Assignment - 10 Report

Tanish Gupta , Divyanshu Agarwal

29th May 2022

1 Introduction

The objective of this assignment was to transfer files between PC and FPGA board. We used several of components of previous assignments to accomplish the tasks of this assignment.

2 Strategy and Logic

All we had to do was to instantiate many of the previously designed components in the main design file and link them. No major difficulty was faced.
To make our design interesting, we had added two kinds of features for transmission. One is called tx_one and the other is called tx_all. The former one trasmits each character one by one, while the latter one trasmit all of the memory at once. The reset button on the other hand clears all the memory.

3 Code and Working

The main directory contains many files, since we tried our best to make the design as modular as possible. The top level entity is in design.vhd file, which has the VHDL code logic for the instantiating all the components and providing the glue logic, timing_circuit.vhd file, which acts as the controller and provides the signals for glue logic. The rest of the files - receiver.vhd, transmitter.vhd, BRAM.vhd, singleDisplay.vhd, multiDisplay.vhd, and myTypes.vhd are same as submitted in assignment 8 and 9. The directory also has the constraint file and the generated bitstream.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity design is
  Port (
    clk          : in std_logic;
    inbit        : in std_logic;
    reset        : in std_logic; --button
    tx_start     : in std_logic; --button
    tx_all       : in std_logic; --button
    LED          : out std_logic_vector(6 downto 0);
    anode        : out std_logic_vector(3 downto 0);
    serial_output: out std_logic
  );
end design;

architecture Behavioral of design is
```

```
signal tx_start_debounced       : std_logic :='0';
signal reset_debounced          : std_logic :='0';
signal done                     : std_logic;
signal serial_data              : std_logic_vector(7 downto 0);
signal inbyte                   : std_logic_vector(7 downto 0);
signal tx_empty                 : std_logic;

signal ld_tx                    : std_logic;
signal rx_full                  : std_logic;
signal rd_addr                  : std_logic_vector(15 downto 0);
signal wr_addr                  : std_logic_vector(15 downto 0);
signal write_en                 : std_logic;

signal byteToDisplay            : std_logic_vector(15 downto 0):= x"0000";
signal counter                  : integer :=0;
signal slow_clock               :std_logic:='0';

type state is (idle, processing,completed);

signal tx_state                 : state := idle ;
signal reset_state              : state :=idle ;
signal tx_start_final           : std_logic := '0';
signal tx_start_state           : state := idle;

begin

    process(clk) is
        begin
            if(rising_edge(clk)) then

                if(counter = 2500000) then
                    counter <= 0;
                    slow_clock <= not(slow_clock);
                else
                    counter <= counter + 1;
                end if;

            case(tx_start_state) is
                when idle =>
                    tx_start_final <= '0';
                    if(tx_start_debounced = '1') then
                        tx_start_state <= processing;
                    else
                        tx_start_state <= idle;
                    end if;

                when processing =>
                    tx_start_final <= '1';
                    tx_start_state <= completed;

                when others =>
                    tx_start_final <= '0';
                    if(tx_start_debounced = '0') then
                        tx_start_state <= idle;
                    else
                        tx_start_state <= completed;

                    end if;
                end case;

            case(tx_start_state) is
                when idle =>
                    tx_start_final <= '0';
                    if(tx_start_debounced = '1') then
                        tx_start_state <= processing;
                    else
                        tx_start_state <= idle;
                    end if;

                when processing =>
                    tx_start_final <= '1';
                    tx_start_state <= completed;

                when others =>
                    tx_start_final <= '0';
                    if(tx_start_debounced = '0') then
                        tx_start_state <= idle;
                    else
                        tx_start_state <= completed;

                    end if;
                end case;

            if(tx_start_debounced = '1') then
                byteToDisplay <= x"00" & inbyte;
            elsif (done = '1') then
                byteToDisplay <= x"00" & serial_data;
            elsif (reset = '1') then
                byteToDisplay <= x"0000";
            end if;
          end if;
    end process;

    process(slow_clock)
        begin
            if(rising_edge(slow_clock) and slow_clock'event) then
                if(tx_start = '1') then
                        case (tx_state) is
                        when idle =>
                            tx_start_debounced <= '1';
                            tx_state<=processing;

                        when processing =>
                            tx_start_debounced <='0';
                            tx_state<=completed;

                        when others =>
```

```
                        tx_start_debounced <='0';
                        tx_state<=completed;

               end case;

                    else
                    tx_start_debounced <= '0';
                    tx_state <= idle ;


                    end if;

         if(reset = '1') then
             case (reset_state) is
                 when idle =>
                     reset_debounced <= '1';
                     reset_state<=processing;

                 when processing =>
                     reset_debounced <='0' ;
                     reset_state<=completed;

                 when others =>
                     reset_debounced <='0' ;
                     reset_state<=completed;

             end case;
         else
             reset_debounced <= '0';
             reset_state<=idle;
         end if;

   end if;

end process;

Receiver:
     entity work.receiver(behavioral)
         port map (
             clk => clk,
             inbit => inbit,
             reset => reset_debounced,
             done => done,
             serial_data => serial_data,
             rx_full => rx_full
         );

Transmitter:
     entity work.transmitter(behavioral)
         port map (
             clk => clk,
             start => ld_tx,
             reset => reset_debounced,
             inbyte => inbyte,
                serial_output => serial_output,
                tx_empty => tx_empty
            );

    TimingCircuit:
        entity work.timing_circuit(behavioral)
            port map (
                clk,
                tx_start_final,
                rx_full,
                tx_empty,
                tx_all,
                ld_tx,
                rd_addr,
                wr_addr,
                write_en,
                done,
                reset_debounced
            );

    myBRAM:
        entity work.BRAM(behavioral)
            port map(
                clk,
                write_en,
                wr_addr,
                rd_addr,
                serial_data,
                inbyte
            );

    Multi_display:
        entity work.lightDisplay(structure)
            port map(
                byteToDisplay,
                clk,
                LED,
                anode
            );

end Behavioral;
```

4 Simulation and Synthesis

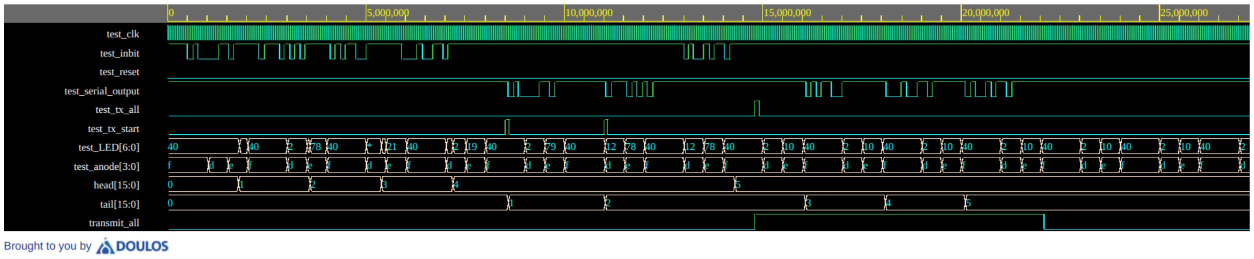

Figure 1: Generated EP Wave - 1

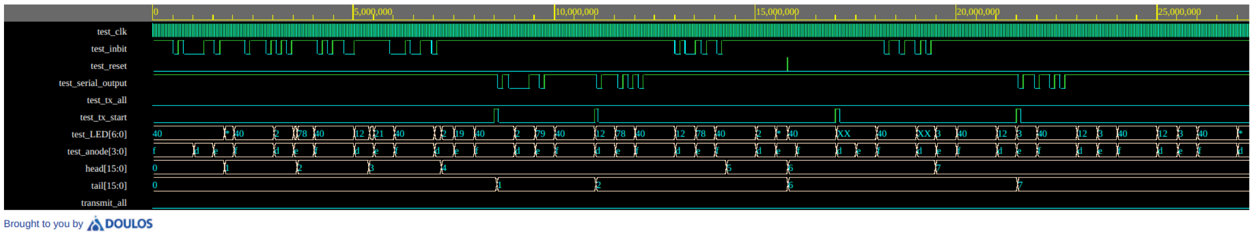

Figure 2: Generated EP Wave - 2

These EPWaves were carefully analysed to check if all outputs were correct!
The Vivado synthesis report is as follows:

```
Copyright 1986-2019 Xilinx, Inc. All Rights Reserved.
------------------------------------------------------------------------------------
| Tool Version : Vivado v.2019.1 (lin64) Build 2552052 Fri May 24 14:47:09 MDT 2019
| Date         : Sat May 28 20:22:04 2022
| Host         : divyanshu-HP-ENVY-x360-Convertible-13-bd0xxx
                 running 64-bit Ubuntu 20.04.2 LTS
| Command      : report_utilization -file design_utilization_synth.rpt
                 -pb design_utilization_synth.pb
| Design       : \design
| Device       : 7a35tcpg236-2L
| Design State : Synthesized
------------------------------------------------------------------------------------

Utilization Design Information

Table of Contents
-----------------
1. Slice Logic
1.1 Summary of Registers by Type
2. Memory
3. DSP
4. IO and GT Specific
5. Clocking
6. Specific Feature
```

```
7. Primitives
8. Black Boxes
9. Instantiated Netlists

1. Slice Logic
--------------

+----------------------------+------+-------+-----------+-------+
|          Site Type         | Used | Fixed | Available | Util% |
+----------------------------+------+-------+-----------+-------+
| Slice LUTs*                |  233 |     0 |     20800 |  1.12 |
|   LUT as Logic             |  233 |     0 |     20800 |  1.12 |
|   LUT as Memory            |    0 |     0 |      9600 |  0.00 |
| Slice Registers            |  249 |     0 |     41600 |  0.60 |
|   Register as Flip Flop    |  249 |     0 |     41600 |  0.60 |
|   Register as Latch        |    0 |     0 |     41600 |  0.00 |
| F7 Muxes                   |    0 |     0 |     16300 |  0.00 |
| F8 Muxes                   |    0 |     0 |      8150 |  0.00 |
+----------------------------+------+-------+-----------+-------+

1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 0     |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 249   |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+


2. Memory
---------

+-------------------+------+-------+-----------+-------+
|     Site Type     | Used | Fixed | Available | Util% |
+-------------------+------+-------+-----------+-------+
| Block RAM Tile    |   16 |     0 |        50 | 32.00 |
|   RAMB36/FIFO*    |   16 |     0 |        50 | 32.00 |
|     RAMB36E1 only |   16 |       |           |       |
|   RAMB18          |    0 |     0 |       100 |  0.00 |
+-------------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and
        therefore can accommodate only one FIFO36E1 or one FIFO18E1.
        However, if a FIFO18E1 occupies a Block RAM Tile,
        that tile can still accommodate a RAMB18E1
```

3. DSP

Site Type	Used	Fixed	Available	Util%
DSPs	0	0	90	0.00

4. IO and GT Specific

Site Type	Used	Fixed	Available	Util%
Bonded IOB	17	0	106	16.04
Bonded IPADs	0	0	10	0.00
Bonded OPADs	0	0	4	0.00
PHY_CONTROL	0	0	5	0.00
PHASER_REF	0	0	5	0.00
OUT_FIFO	0	0	20	0.00
IN_FIFO	0	0	20	0.00
IDELAYCTRL	0	0	5	0.00
IBUFDS	0	0	104	0.00
GTPE2_CHANNEL	0	0	2	0.00
PHASER_OUT/PHASER_OUT_PHY	0	0	20	0.00
PHASER_IN/PHASER_IN_PHY	0	0	20	0.00
IDELAYE2/IDELAYE2_FINEDELAY	0	0	250	0.00
IBUFDS_GTE2	0	0	2	0.00
ILOGIC	0	0	106	0.00
OLOGIC	0	0	106	0.00

5. Clocking

Site Type	Used	Fixed	Available	Util%
BUFGCTRL	1	0	32	3.13
BUFIO	0	0	20	0.00
MMCME2_ADV	0	0	5	0.00
PLLE2_ADV	0	0	5	0.00
BUFMRCE	0	0	10	0.00
BUFHCE	0	0	72	0.00
BUFR	0	0	20	0.00

6. Specific Feature

```
+--------------+------+-------+-----------+-------+
|   Site Type  | Used | Fixed | Available | Util% |
+--------------+------+-------+-----------+-------+
| BSCANE2      |    0 |     0 |         4 |  0.00 |
| CAPTUREE2    |    0 |     0 |         1 |  0.00 |
| DNA_PORT     |    0 |     0 |         1 |  0.00 |
| EFUSE_USR    |    0 |     0 |         1 |  0.00 |
| FRAME_ECCE2  |    0 |     0 |         1 |  0.00 |
| ICAPE2       |    0 |     0 |         2 |  0.00 |
| PCIE_2_1     |    0 |     0 |         1 |  0.00 |
| STARTUPE2    |    0 |     0 |         1 |  0.00 |
| XADC         |    0 |     0 |         1 |  0.00 |
+--------------+------+-------+-----------+-------+
```

```
7. Primitives
-------------

+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| FDRE     |  249 |        Flop & Latch |
| LUT4     |   79 |                 LUT |
| LUT2     |   73 |                 LUT |
| CARRY4   |   61 |          CarryLogic |
| LUT6     |   47 |                 LUT |
| LUT5     |   39 |                 LUT |
| LUT1     |   30 |                 LUT |
| RAMB36E1 |   16 |        Block Memory |
| LUT3     |   16 |                 LUT |
| OBUF     |   12 |                  IO |
| IBUF     |    5 |                  IO |
| BUFG     |    1 |               Clock |
+----------+------+---------------------+
```

```
8. Black Boxes
--------------

+----------+------+
| Ref Name | Used |
+----------+------+
```

```
9. Instantiated Netlists
------------------------

+----------+------+
| Ref Name | Used |
+----------+------+
```