EXPERIMENT4 : BCD Number subtratcion.

Devesh Soni
Roll no. :21D070025

August 19, 2022

1 Overview of the Experiment

- The main aim of this experiment is to get familiarized with XEN10 board (10M25SAE1448CG) .
- Through this experiment we designed the 4bit Binary coded decimal number subtractor using the previously designed 4 bit ADDER_SUBTRACTOR and some usual logic gates.
- In this report firstly I will explain:

1. Pen and Paper diagram of logic gate design of the function.
2. VHDL code , changes in DUT and testbench files and finally NETLIST view and RTL simulation of the implemented function.
3. PIN mapping , SVF file creation , and finally dumping this SVF file onto the board ,after successfully testing the working of board.

2 Experiment setup / approach to the assignment

2.1 BCD number subtractor using 10's compliment

- ***follow the given to subtract two numbers using 10's compliment method***

1. When subtracting two BCD numbers(A-B), find the 10's complement of the negative decimal number(B).
2. Add A+10's complement of B.
3. If carry is generated, the obtained result is positive. Discard the carry to get the result.
4. If carry is not generated, the obtained result is negative, find the 10's complement to get the final result.

9's complement of 48 = 99 − 48 = 51
10's complement of 48 = 9's complement + 1 = 51 + 1 = 52

	1	1	1						
BCD for 35	0	0	1	1		0	1	0	1
BCD for 10's complement of 48	0	1	0	1		0	0	1	0	+
Result, No carry	1	0	0	0		0	1	1	1
Decimal Equivalent of result	8					7			

9's complement of 87 = 99 − 87 = 12
10's complement of 87 = 9's complement + 1 = 12 + 1 = 13

Figure 1: 10's Compliment example

- Created design on paper as shown, marked signals and numbered various gates, optimized the boolean expression using K-map.

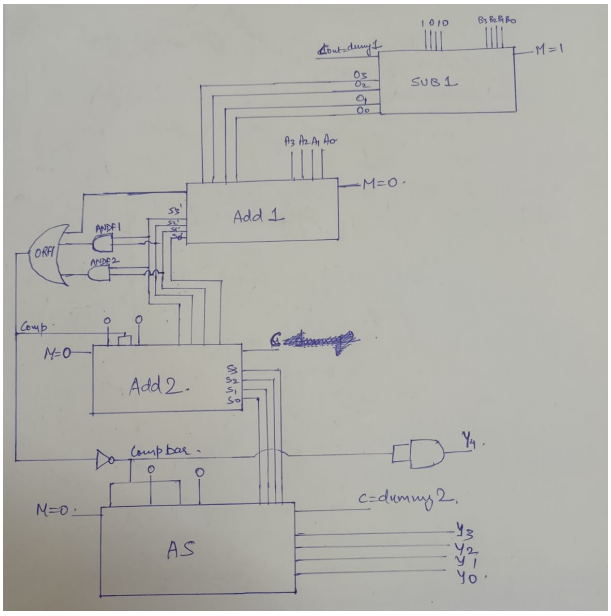

Figure 2:

2.2 QUADSUB.vhdl

1. Import required libraries
2. Created two packages FULLADD and ADD_SUB.
3. Construct entity QUADSUB.
4. Construct architecture, define signals.
5. Component instantiation.
6. final code is as follows:

```
library ieee;
use ieee.std_logic_1164.all;
library work;
use work.Gates.all;
use work .FULLADD.all;
use work.ADD_SUB.all;

entity QUADSUB  is
  port (A3, A2,A1,A0, B3,B2,B1,B0: in std_logic; Y4, Y3, Y2, Y1, Y0: out std_logic);
end entity QUADSUB;


architecture FFFF of QUADSUB is
  signal o0,o1,o2,o3,andd1 , andd2 , s3b, s2b ,s1b,s0b, C , Cb,compbar, s3,s2,s1,s0, dummy1,dummy2,co
begin
  -- component instances
  SUB1: ADDER_SUBTRACTOR port map ('1','0','1','0', B3,B2,B1,B0,'1',dummy1,o3,o2,o1,o0);
  ADD1: ADDER_SUBTRACTOR port map (o3,o2,o1,o0,A3,A2,A1,A0,'0',cb,s3b,s2b,s1b,s0b);
  ANDF1: AND_2 port map (s3b,s2b,andd1);
  ANDF2: AND_2 port map (s3b,s1b,andd2);
  ORF1: OR_2 port map (andd1,andd2,orff1);
  ORF2: OR_2 port map (orff1,cb,comp);
  ADD2: ADDER_SUBTRACTOR port map ('0',comp,comp,'0',s3b,s2b,s1b,s0b,'0', c,s3,s2,s1,s0);
  INV: INVERTER port map(comp,compbar);
  AS: ADDER_SUBTRACTOR port map ( compbar,'0',compbar, '0',s3,s2,s1,s0, compbar, dummy2,Y3,Y2,Y1,Y0);
  ANDZ: AND_2 port map (compbar,compbar,Y4);
```

```
end FFFF;
```

- Gates.vhdl(no change), Testbench.vhdl- change the number of outputs and inputs
- In DUT file(important to make it top level entity)change component's name and instance name to QUADSUB, changed input vector to 7 (with A3 as MSB), and output vector to 4 .

2.3 Working with XEN10 board

- Go to assignments —-> pin planning—-> and input the correct pin no.

Switch	FPGA Pin no.	LED	FPGA Pin no.
SW 8	47	LED 8	60
SW 7	46	LED 7	59
SW 6	45	LED 6	58
SW 5	44	LED 5	57
SW 4	43	LED 4	56
SW 3	41	LED 3	54
SW 2	39	LED 2	52
SW 1	38	LED 1	50

Pin Mapping for On-board switches and leds

Figure 3: Pin Mapping for switches and leds

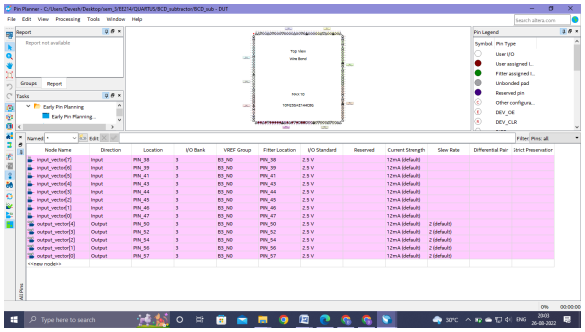

Figure 4: PIN MAPPING FOR FUNCTION

- after pin planning,
  in the same window go to processing—-> start I/O assignment analysis—-> and compile again.
- once compilation is complete create a **.svf** file
  in tools—->go to programmer—-> new file of format SVF and press ok .
  A new .svf file will be created in the same folder as of the project.
- once the svf file is created,connect the board to the laptop through USB cabel provided, open the **jtag.exe** application and write the required code.
- while copying the path of .svf file make sure that thesvf must not be stored in the directory which contains spaces in its name .

3 Observations

3.1 NETLIST and RTL viewer

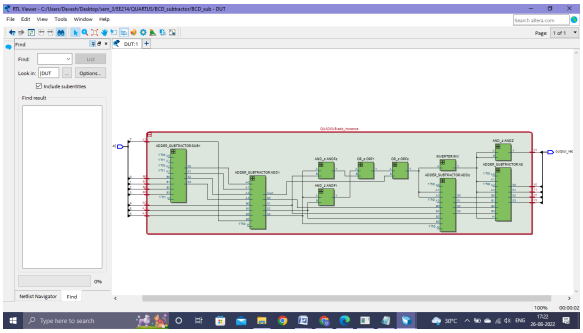

Figure 5: NETLIST VIEWER

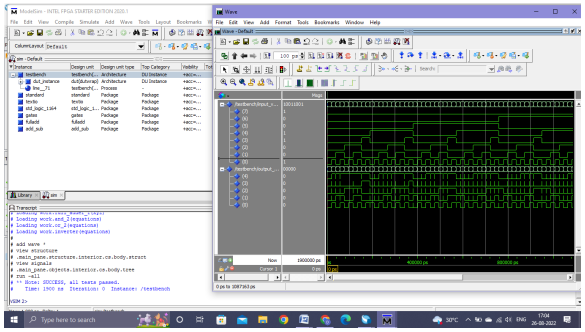

Figure 6: RTL VIEWER