Mack Usmanova
Computer engineer
ECE4525
Homework2

Table of Contents

Project summary report

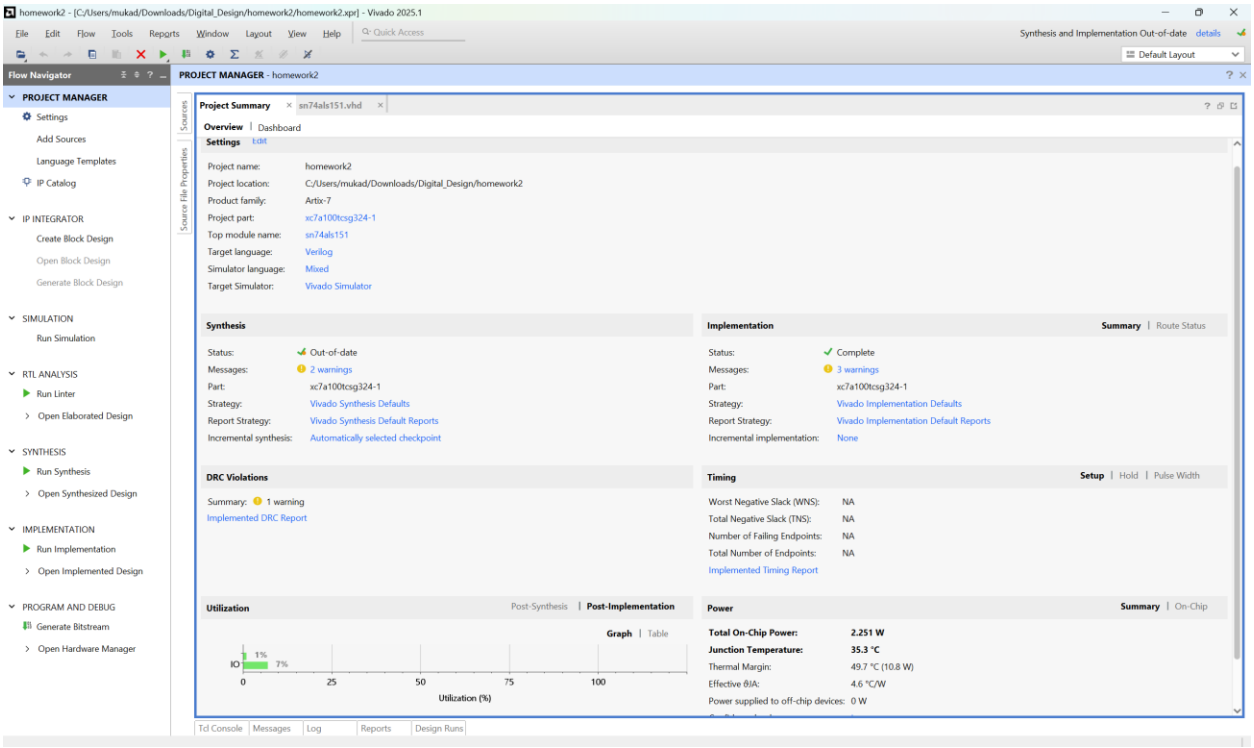

Figure1: the project summary report screenshot

This assignment successfully gave me an individual hands on experience with using resources given in eLearning and the homework specifications to create a VHDL code and an XDC pins assignment code. This homework introduced me to case statements and gave me an opportunity to practice process statements in VHDL. The challenges faced were figuring out how to equate W to not Y without assigning it to not the output of Y. I tried using signals and components to see if a shorter code is possible but it only made my code longer. I ended up reassigning W to an output each time I assigned Y to an output all in one source code with no components. I also faced a challenge figuring out if I can have a case statement inside an if statement. Turns out I can.

Tcl

```
restart
add_force CBA {000 0ns}
add_force Gnot {1 0ns}
add_force D {10101010 0ns}
run 100ns

add_force CBA {000 0ns}
add_force Gnot {0 0ns}
run 20ns
```

```
add_force CBA {001 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {010 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {011 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {100 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {101 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {110 0ns}
add_force Gnot {0 0ns}
run 20ns

add_force CBA {111 0ns}
add_force Gnot {0 0ns}
run 20ns
```

I made my tcl the exact way the function table showed its tests. It was easy because which D is being displayed depended on what number CBA is

Xdc

```
## This file is a general .xdc for the Nexys A7-100T
##Switches
set_property -dict { PACKAGE_PIN R17 IOSTANDARD LVCMOS33 } [get_ports { Gnot }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18 IOSTANDARD LVCMOS33 } [get_ports { CBA[2] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 } [get_ports { CBA[1] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13 IOSTANDARD LVCMOS33 } [get_ports { CBA[0] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8 IOSTANDARD LVCMOS18 } [get_ports { D[7] }]; #IO_L24N_T3_34 Sch=sw[8]
set_property -dict { PACKAGE_PIN U8 IOSTANDARD LVCMOS18 } [get_ports { D[6] }]; #IO_25_34 Sch=sw[9]
set_property -dict { PACKAGE_PIN R16 IOSTANDARD LVCMOS33 } [get_ports { D[5] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
set_property -dict { PACKAGE_PIN T13 IOSTANDARD LVCMOS33 } [get_ports { D[4] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
set_property -dict { PACKAGE_PIN H6 IOSTANDARD LVCMOS33 } [get_ports { D[3] }]; #IO_L24P_T3_35 Sch=sw[12]
set_property -dict { PACKAGE_PIN U12 IOSTANDARD LVCMOS33 } [get_ports { D[2] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
set_property -dict { PACKAGE_PIN U11 IOSTANDARD LVCMOS33 } [get_ports { D[1] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
set_property -dict { PACKAGE_PIN V10 IOSTANDARD LVCMOS33 } [get_ports { D[0] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

## LEDs
set_property -dict { PACKAGE_PIN V12 IOSTANDARD LVCMOS33 } [get_ports { W }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
set_property -dict { PACKAGE_PIN V11 IOSTANDARD LVCMOS33 } [get_ports { Y }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]
```

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 sn74als151 is
  Port (
  CBA: IN std_logic_vector(2 downto 0);
  Gnot: IN std_logic;
  D: IN std_logic_vector(7 downto 0);
  Y, W: OUT std_logic
  );
end sn74als151;

architecture Behavioral of sn74als151 is
begin
GNOTT: process (Gnot,CBA)
        begin
            if ((Gnot)='1') then
                Y <= '0';
                w <= '1';
            else
            case CBA is
                when "000" => y <= D(0);
                W <= not D(0);
                when "001" => y <= D(1);
                w <= not D(1);
                when "010" => y <= D(2);
                w <= not D(2);
                when "011" => y <= D(3);
                w <= not D(3);
                when "100" => y <= D(4);
                w <= not D(4);
                when "101" => y <= D(5);
                w <= not D(5);
                when "110" => y <= D(6);
                w <= not D(6);
                when "111" => y <= D(7);
                w <= not D(7);
                when others => Y <= '0';
            end case;
       end if;
end process GNOTT;

end Behavioral;
```

Behavioral simulation

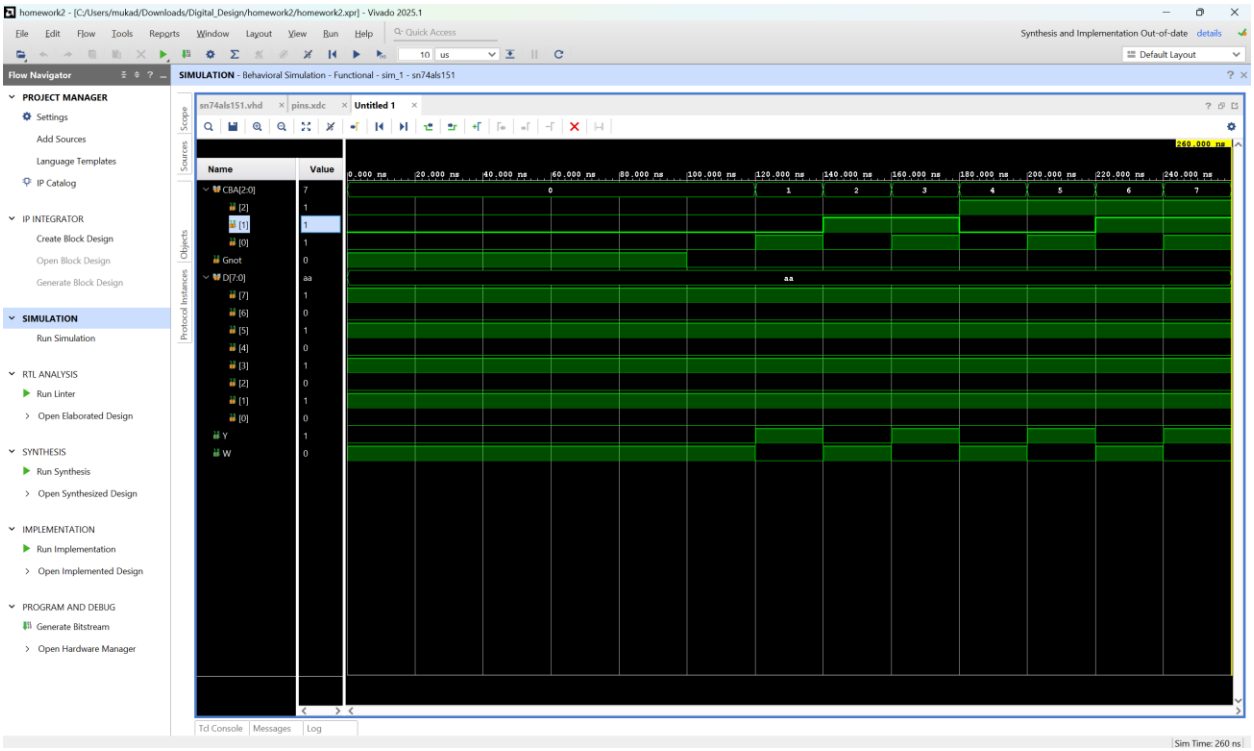

Figure 2: behavioral simulation shows the outputs Y and W changing depending on the input in CBA changing. I made the D in put high for each odd number of D and the CBA input the same as the function table in the data sheet given (increasing from 000 to 111). Therefore you can see the Y and W switching after each CBA increase.

'151A, 'LS151, 'S151

FUNCTION TABLE

INPUTS				OUTPUTS	
SELECT			STROBE		
C	B	A	$\overline{G}$	Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$