Lab 6 Report

ECE 4525

Jena Francis

Mack Usmanova

10/8/2025

Table of Contents

Introduction

The objective of this laboratory experiment was to develop and analyze synchronous sequential circuits using VHDL on the Nexys A7 FPGA board. Specifically, this lab focused on two key components: a bit-serial adder and an interface circuit for a 74HC299 shift register. The first task involved designing a Moore-type bit-serial adder circuit, implementing a bounce-free manual clock, and verifying its functionality through simulation and hardware testing. This demonstrated how sequential logic can be used to perform arithmetic operations serially using limited hardware resources.

In the second task, a VHDL program was written to control the 74HC299 shift register, which performed Parallel Load, Shift Right, and Hold operations. The circuit interfaced with external components such as DIP switches, LEDs, and push buttons to visualize and control its behavior. Both behavioral and post-route simulations were conducted to ensure that the design met functional and timing specifications. Overall, this lab provided hands-on experience with VHDL design flow, synchronization, and interfacing between FPGA logic and external digital components.

Procedure

Task 1.1

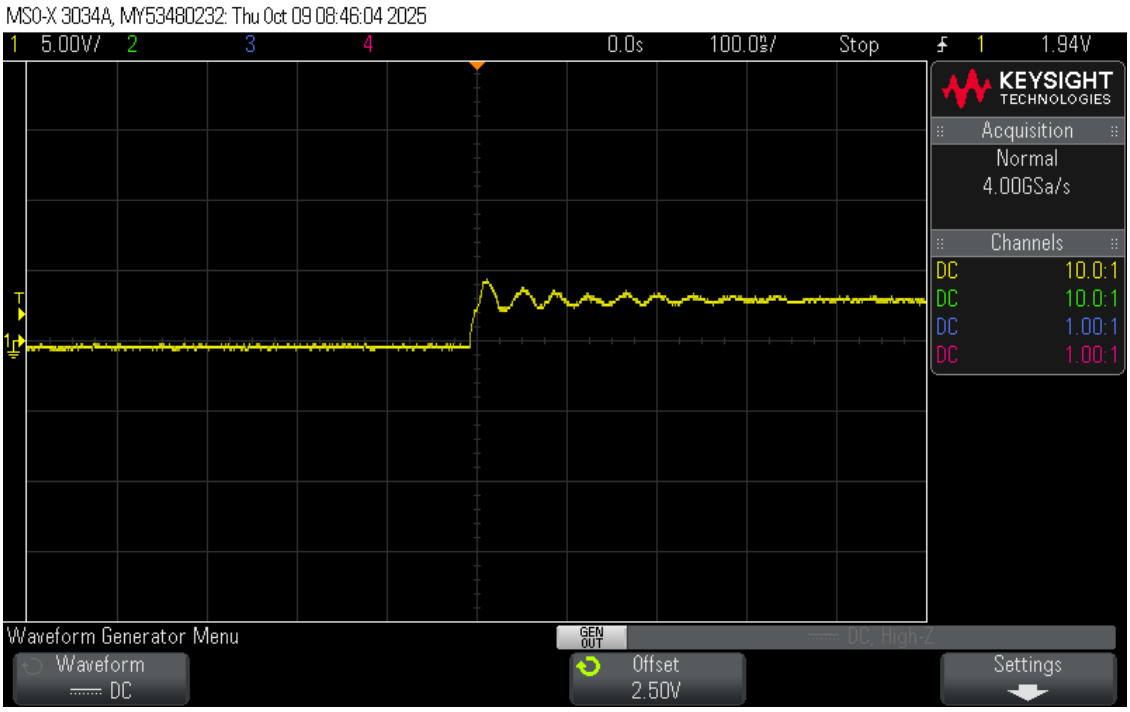

Fig 1: This shows the oscilloscope view that verifies the bounce free clock.

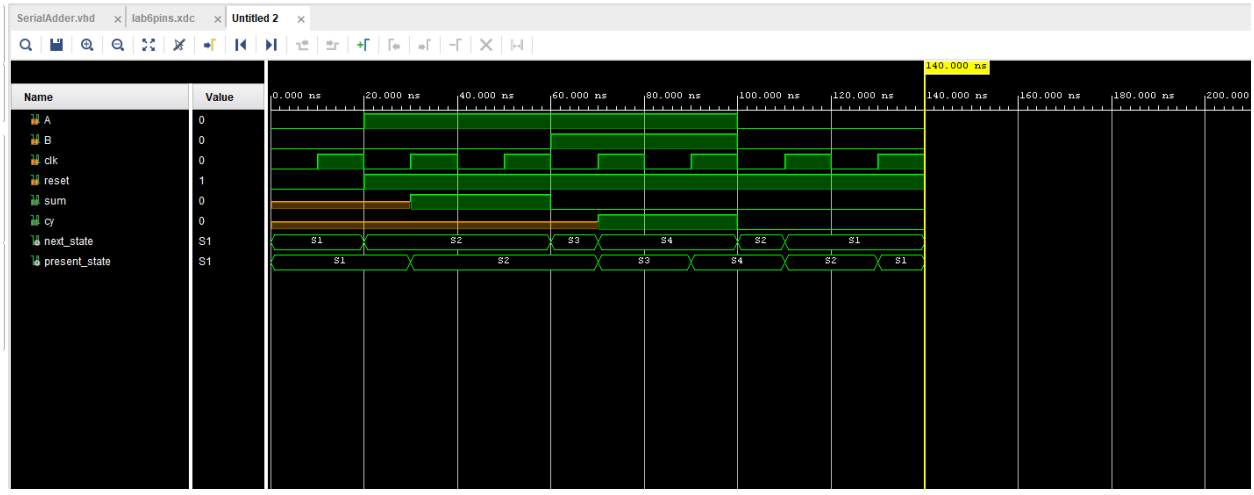

FIg 2: This shows the behavioral simulation. When reset is high, A is high and B is low, so once the clock is at the rising edge, it goes into state 2 where the sum is calculated. From there A and B are both high, so it moves to state 3 where the carry is shown. A and B are still high, so it moves to state 4 where both the carry and sum are shown.

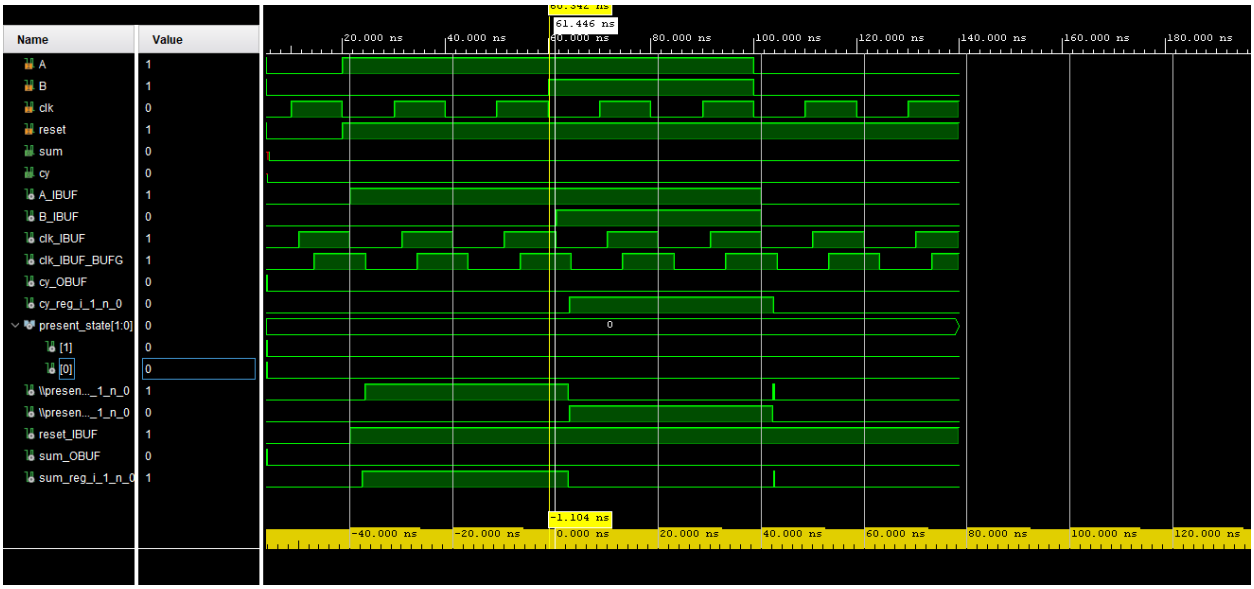

Fig 3: This is the Post route simulation with a propagation delay of 1.104ns.

Task 1.2

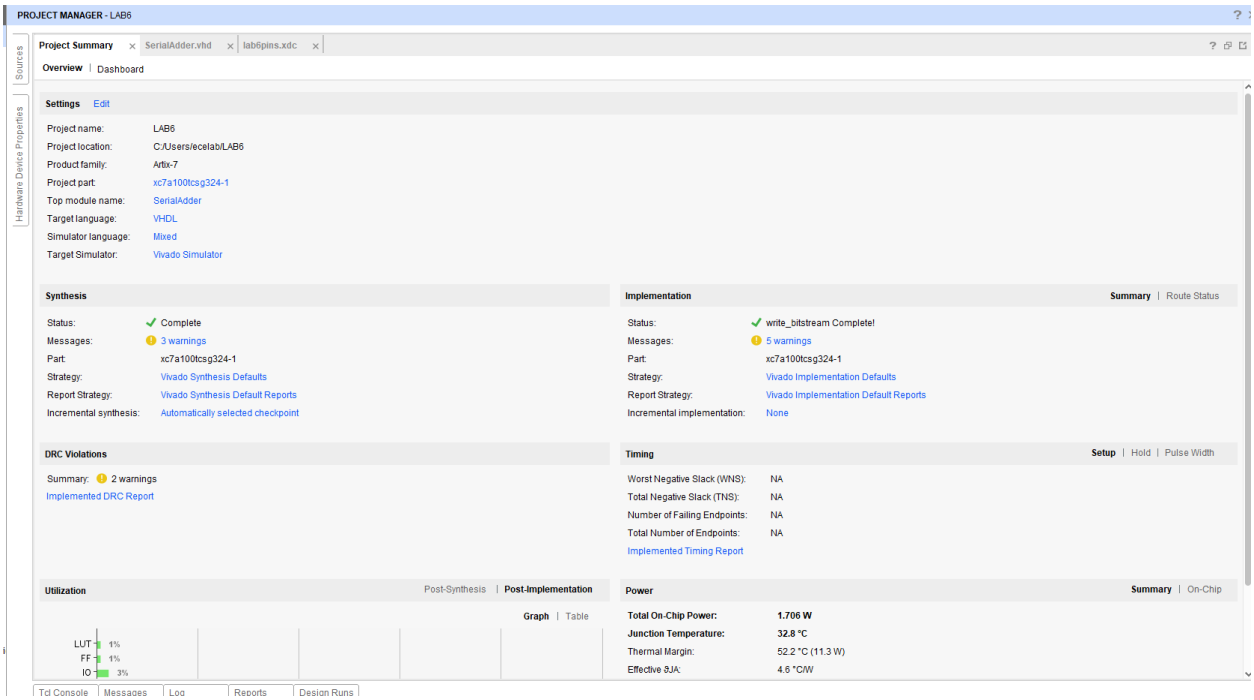

Fig 4: This shows the summary report.

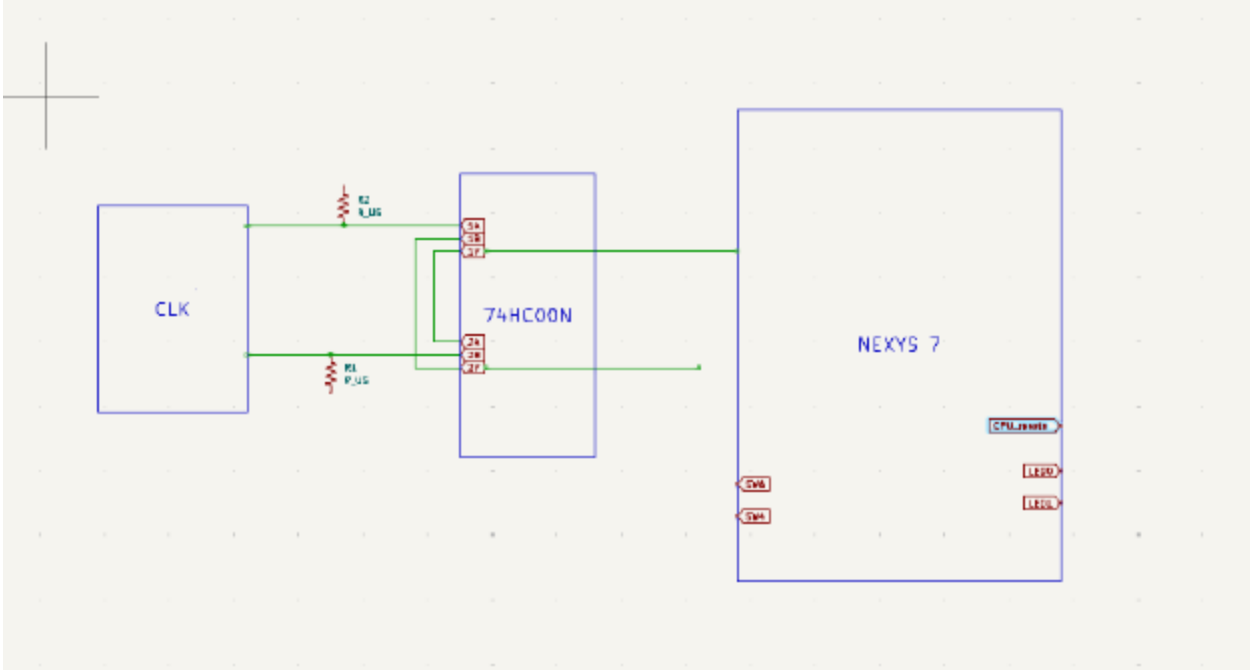

Fig 5: This is the schematic for task one. The bounce free clock circuit is connected to the board by the pmod header jb3. A and B are switches 4 and 6 on the board and cy and sum are outputed on LED[0] and LED[1].

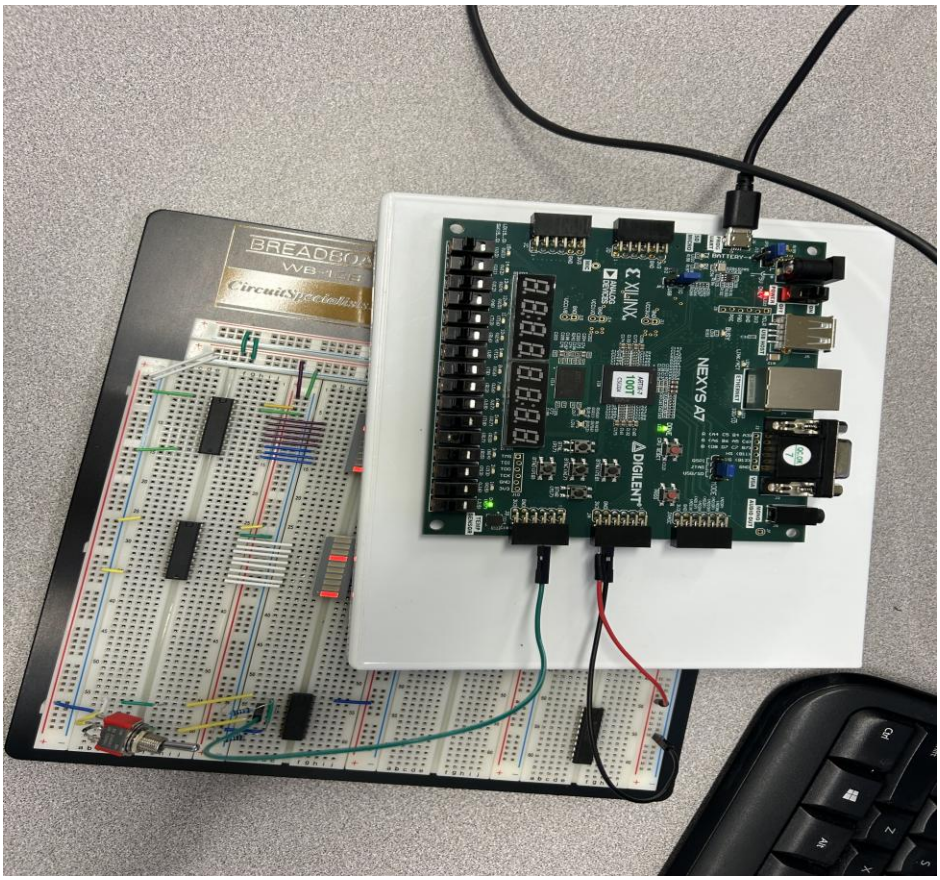

Fig 6: This shows the circuit. The bounce free clock is in the left corner. Here A is high and B is low, so it is in state 2 where the sum is outputted on LED[0].

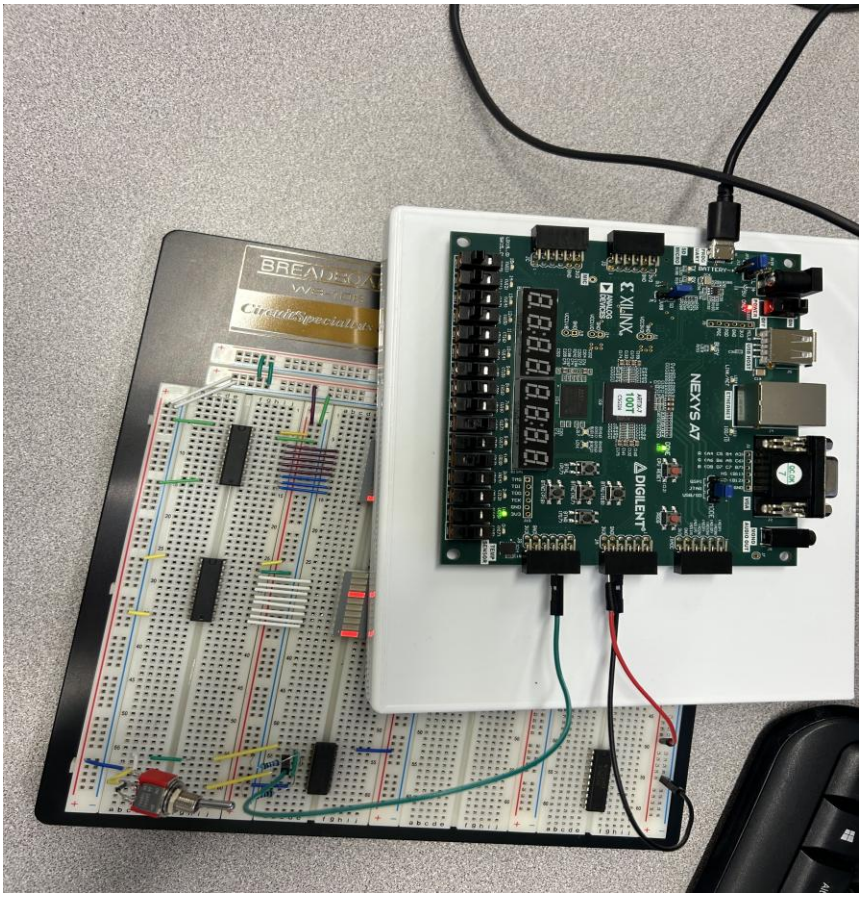

Fig 7: Here it is in state 3, A and B are high, so it is outputting the carry out on LED[1].

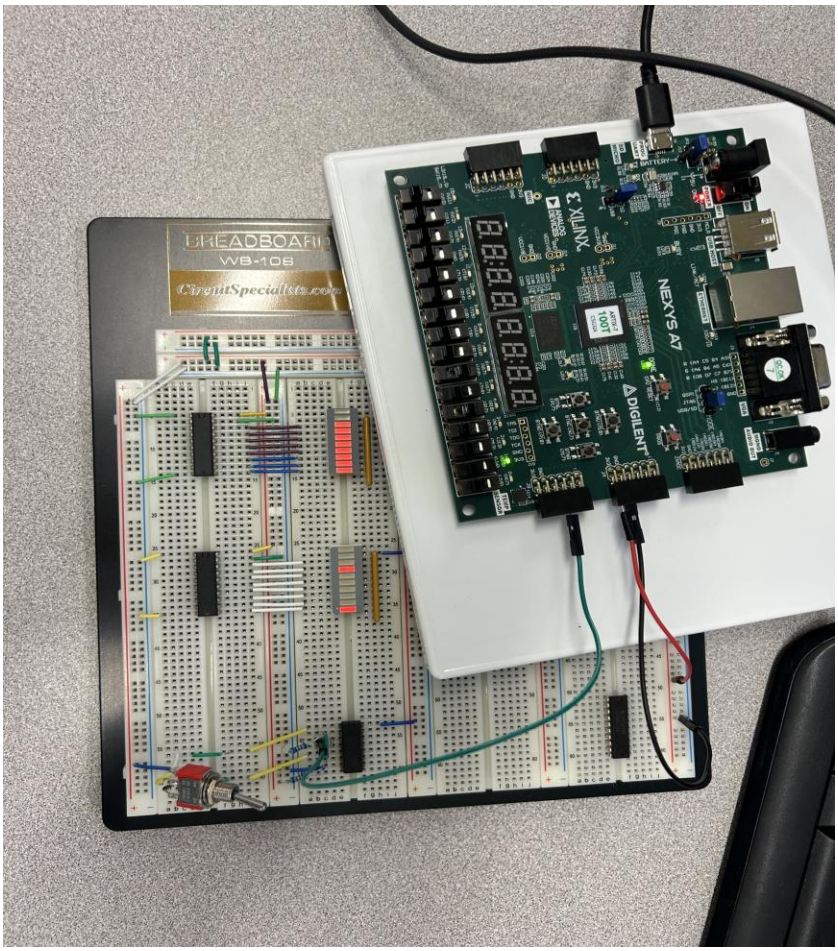

Fig 8: Here it is in state 4, A and B are high, so it shows the sum as 0 and the carry as 1.

Task 2.1

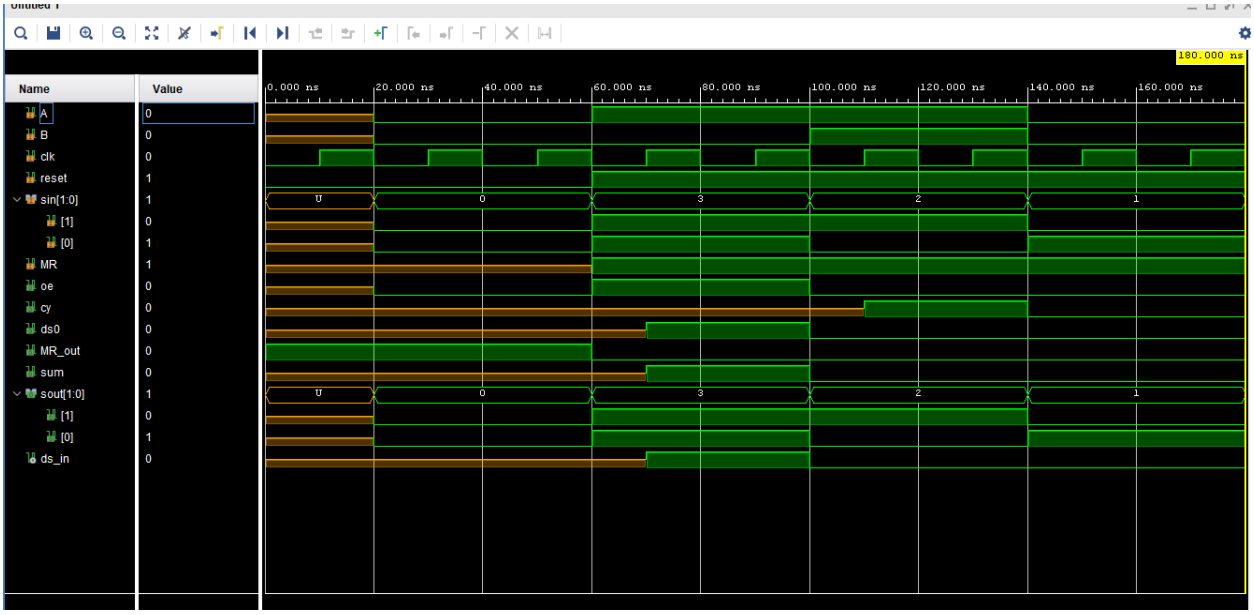

Fig 9: this is the behavioral simulation for task 2.

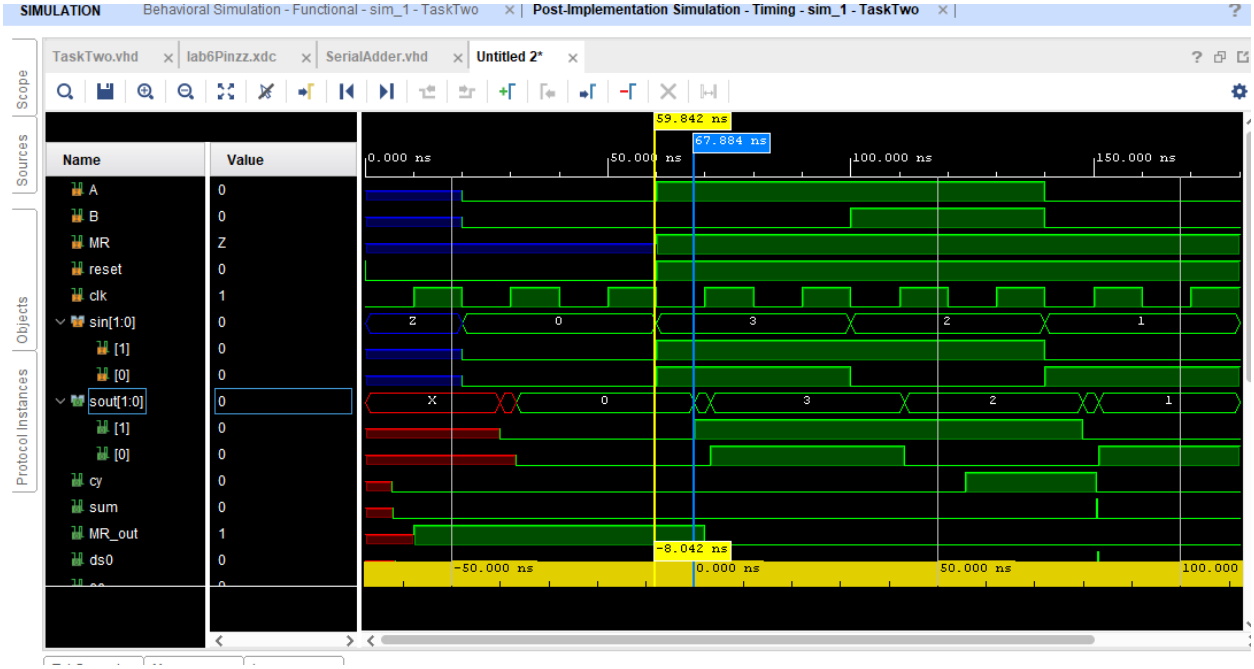

Fig 10: This is the post-route simulation for task two with a propagation delay of 8.042ns.

Task 2.2

Overview | Dashboard

Settings Edit

Project name:	Lab6
Project location:	C:/Users/jenaf/Digital D/Lab6
Product family:	Artix-7
Project part:	xc7a100tcsg324-1
Top module name:	TaskTwo
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:	✓ write_bitstream Complete!
Messages:	! 5 warnings
Part:	xc7a100tcsg324-1
Strategy:	Vivado Implementation Defaults
Report Strategy:	Vivado Implementation Default Reports
Incremental implementation:	None

Fig 11: this is the summary report for task 2.

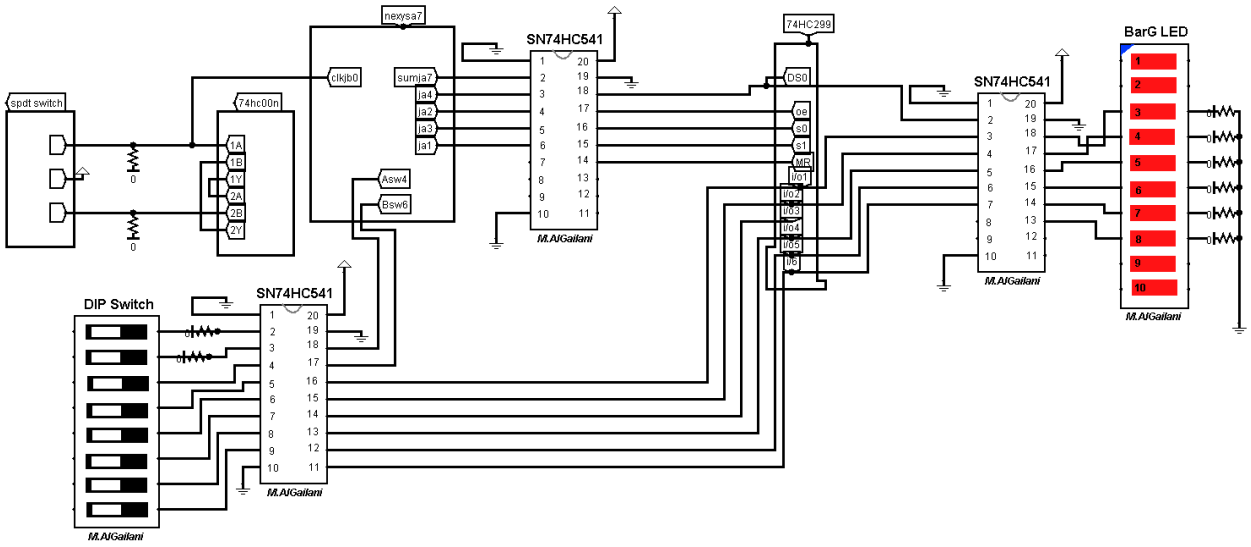

Fig12, breadboard schematic

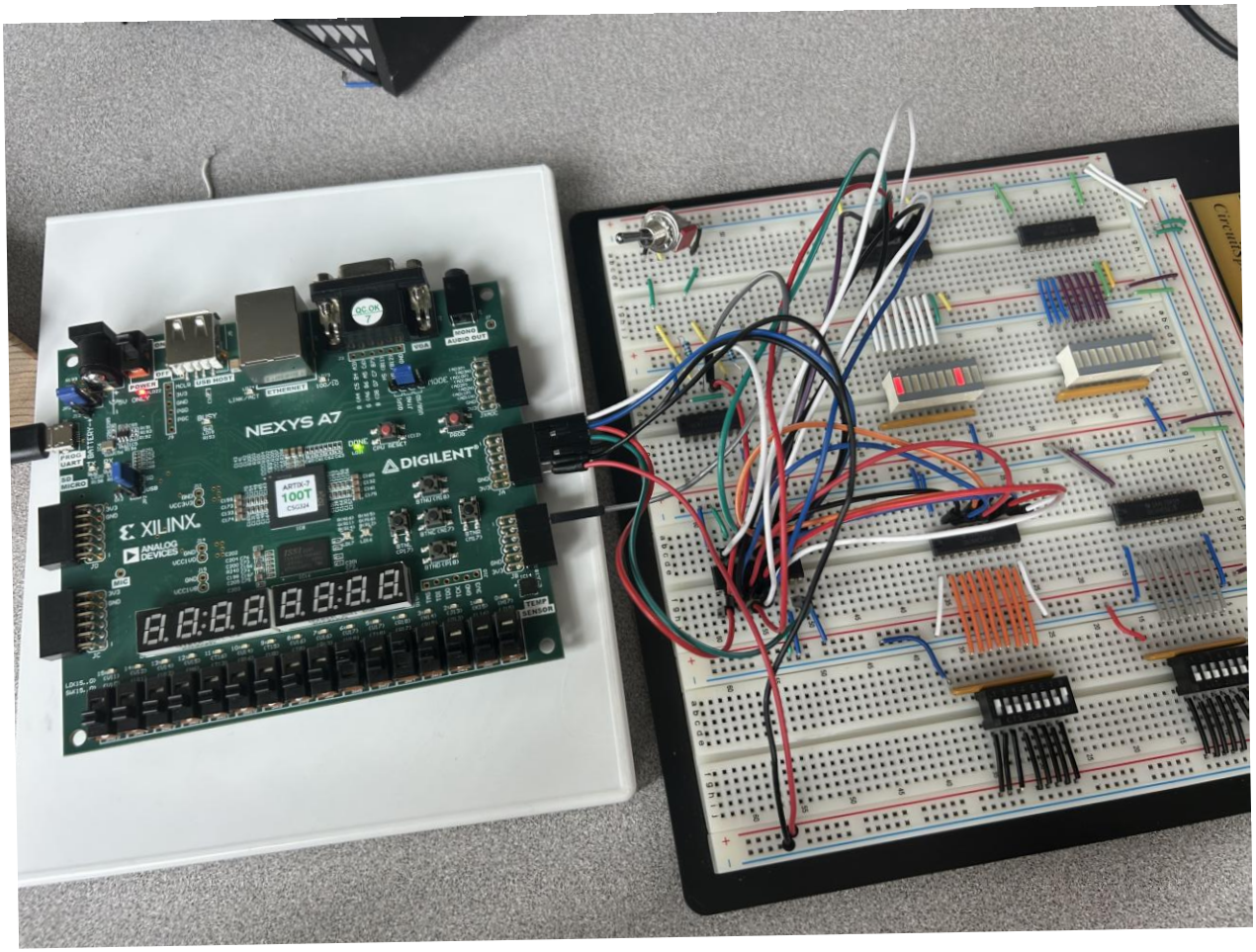

Fig 13: This shows the breadboard set up. I/O pins on the shift register are wired to both LEDs for outputs and dipswitches for inputs. S0 and S1 are switches 11 and 9 on the FPGA board. Here the dipswitches 0 and 7 are high and shown on the LEDs.

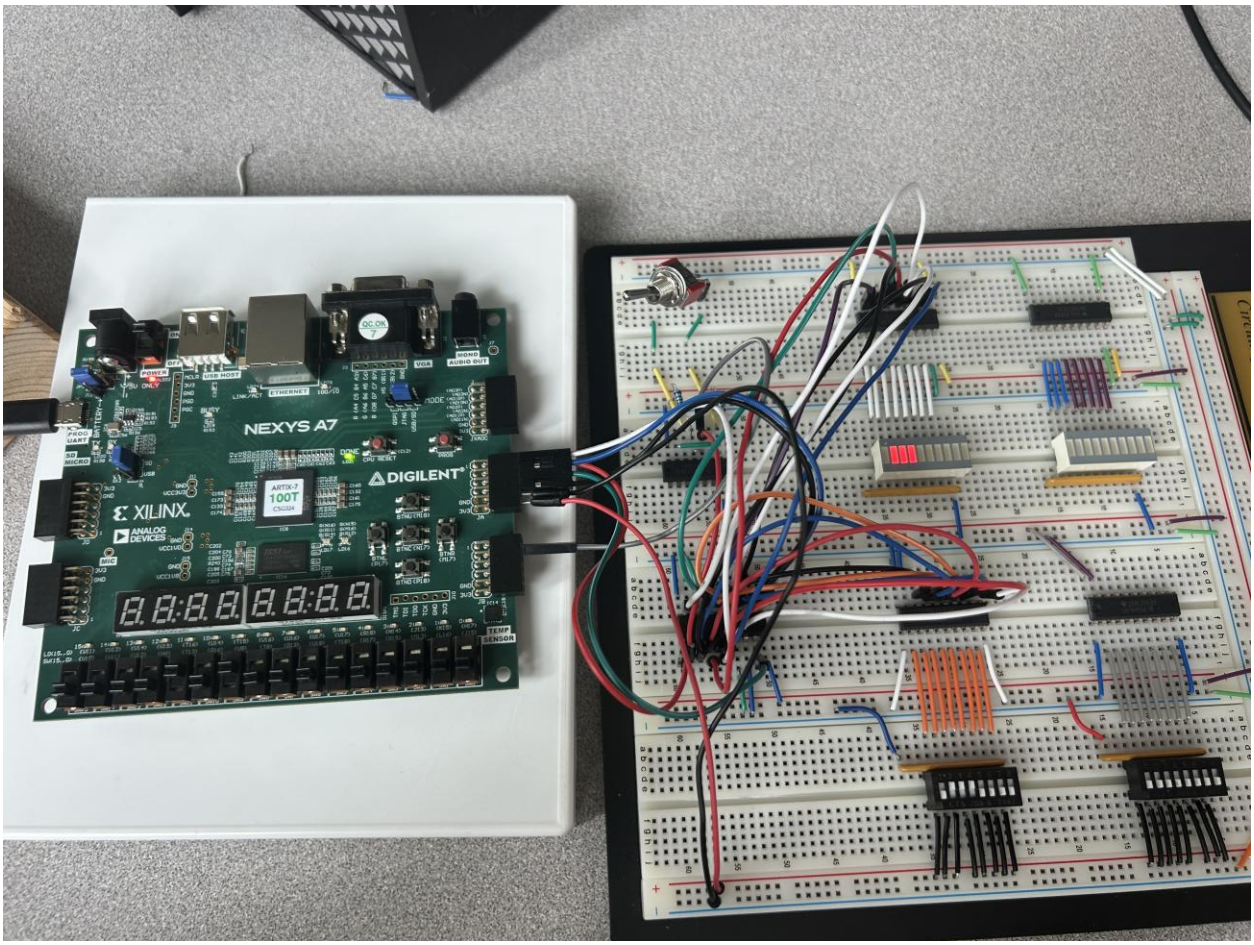

Fig 14: Here switches 1, 2, and 3 are high and are shown on the LEDs

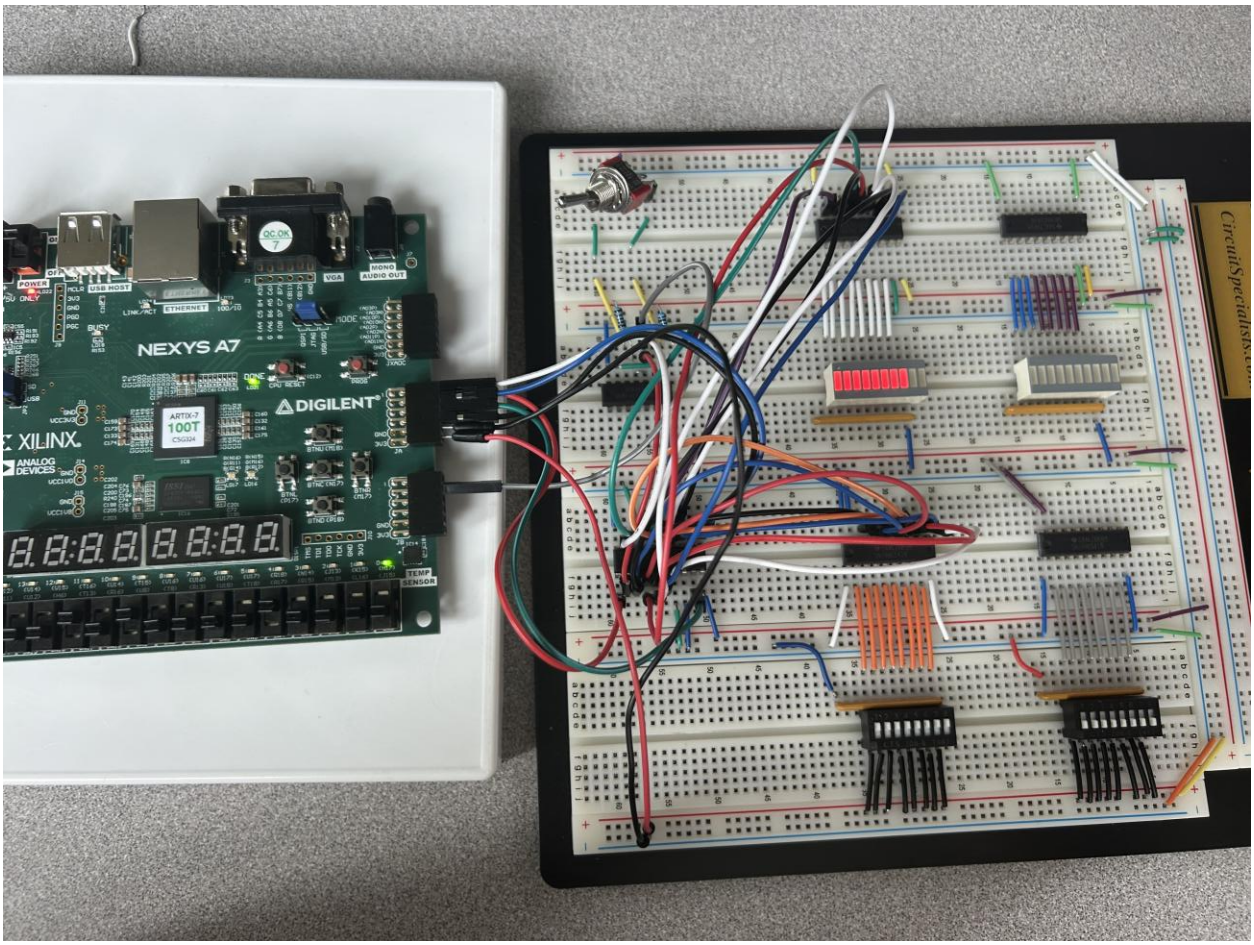

Fig 15: Here all dipswitches are high and are shown on the LEDs.

Conclusion

Through this experiment, the principles of synchronous sequential circuit design were successfully demonstrated using VHDL on the Nexys A7 FPGA platform. The bit-serial adder circuit verified the concept of serial data processing, where addition is performed one bit at a time under clock control. The 74HC299 shift register interface further illustrated how sequential logic can be expanded to implement data storage, shifting, and parallel loading functions.

The behavioral and post-route simulations, along with hardware verification, confirmed that both circuits operated as intended. This lab reinforced the importance of proper clock synchronization, timing analysis, and signal interfacing in digital system design. By completing these tasks, a deeper understanding was gained of how VHDL can be used to model, simulate, and implement real-time digital circuits in FPGA-based systems.

Appendix

Task one:

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;
-- Uncomment the following library declaration if instantiating -- any Xilinx leaf cells in this code. --library UNISIM; --use
UNISIM.VComponents.all;
entity SerialAdder is
Port ( A, B : in std_logic;
clk : in std_logic;
reset : in std_logic;
sum, cy : out std_logic);
end SerialAdder;
architecture Behavioral of SerialAdder is
 type state_type is (S1, S2, S3, S4);
signal next_state, present_state: state_type := S1;
begin
process (present_state, next_state, A, B)
begin
case present_state is
   when S1 =>
      if A ='1' AND B = '1' then
         next_state <= S3;
      elsif A = '1' AND B = '0' then
         next_state <= S2;
      elsif A = '0' AND B = '1' then
         next_state <= S2;
      else
         next_state <= S1;
      end if;




   when S2 =>
      sum <= A XOR B;
      if A ='1' AND B = '1' then
         next_state <= S3;
      elsif A = '1' AND B = '0' then
         next_state <= S2;
      elsif A = '0' AND B = '1' then
         next_state <= S2;
      else
         next_state <= S1;
      end if;



   when S3 =>
      cy <= (A and B);
      if A ='1' AND B = '1' then
         next_state <= S4;
      elsif A = '1' AND B = '0' then
         next_state <= S3;
      elsif A = '0' AND B = '1' then
         next_state <= S3;
      else
         next_state <= S2;
```

```
      end if;



   when S4 =>
      sum <= A xor B;
      cy <= (A and B);
      if A ='1' AND B = '1' then
         next_state <= S4;
      elsif A = '1' AND B = '0' then
         next_state <= S3;
      elsif A = '0' AND B = '1' then
         next_state <= S3;
      else
         next_state <= S2;
      end if;


  when others => next_state <= S1;
  end case;
  end process;



clk_process : process
begin
wait until (CLK'event and CLK = '1');
 if reset = '0' then
 present_state <= S1;
else present_state <= next_state;
end if;
end process clk_process;
end Behavioral;
```

Tcl

```
restart

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

add_force reset {0 0ns}

add_force A {0 0ns}

 add_force B {0 0ns}

run 20ns

add_force reset {1 0ns}

add_force A {1 0ns}

add_force B {0 0ns}
```

```
run 40ns

add_force A {1 0ns}

add_force B {1 0ns}

run 40ns

add_force A {0 0ns}

add_force B {0 0ns}

run 40ns
```

XDC

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets clk];

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

set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { sum }]; #IO_L18P_T2_A24_15 Sch=led[0]

set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { cy }]; #IO_L24P_T3_RS1_15 Sch=led[1]

set_property -dict { PACKAGE_PIN R17 IOSTANDARD LVCMOS33 } [get_ports { A }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]

set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 } [get_ports { B }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]

set_property -dict { PACKAGE_PIN C12 IOSTANDARD LVCMOS33 } [get_ports { reset }]; #IO_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn
```

Task Two

Vdhl

```
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 TaskTwo is Port ( A, B, clk, reset : in std_logic;
 sin : in std_logic_vector(1 downto 0);
MR : in std_logic;
oe : out std_logic;
cy, ds0, MR_out, sum : out std_logic;
sout : out std_logic_vector(1 downto 0));

end TaskTwo;
```

```
architecture Behavioral of TaskTwo is
signal ds_in : std_logic;

component SerialAdder
 Port ( A, B : in std_logic;
clk : in std_logic;
 reset : in std_logic;
sum, cy : out std_logic);

end component;

Begin

sout <= sin;
MR_out <= not reset;
oe <= sin(0) and sin(1);

SBA: SerialAdder port map(        --port map connection with serial adder
   clk => clk,
   reset => MR,
   A => A,
   B => B,
   cy => cy,
   sum => ds_in);



   ds0 <= ds_in;
   sum <= ds_in;




end Behavioral;
```

Tcl

```
restart

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

add_force reset {0 0ns}

 run 20ns

add_force A {0 0ns}

add_force B {0 0ns}

add_force sin {00 0ns}

 run 40ns

add_force reset {1 0ns}

add_force MR {1 0ns}
```

```
add_force A {1 0ns}

add_force B {0 0ns}

add_force sin {11 0ns}

run 40ns

add_force A {1 0ns}

add_force B {1 0ns}

add_force sin {10 0ns}

run 40ns

add_force A {0 0ns}

add_force B {0 0ns}

add_force sin {01 0ns}

run 40ns
```

XDC

```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets clk];

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

set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { sum }]; #IO_L18P_T2_A24_15 Sch=led[0]

set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { cy }]; #IO_L24P_T3_RS1_15 Sch=led[1]

set_property -dict { PACKAGE_PIN R17 IOSTANDARD LVCMOS33 } [get_ports { A }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]

set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 } [get_ports { B }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]

set_property -dict { PACKAGE_PIN C12 IOSTANDARD LVCMOS33 } [get_ports { reset }]; #IO_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn

set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 } [get_ports { MR_out }]; #IO_L9P_T1_DQS_14 Sch=btnc

set_property -dict { PACKAGE_PIN U8 IOSTANDARD LVCMOS18 } [get_ports { sin[0] }]; #IO_25_34 Sch=sw[9]

set_property -dict { PACKAGE_PIN T13 IOSTANDARD LVCMOS33 } [get_ports { sin[1] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]

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

set_property -dict { PACKAGE_PIN D18 IOSTANDARD LVCMOS33 } [get_ports { sout[0] }]; #IO_L21N_T3_DQS_A18_15 Sch=ja[2]
```

```
set_property -dict { PACKAGE_PIN E18 IOSTANDARD LVCMOS33 } [get_ports { sout[1] }]; #IO_L21P_T3_DQS_15 Sch=ja[3]

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

set_property -dict { PACKAGE_PIN D17 IOSTANDARD LVCMOS33 } [get_ports { ds0 }]; #IO_L16N_T2_A27_15 Sch=ja[7]
```