vhdl

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 11/17/2025 06:47:40 PM
-- Design Name: 
-- Module Name: mod6 - Behavioral
-- Project Name: 
-- Target Devices: 
-- Tool Versions: 
-- Description: 
-- 
-- Dependencies: 
-- 
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-- 
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
--use IEEE.NUMERIC_STD.ALL;

entity mod6 is
    Port (
        CKA, RESET, R0, R5: in  STD_LOGIC;     -- falling-edge sensitive input clock
        y1,y2,y3,y4: inout std_logic;    --state variables
        QA, QB, QC  : out STD_LOGIC      --q outputs
    );
end mod6;

architecture Behavioral of mod6 is

begin

process(cka, reset,y1,y2,y3,y4,r0,r5)
begin

if r5 = '1' then  --making sure r5 has priority over r0
    y1 <= '0';     --this is not taken into the state asignment therefore it has its own code
    y2 <= '0';
    y3 <= '0';
    y4 <= '1';
    qa <= '1';
    qb <= '0';
    qc <= '1';
else
    if r0 = '1' then   --r0 = 1 will clear everything
        y1 <= '0';
        y2 <= '0';
        y3 <= '0';
        y4 <= '0';
        qa <= '0';
        qb <= '0';
        qc <= '0';
    else
        y1 <= ((not(cka) and y1 and not(y4))        -- i got the values from my state 8x4 
asignment kmap
                or (y1 and not(y2) and not(y3) and not(y4))
                or (y1 and y2 and y3 and not(y4))
                or (cka and not(y2) and not(y3) and not(y4))
                or (cka and y2 and y3 and not(y4))) and reset;
        y2 <= ((not(cka) and y1 and not(y3) and not(y4))
                or (y2 and not(y3) and not(y4))
                or (not(y1) and y2)
                or (cka and y2 and not(y4))) and reset;

        y3 <= ((cka and not(y1) and y2 and y4)
                or (not(y1) and y2 and y3)
                or (y3 and not(y4))
                or (not(cka) and not(y1) and not(y2) and y3)) and reset;

        y4 <= ((cka and not(y1) and not(y2) and y4)
                or (not(cka) and not(y1) and y2 and not(y3))
                or (not(y1) and y2 and not(y3) and y4)
                or (cka and not(y1) and y2 and y4)
                or (not(cka) and not(y1) and not(y2) and y3)
                or (not(y1) and not(y2) and y3 and y4)) and reset;

        QA <= ((not(y1) and not(y2) and y4)
                or (y2 and not(y4))) and reset;

        QB <= ((not(y1) and y2 and y4)
                or (y2 and y3)) and reset;

        QC <= ((not(y1) and not(y2) and y4)
                or (not(y2) and y3)) and reset;
    end if;
end if;
end process;

end Behavioral;
```

Tcl

```
restart

add_force R0 {0 0ns}
add_force R5 {0 0ns}
add_force CKA {0 0ns}
add_force RESET {0 0ns}
run 50ns

add_force RESET {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns
```

```
add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force CKA {1 0ns}
run 50ns

add_force CKA {0 0ns}
run 50ns

add_force R0 {1 0ns}
run 50ns

add_force R5 {1 0ns}
run 50ns
```

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 L16 IOSTANDARD LVCMOS33 } [get_ports { R5 }];
#IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [get_ports { R0 }];
#IO_L6N_T0_D08_VREF_14 Sch=sw[2]

## LEDs
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { y4 }];
#IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { y3 }];
#IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { y2 }];
#IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33 } [get_ports { y1 }];
#IO_L8P_T1_D11_14 Sch=led[3]

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*];
set_property ALLOW_COMBINATORIAL_LOOPS TRUE [get_nets y4*];

##Pmod Headers
##Pmod Header JA
#set_property -dict { PACKAGE_PIN C17 IOSTANDARD LVCMOS33 } [get_ports { CKA }];
#IO_L20N_T3_A19_15 Sch=ja[1]
set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { QA }];
#IO_L21N_T3_DQS_A18_15 Sch=ja[2]
set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { QB }];
#IO_L21P_T3_DQS_15 Sch=ja[3]
set_property -dict { PACKAGE_PIN G17 IOSTANDARD LVCMOS33 } [get_ports { QC }];
#IO_L18N_T2_A23_15 Sch=ja[4]
#set_property -dict { PACKAGE_PIN P17 IOSTANDARD LVCMOS33 } [get_ports { CKA }];
#IO_L12P_T1_MRCC_14 Sch=btnl
```