DSD ASSIGNMENT REPORT

Team Name : 221B
Team Members :

Rokaya Mohamed Elziady 43-5612 T-21
Nadine Mohamed Hassan 43-2145 T-21
Mariam Adel Enany 43-4045 T-21
Monica George 43-0818 T-21
Hoda Amr Elhemaly 43-5612 T-21

Introduction

This assignment uses FPGA to implement a counter that counts the seconds from 0 to 59 and has the option to reset and pause. The output should be displayed on two seven segment display.

The Digital timer is divided into 4 main entities:

The first entity is the clk_generator that takes the 50MHz as an input clock and returns the output of 1Hz

inputs: clki:takes 50MHz clock from fpga

outputs: clko:returns 1Hz clock

Signals:

1) count: counts periods at rising edge and when half 50 million passes temp should be inverted

2) temp: starts with 0 and gets inverted after 25 million periods

Output: clko takes the value of the temp

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY clk_generator IS
port (clki : in std_logic;
       clko : out std_logic
     );
 END clk_generator ;


 architecture arch1 of clk_generator is

signal count : integer :=1;
signal temp : std_logic :='0';
BEGIN

 --clk generation.For 50 MHz clock this generates 1 Hz clock.
PROCESS(clki) begin

if(clki'event and clki='1')
 then  count <= count+1;
 clko <=temp;

 if(count = 25000000)then temp <= not temp;
      count <= 1;
                clko <= temp ;
end if;

end if;
--clko <= temp ;
end process;
end arch1 ;
```

fig(1) Code for clk_generator

The second entity is the sixty_counter it counts the seconds from 0 to 59 using the input clock and can pause or reset
Inputs: clk,reset and pause
Outputs:f1 and f2
When reset is zero signal temp1 and temp2 are reset to 0, and when pause is zero nothing happens, but if pause is 1 and reset is 0 temp1 and temp2 are reset to 0, and pause is 1 , reset is 1 and it is a rising clock edge if temp1=9 it is set to 0 and temp 2 increases ,but if temp2=5 it is also set to 0.
Finally, f1 and f2 takes the value of temp1 and temp2 respectively.

```
USE ieee.std_logic_1164.all;
USE ieee.std_logic_ARITH.all;
USE ieee.std_logic_UNSIGNED.all;
use ieee.numeric_std.all;

ENTITY  sixty_counter IS
PORT ( clk : IN std_logic;
      reset,pause : IN std_logic := '0';
f1,f2 : OUT std_logic_vector(3 downto 0 ));
 END  sixty_counter ;

 ARCHITECTURE arch2 OF sixty_counter Is
   signal temp1:iNTEGER:=0;                    --UNSIGNED(3 downto 0):= "0000";
        signal temp2:INTEGER:=0;                        --(3 downto 0):= "0000";
begin
  process(clk, reset)
   begin
        if Reset='0' then
         temp1 <= 0;
                      temp2 <= 0;
       elsif pause='1'  then
      if Reset='0' then
         temp1 <= 0;
                      temp2 <= 0;
      elsif(clk'event and clk='1') then    --elsif(rising_edge(clock))
            if temp1=9 then
               temp1<=0;
                                        if(temp2=5) then
                                        temp2 <= 0;

                                        else
                                        temp2 <= temp2 + 1;
                                    end if;
            else
               temp1 <= temp1 + 1;
                                    end if;
         end if;
      end if;
   end process;
   f1 <= std_logic_vector(to_unsigned(temp1,4));
        f2 <= std_logic_vector(to_unsigned(temp2,4));
end arch2;
```

fig(2) Code for sixty_counter

The third entity is the seven_ segment _decoder which takes as an input the 2 std_logic_vector of length 4 and display them into a 2 std_logic_vector of length 7 as a seven segment display that will be displayed as an output on the fpga.
Inputs:in1 and in2 2 std_logic_vector of length 4
Output:Hout1 and Hout2 2 std_logic_vector of length 7
According the value of in1 pins of Hout1 are set high or low and according to in2 pins of Hout2 are set high or low.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_ARITH.all;
USE ieee.std_logic_UNSIGNED.all;

ENTITY  seven_seg_dec IS
PORT(in1, in2: IN   std_logic_vector(3 downto 0  ) ;
Hout1 , Hout2 : OUT std_logic_vector(6 downto 0) );

 END seven_seg_dec ;

ARCHITECTURE arch3 OF seven_seg_dec IS
 begin
 process(in1,in2)
 begin
  case(in1) is
   when "0000" =>  Hout1 <= "1000000"; --0--
   when "0001" =>  Hout1 <= "1111001"; --1--
   when "0010" =>  Hout1 <= "0100100"; --2--
   when "0011" =>  Hout1 <= "0110000"; --3--
   when "0100" =>  Hout1 <= "0011001"; --4--
   when "0101" =>  Hout1 <= "0010010"; --5--
   when "0110" =>  Hout1 <= "0000010"; --6--
   when "0111" =>  Hout1 <= "1111000"; --7--
   when "1000" =>  Hout1 <= "0000000"; --8--
   when others =>  Hout1 <= "0010000"; --9--"1001"
   end case;
        case(in2) is
   when "0000" =>  Hout2 <= "1000000"; --0--
   when "0001" =>  Hout2 <= "1111001"; --1--
   when "0010" =>  Hout2 <= "0100100"; --2--
   when "0011" =>  Hout2 <= "0110000"; --3--
   when "0100" =>  Hout2 <= "0011001"; --4--
   when others  =>  Hout2 <= "0010010"; --5-- "0101"
   end case;

 end process;
 end arch3 ;
```

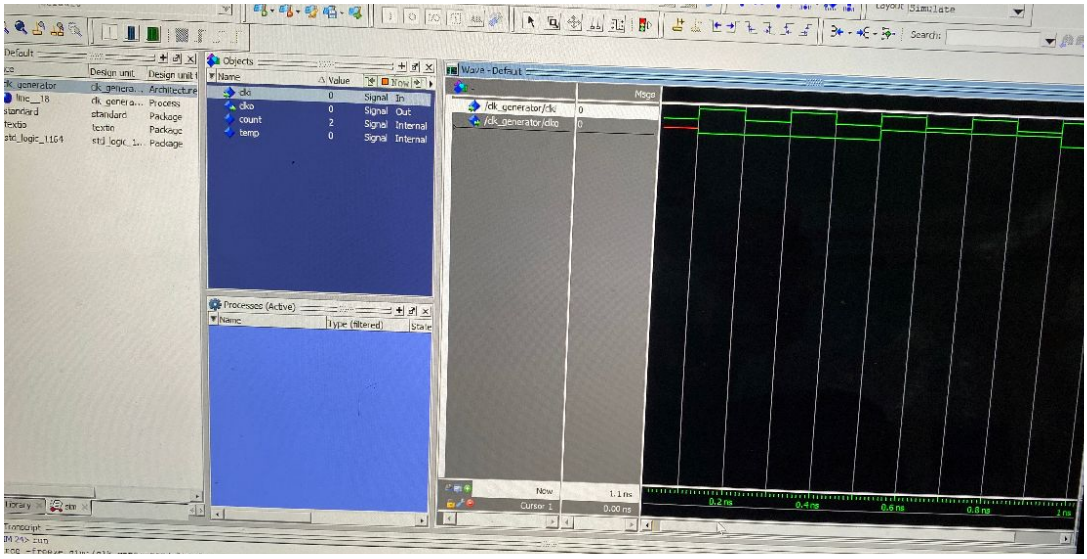
fig(3) code for seven segment decoder

The fourth entity is DSD_Ass2 takes as input the clock, reset and pause, and outputs the 2 7segment display vector and uses the 3 entities created above as components in its architecture and it contains signals to port map the outputs of entities in the hierarchy.
It has 3 components the clk_generator, the sixty_counter,seven segment decoder
We mapped the clk to the input of the clk_generator to produce clknew,and then we mapped the clknew,reset and pause as input to the sixty_counter to produce sig1 and sig2 ,and finally we mapped sig1 and sig2 as input to the seven segment decoder and mapped the clko to its output.

Simulation

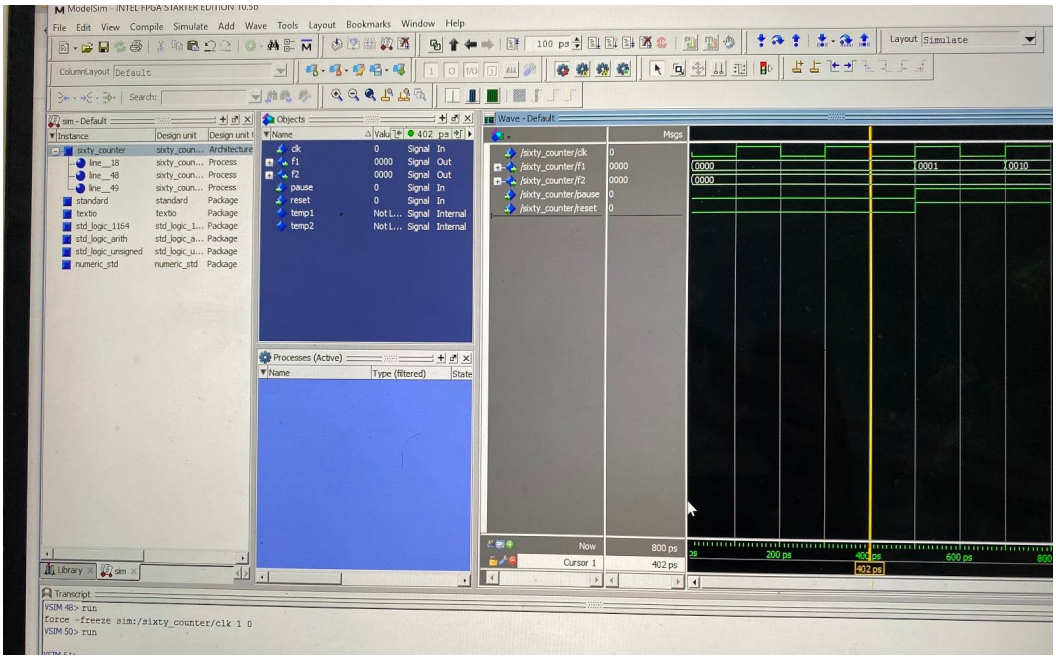
Clk_generator: we tested with count 2 to be able to see the change .

Result to this ,the frequency decreases to ¼ , making period increased by 4x. So the Clk_generator is working.



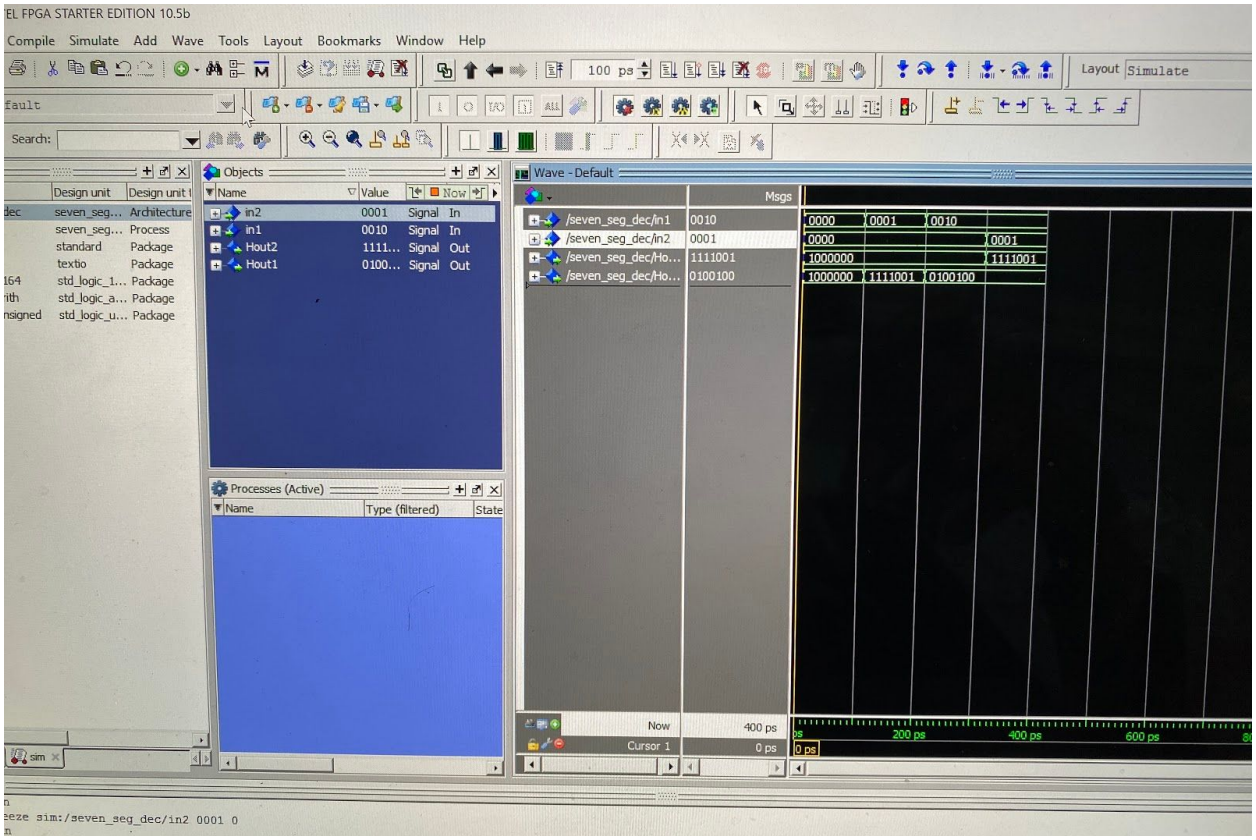
fig(4)

Sixty_counter: when it was paused nothing happened and every rising edge the count increases by 1 output is displayed as a 2 std_logic_vector of length 4 .



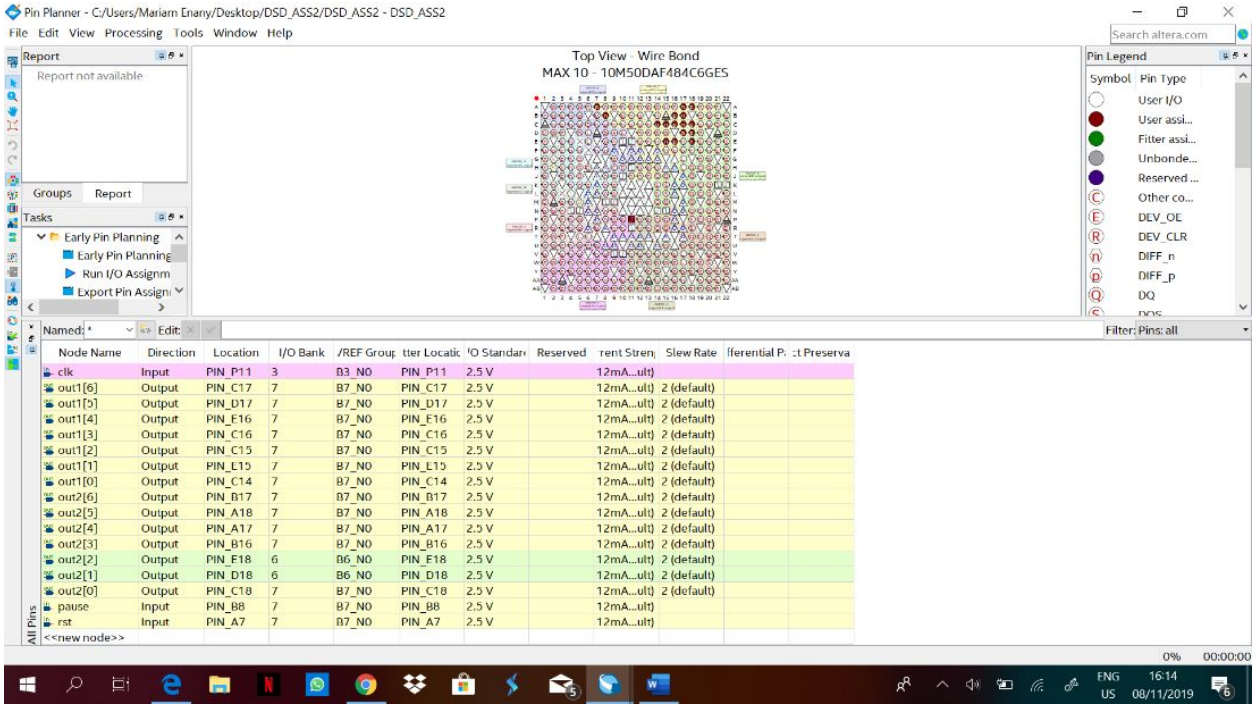
fig(5)

Seven_segment_decoder: Displays the low and high pins of the 2 seven segment decoders according to the two input std_logic_vectors of length 4 in1 and in2.



fig(6)

Pin Assignments



- Clk --> PIN_P11
- Pause --> PIN_B8
- Rst --> PIN_A7
- OUT1[6] --> PIN_C17
- OUT1[5] --> PIN_D17
- OUT1[4] --> PIN_E16
- OUT1[3] --> PIN_C16
- OUT1[2] --> PIN_C15
- OUT1[1] --> PIN_E15
- OUT1[0] --> PIN_C14
- OUT2[6] --> PIN_B17
- OUT2[5] --> PIN_A18
- OUT2[4] --> PIN_A17
- OUT2[3] --> PIN_B16
- OUT2[2] --> PIN_E18
- OUT2[1] --> PIN_D18
- OUT2[0] --> PIN_C18