Assignment - 3 Report

Tanish Gupta , Divyanshu Agarwal

8th May 2022

1 Introduction

The objective of this assignment is to take 4-digit decimal/hexadecimal number from slide switches and display it on seven segment displays of BASYS3 FPGA board.

2 Strategy and Logic

First, we did not know about the on board clock of BASYS3 board, and so we were using "for" loops, but that did not help us. This was because, due to delta delay in process assignments, we could only get the last iteration of the for loop, which was of no use.
Then we studied about on-board clock, and set its time period to 10 ns.
Now, We already knew that we need to display each digit for 4 ms, so that within 16 ms, we can display all 4 digits and they appear at same time to human eye.
So for 4 ms, we needed only one out of $\frac{4ms}{10ns} = 4 \times 10^5$ clock events. We named a signal counter_new and assigned it the value $\frac{counter}{400000}$ mod 4 (counter increments on each rising edge of clock) (We did modulo 4, to determine which digit to display).

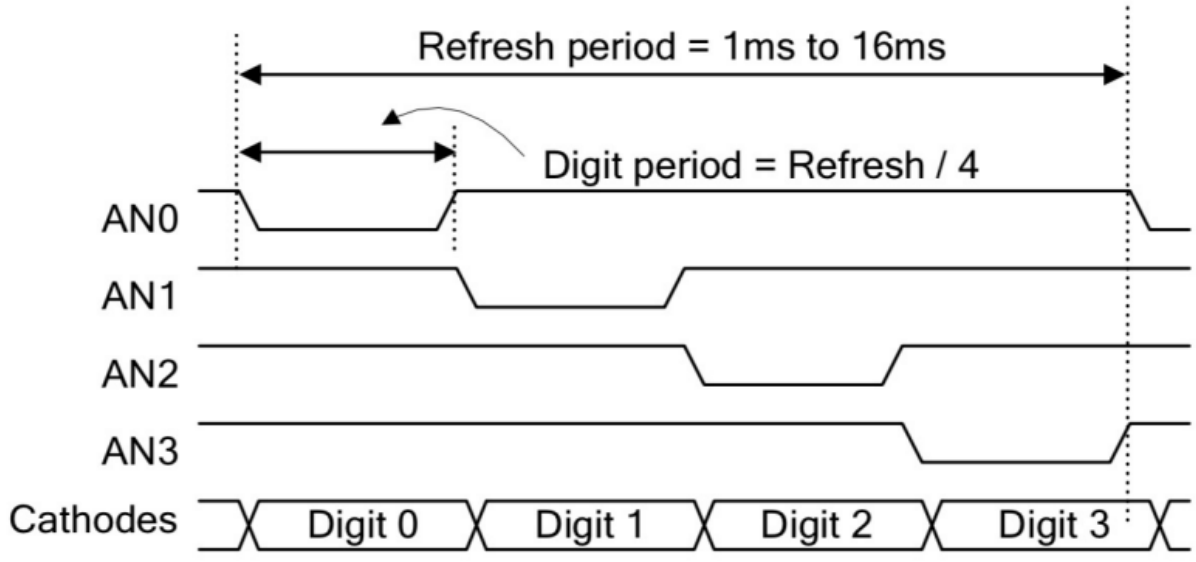

Figure 1: Refresh Timings

3 Code and Working

The main directory contains the design.vhd file, which has the VHDL code logic and the singleDisplay.vhd, which is the same file as submitted in assignment 2. The directory also has the constraint file and the generated bitstream.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_Std.all;

entity lightDisplay is
        Port (
            Number : in std_logic_vector(15 downto 0);
            Clk : in std_logic;
            LED : out std_logic_vector(6 downto 0);
            anode : out STD_LOGIC_Vector(3 downto 0));
end lightDisplay;

architecture structure of lightDisplay is
    component singleDisplay
        Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C : in STD_LOGIC;
           D : in STD_LOGIC;

           light_num : in integer;
           LED : out std_logic_vector(6 downto 0);
           anode : out STD_LOGIC_Vector(3 downto 0));
end component singleDisplay;

    signal light_num    : integer;
    signal digit_A      : std_logic;
    signal digit_B      : std_logic;
    signal digit_C      : std_logic;
    signal digit_D      : std_logic;
    signal counter      : integer :=0;
    signal counter_new  : integer := 0 ;

    begin
        process(clk)
                begin
                if(Clk'event  and rising_Edge(clk)) then
                    counter<= ((counter+1)) ;
                    counter_new<= ((counter) / 40000 )mod 4;

                    case (counter_new) is
                        when 0 =>
                            digit_A<=Number(15);
                            digit_B<=Number(14);
                            digit_C<=Number(13);
                            digit_D<=Number(12);
                            light_num<=(0) ;

                        when 1 =>
                            digit_A<=Number(11);
```

```
                    digit_B<=Number(10);
                    digit_C<=Number(9);
                    digit_D<=Number(8);
                    light_num<=(1) ;

                when 2 =>
                    digit_A<=Number(7);
                    digit_B<=Number(6);
                    digit_C<=Number(5);
                    digit_D<=Number(4);
                    light_num<=(2) ;

                when others =>
                    digit_A<=Number(3);
                    digit_B<=Number(2);
                    digit_C<=Number(1);
                    digit_D<=Number(0);
                    light_num<=(3) ;

                end case;

            end if;

        end process;


    Single_display:
        entity work.singleDisplay(Design_arch) port map (
            digit_A , digit_B, digit_C , digit_D ,light_num , LED ,anode
            );


end structure;
```

4 Simulation and Synthesis

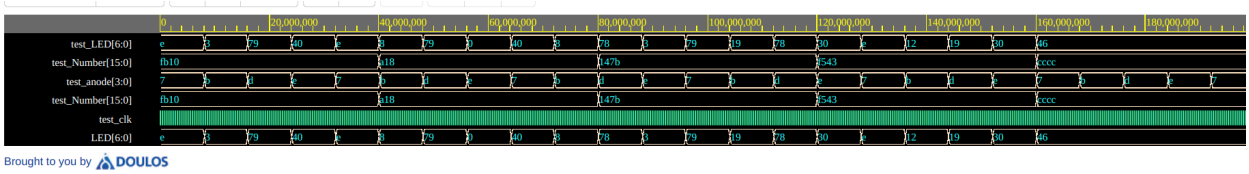

Figure 2: Generated EP Wave

This EPWave was 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         : Sun May  8 21:36:37 2022
| Host         : divyanshu-HP-ENVY-x360-Convertible-13-bd0xxx
                 running 64-bit Ubuntu 20.04.2 LTS
| Command      : report_utilization -file lightDisplay_utilization_synth.rpt
                 -pb lightDisplay_utilization_synth.pb
| Design       : lightDisplay
| Device       : 7a35tcpg236-2
| 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*                |   50 |     0 |     20800 |  0.24 |
|   LUT as Logic             |   50 |     0 |     20800 |  0.24 |
|   LUT as Memory            |    0 |     0 |      9600 |  0.00 |
| Slice Registers            |   40 |     0 |     41600 |  0.10 |
```

Register as Flip Flop	40	0	41600	0.10
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	-
40	Yes	Reset	-

2. Memory

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	0	0	50	0.00
RAMB36/FIFO*	0	0	50	0.00
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	28	0	106	26.42
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	40	Flop & Latch
LUT2	36	LUT
LUT1	22	LUT
IBUF	17	IO
CARRY4	17	CarryLogic
OBUF	11	IO
LUT4	7	LUT
LUT6	5	LUT
LUT3	1	LUT
BUFG	1	Clock

8. Black Boxes

Ref Name	Used

9. Instantiated Netlists

Ref Name	Used