Anish Ghosh

Lab Report for Lab 3:7 Segment Display

Digital Design

Instructor: Dr.Fanging Hu

2021

Abstract

The objective of this lab was to design a circuit in VHDL to display some hexadecimal values on the board's the 7-segment decoder.

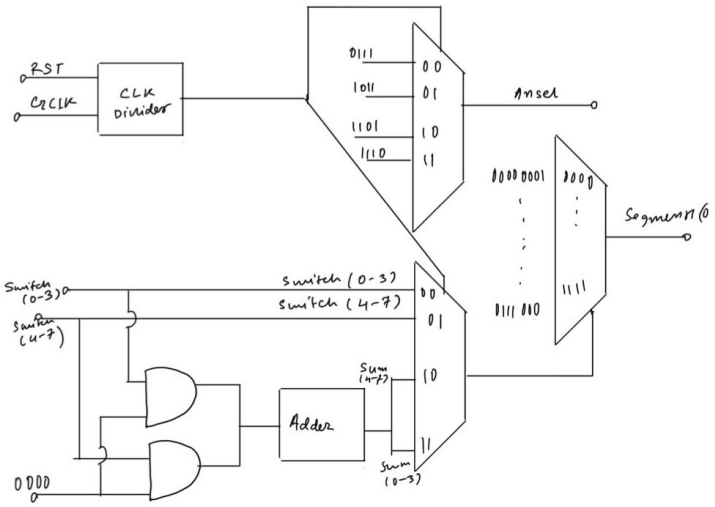
The interface of this module contains the following ports:

The gclk port is the global clock input (50MHz). The rst is the global reset signal. The $an_sel(3 downto 0)$ is the output signal which selects which seven segment display anode line to be active: $an_sel(3)$ selects AN3, $an_sel(2)$ selects AN2, etc. The switches(7 downto 0) is the input port for the 8 slide switches. The segments(6 downto 0) is the output port for the 7 segment control signals.

1 Aim

The 7-segment display has 4 digits. The 2 leftmost digits are to be used to display, in hexadecimal, the binary values of the left-most 4 switches and rightmost 4- switches. I can easily say that the maximum value the switches can reach is 1111 which in decimal is 15 or in hexadecimal F. Therefore, I can say that the value of the digit is never going to be over 1 hexadecimal digit. The sum, on the other way as also seen from the maximum values of the two binary strings I am interested it might have 1 or 2 digits depending on the values. As soon as the sum goes over 15, it must be represented in two digits space and that is the reason why there are needed two digits spaces for the sum.

2 Circuit Diagram



3 VHDL code

```
    library ieee;
use ieee.STD_LOGIC_1164.all;
use ieee.numeric_std.all;

entity seg_ctrl is
  port(
    gclk     : in  STD_LOGIC;                     -- Global clock input, 50MHz
    rst      : in  STD_LOGIC;                     -- Global reset input
    switches : in STD_LOGIC_VECTOR(7 downto 0);  -- Input switches
    segments : out STD_LOGIC_VECTOR(6 downto 0); -- Segment control
    an_sel   : out STD_LOGIC_VECTOR(3 downto 0)  -- Anode control
    );
end seg_ctrl;

architecture rtl of seg_ctrl is

  -- A 15 bit counter
  signal count : unsigned(14 downto 0);
  signal position : unsigned(1 downto 0);
  -- Storing variables
  signal sumlr : unsigned (7 downto 0);
  signal digit1 : unsigned (3 downto 0);

 begin
  -- Clock divider
  process (gclk, rst)
  begin
    if gclk='1' and gclk'event then
      if rst='1' then
         count <= (others=>'0');     --   asynchronous reset
      else
         count <= count+1;
      end if;
end if;
  end process;

  -- Anode decode circuitry
  position <= count(14 downto 13);

  with position select
  an_sel <= --anodes are low asserted
    "0111" when "00",
    "1011" when "01",
    "1101" when "10",
    "1110" when others;

with position select
digit1 <=
```

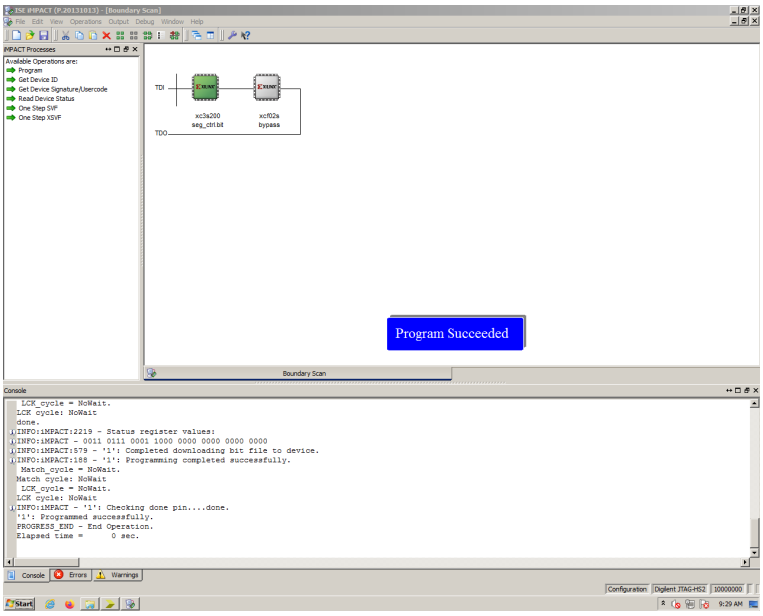
```
sumlr(7 downto 4) when "00",
sumlr(3 downto 0) when "01",
unsigned(switches(7 downto 4)) when "10",
unsigned(switches(3 downto 0)) when others;

with digit1 select --select segments to be lit up with each value (hard coded)

  segments <= "0000001" when "0000",
  "1001111" when "0001",
  "0010010" when "0010",
  "0000110" when "0011",
  "1001100" when "0100",
  "0100100" when "0101",
  "0100000" when "0110",
  "0001111" when "0111",
  "0000000" when "1000",
  "0000100" when "1001",
  "0001000" when "1010",
  "1100000" when "1011",
  "0110001" when "1100",
  "1000010" when "1101",
  "0110000" when "1110",
  "0111000" when others;


  process(switches(3 downto 0), switches(7 downto 4)) -- sum process
  begin
  sumlr <= ("0000" &

  unsigned(switches(3 downto 0)))+("0000" & unsigned(switches(7 downto 4)));

  end process;

end rtl;
```

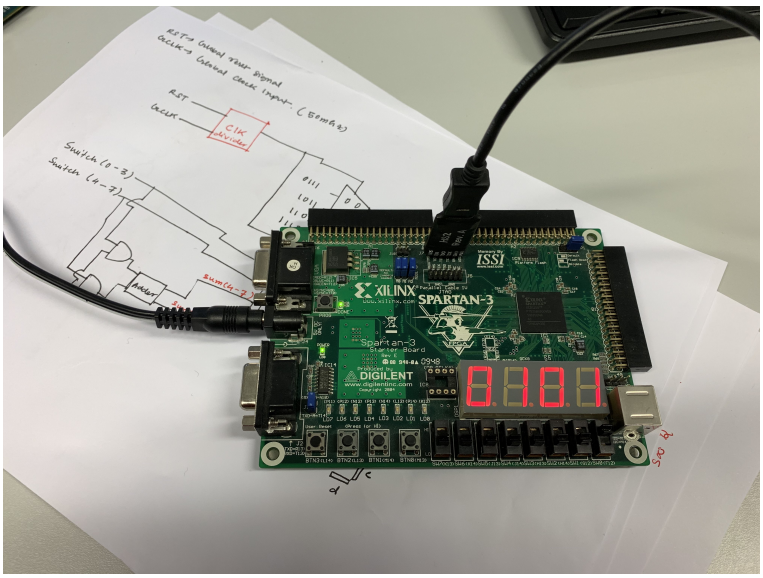
4 UCF code

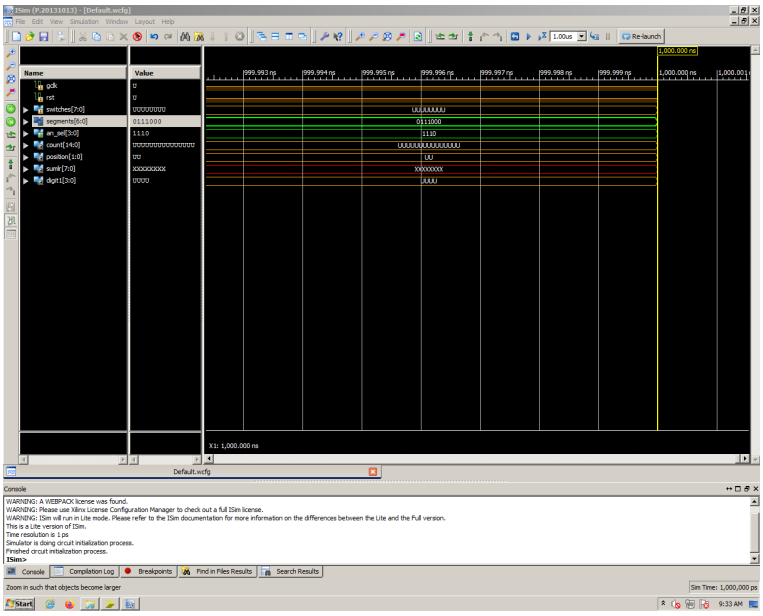
```
NET "switches<0>" LOC="F12" ;
NET "switches<1>" LOC="G12" ;
NET "switches<2>" LOC="H14" ;
NET "switches<3>" LOC="H13" ;

NET "switches<4>" LOC="J14" ;
NET "switches<5>" LOC="J13" ;
NET "switches<6>" LOC="K14" ;
NET "switches<7>" LOC="K13" ;

NET "gclk" LOC="T9";
```

```
NET "rst" LOC="M13";


NET "an_sel<0>" LOC = "D14";
NET "an_sel<1>" LOC = "G14";
NET "an_sel<2>" LOC = "F14";
NET "an_sel<3>" LOC = "E13";


NET "segments<0>" LOC = "N16";
NET "segments<1>" LOC = "F13";
NET "segments<2>" LOC = "R16";
NET "segments<3>" LOC = "P15";
NET "segments<4>" LOC = "N15";
NET "segments<5>" LOC = "G13";
NET "segments<6>" LOC = "E14";
```

5 Images of the results







6 Conclusion

In this lab I learned about using 7 segment display through Spartan 3 and Xilinx simulation. I was introduced with the concept of block diagrams and how to make them for given assignment. The VHDL and UCF concepts learned

in earlier labs were tested here, and using that I fulfilled the task.

7 References

1. $http://fpga-fhu.user.jacobs-university.de/?page_id=402$
2. $https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471786411.ch2$
3. $https://www.electronics-tutorials.ws/logic/logic_7.html$
4. $https://web.itu.edu.tr/\ ateserd/VHDL.pdf$