Lab 2: part 4

**Final purpose: Turing a 6 bit binary number to a decimal number in a seven segment display.**

**VHDL DESCRIPTION**

In the beginning it is necessary to include also the ieee\_numeric\_std library , as it is needed to compute subtraction in binary numbers ( not possible std\_logic\_vector library ) primarily used to find the units digit of our decimal number.

The entity of the our project is :

entity binarytoBCD is

port(

sw : in std\_logic\_vector (5 downto 0);

HEX1 : out std\_logic\_vector(6 downto 0);

HEX0: out std\_logic\_vector(6 downto 0)

);

end binarytoBCD;

Where the sw is our input and HEX1 and HEX0 are the outputs, respectively used for the decimals digit and units digit representation of the decimal number in the seven segment display. Note: all the about I/O -s are std\_logic vectors.

The outputs and inputs are port mapped by the file already given which is included in the project.

The work accomplished by using the component define in part three of the same lab:

Named bcd\_to\_7s , whose entity declared in vhdl is as follows:

entity bcd\_to\_7s is

port( bcd: in std\_logic\_vector (5 downto 0);

segments: out std\_logic\_vector (6 downto 0));

end bcd\_to\_7s;

The component was declared in the main file in the following way:

component bcd\_to\_7s is

port(

bcd: in std\_logic\_vector (5 downto 0);

segments: out std\_logic\_vector (6 downto 0)

);

The core of the computation is done in the architecture part of the project , where there were used three signals namely : sw\_u , dec , unit for the respective sw, HEX1 and HEX0 I/O -s of the entity.

Then the real computation part begins:

begin

--Input signal cast

sw\_u <= signed(sw);

--Component instantiation

converter\_dec:bcd\_to\_7s port map(dec,HEX1);

converter\_unit:bcd\_to\_7s port map(unit,HEX0);

--Decimal digit assignment

dec <="000000" when sw\_u<10 else

"000001" when sw\_u>9 and sw\_u<20 else

"000010" when sw\_u>19 and sw\_u<30 else

"000011" when sw\_u>29 and sw\_u<40 else

"000100" when sw\_u>39 and sw\_u<50 else

"000101" when sw\_u>49 and sw\_u<60 else

"000110";

----Units digit assignment

unit <= std\_logic\_vector(sw\_u - 0) when sw\_u<10 else

std\_logic\_vector(sw\_u - 10) when sw\_u>9 and sw\_u<20 else

std\_logic\_vector(sw\_u - 20) when sw\_u>19 and sw\_u<30 else

std\_logic\_vector(sw\_u - 30) when sw\_u>29 and sw\_u<40 else

std\_logic\_vector(sw\_u - 40) when sw\_u>39 and sw\_u<50 else

std\_logic\_vector(sw\_u - 50) when sw\_u>49 and sw\_u<60 else

std\_logic\_vector(sw\_u - 60);

end Behavioral;

The first action overtaken was the casting of the std\_logic vector into a signed vector.

As with this now it is possible to perform a subtraction operation to find the units digit.

The assignments were performed using the conditional assignment with the when-else structure. As it is seen all the assignments are coherent ones.

**TESTBENCH**

Finally, part of the project is also the testbench file used for simulation.

As with the vhdl description there were include obviously the same libraries.

Then comes the entity part which is empty inside:

--Entity definition

entity binarytoBCD\_tb is

end binarytoBCD\_tb;

The architecture part contains the following signals:

--Signals: testbench signals for the inputs and the outputs

signal sw\_tb : std\_logic\_vector(5 downto 0);

signal HEX1\_tb , HEX0\_tb : std\_logic\_vector(6 downto 0);

These are the testbench signals used to connect the vhdl description part.

The component is of course the just previously defined entity:

component binarytoBCD is

port(

sw : in std\_logic\_vector (5 downto 0);

HEX1 : out std\_logic\_vector(6 downto 0);

HEX0: out std\_logic\_vector(6 downto 0)

);

end component;

The operations for the simulation were done using a process:

--Begin assignments part

begin

binarytoBCD\_under\_test: binarytoBCD port map( sw\_tb , Hex1\_tb , HEX0\_tb );

--Process that generates the assignments necessary for the testing

process

variable sw\_v : unsigned(5 downto 0) := "000000";

begin

sw\_tb <= std\_logic\_vector(sw\_v);

while (sw\_v /= "111111") loop

sw\_v := sw\_v + "000001";

sw\_tb <= std\_logic\_vector(sw\_v);

wait for 200ns;

end loop;

wait;

end process;

end behaviour\_binarytoBCD\_tb;

It is a group of sequential assignments contained in a loop that is supposed to give all the possible inputs to our component from “000000”( or 0 (as the output) ) to “111111” ( or 63 (as the output)). So according to the 64 input waves the output wave is generated.

At the end the result is as follows:

While switching the buttons in the Altera FPGA the seven segment display represents the decimal number ( from 00 to 63 ) . Note there are used two displays HEX1 and HEX0 which are physical pins in the FPGA , since there are needed at least two digits to show the numbers from 10 to 63.