

Experiment 1

Combinational Circuits - 1

Part 3

Mayur Ware, 19D070070
EE-214, WEL, IIT Bombay

February 15, 2021

Overview of the experiment

- The purpose of the experiment is to design a special combinational circuit which gets a two digit number AB as an input and then perform following actions :

Digit A	Digit B	Output
Valid BCD	Valid BCD	AB + 29
Valid BCD	Not BCD	A 1111
Not BCD	Valid BCD	1111 B
Not BCD	Not BCD	1111 1111

- I used Part 1 logic to convert the numbers in their respective BCD forms or into 1111 accordingly. Then I used BCD_Adder, Eight_Bit_Multiply and Eight_Bit_Adder to implement my design.
- The report contains a handmade diagram which describes my approach, some important extracts of VHDL code, the output waveforms and Krypton Board Observations.

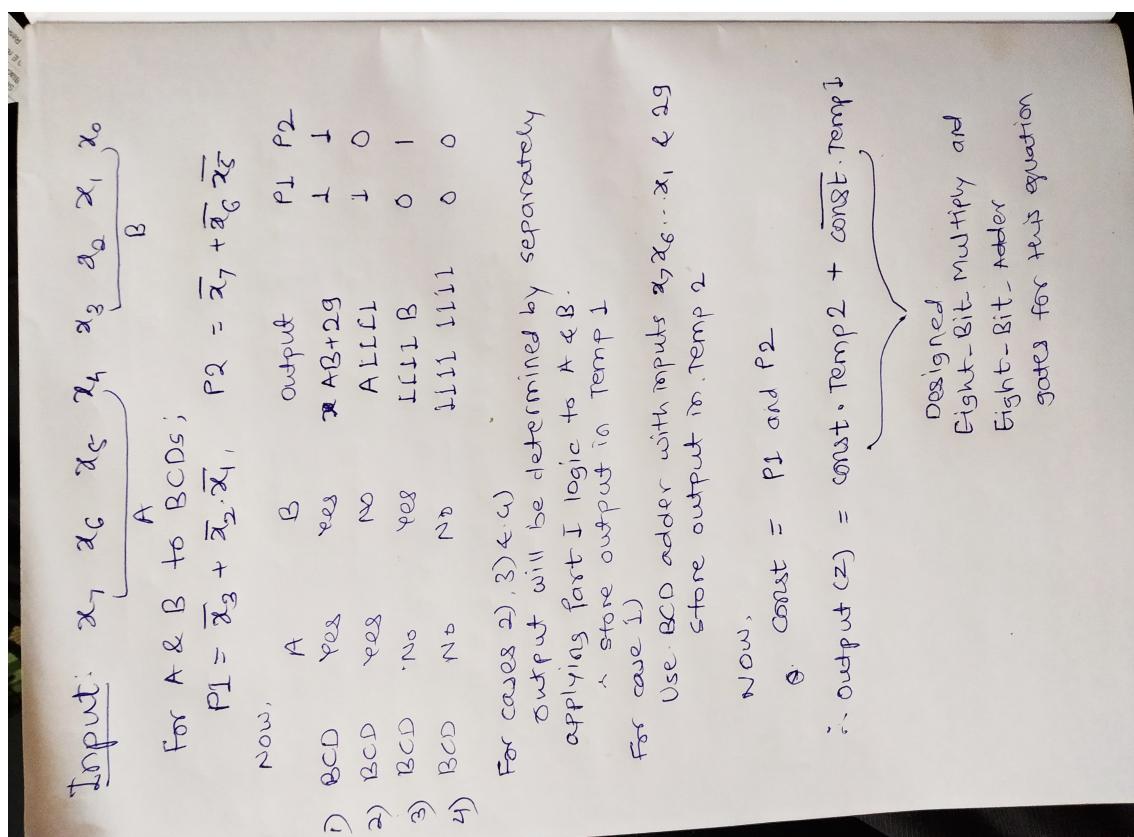
Approach to the experiment

Using Part 1 equations and logic, I converted No.s in the 'AB', 'A 1111', '1111 B', '1111 1111' forms accordingly and stored results into vector T1. Then, I used BCD_Adder and gave inputs as 29 and T1 to it and stored its value in T2.

Now, the only condition when BCD Converting is done is 'P1'=1 and 'P2'=1 I stored ('P1' and 'P2') into a variable 'Const'. So, for 'Const'=0, we would directly get the output and for 'Const'=1, we will add 29 into it. Using Eight_Bit_Multiply, I converted 'T1' into 'Temp1' and 'T2' into 'Temp2' accordingly

∴ Our Output will Const*Temp1 + Const*Temp2

Then, I used the Eight_Bit_Adder to implement this equation.



Design document and VHDL code if relevant

Main Entity :

```

entity Part3 is
port (X : in std_logic_vector(7 downto 0);
Z : out std_logic_vector(7 downto 0));
end entity Part3;
P1 <= (not X(3) or ((not X(2)) and (not X(1))));           -Eqn for checking if integer B is BCD
P2 <= (not X(7) or ((not X(6)) and (not X(5))));           -Eqn for checking if integer B is BCD

```

BCD_Adder :

```

component BCD_Adder is
port(W : in std_logic_vector(7 downto 0);
X : in std_logic_vector(7 downto 0);
Y : out std_logic_vector(7 downto 0);
Z : out std_logic);
end component BCD_Adder;

```

Eight_Bit_Multiply :

```

component Eight_Bit_Multiply is
port(inp1 : in std_logic_vector(7 downto 0);
inp2 : in std_logic_vector(7 downto 0);
outp : out std_logic_vector(7 downto 0));
end component Eight_Bit_Multiply;

```

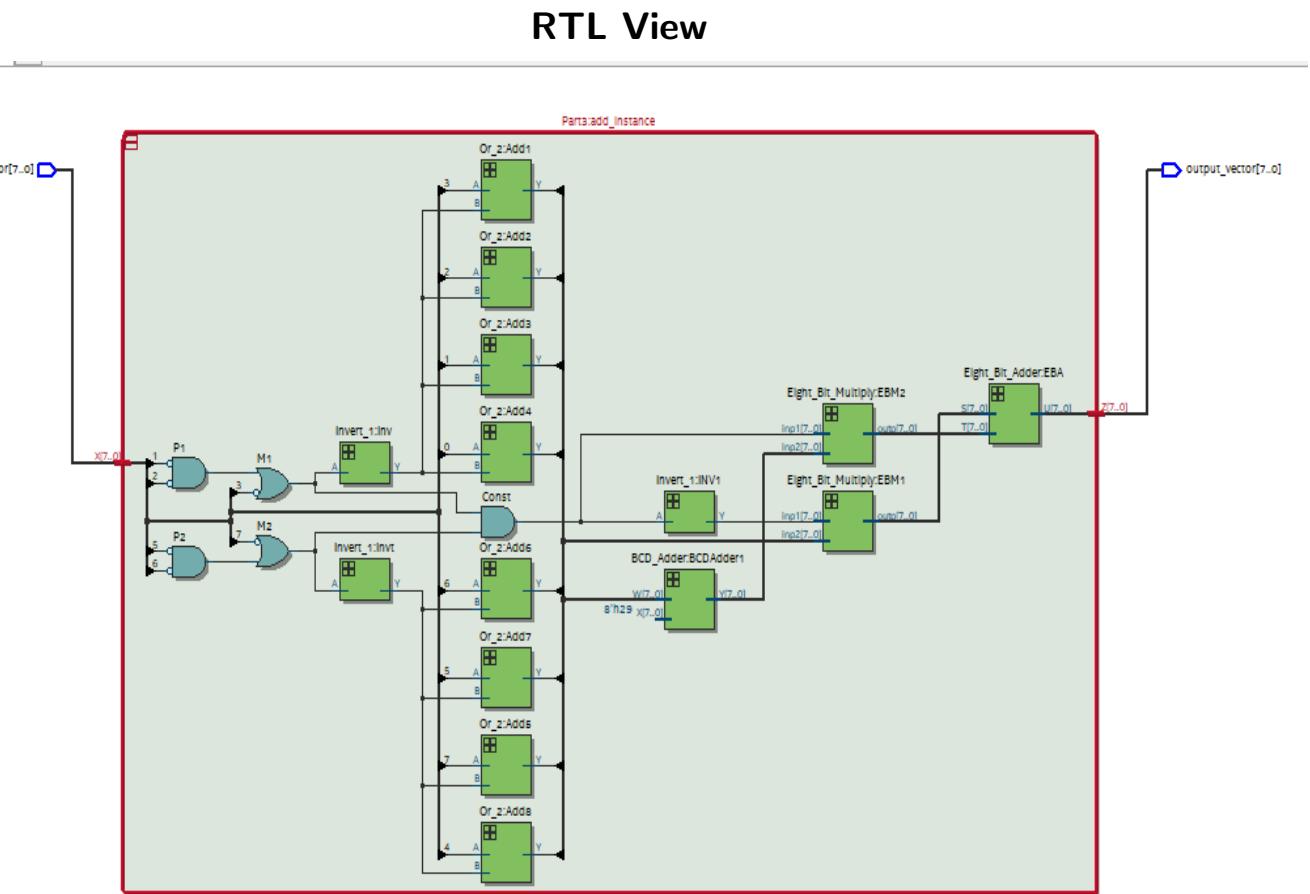
Eight_Bit_Adder :

```

component Eight_Bit_Adder is
port(S: in std_logic_vector(7 downto 0);
T: in std_logic_vector(7 downto 0);
U: out std_logic_vector(7 downto 0));
end component Eight_Bit_Adder;

```

-Eqn for checking if integer B is BCD
-Eqn for checking if integer B is BCD

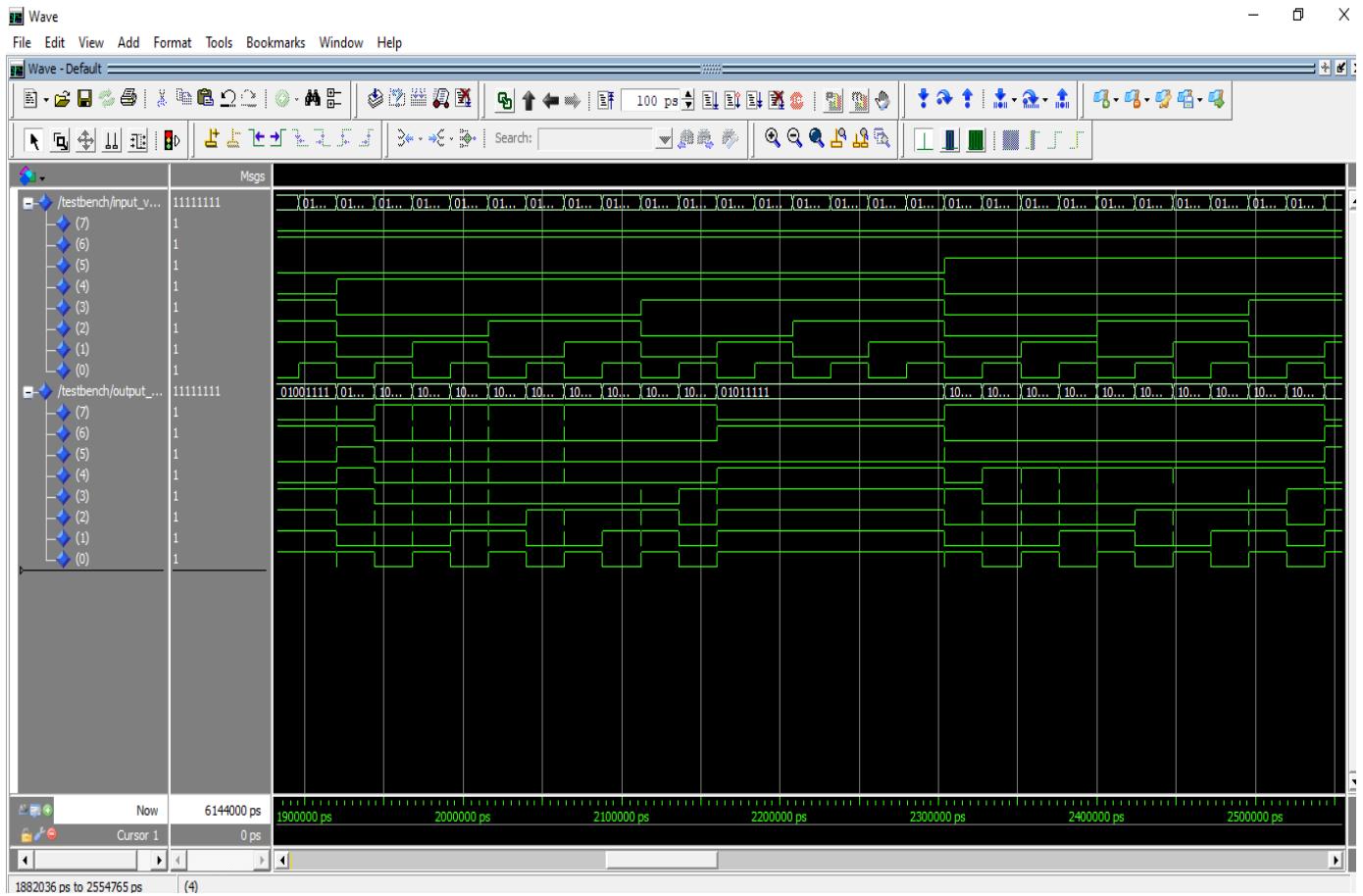


DUT Input/Output Format

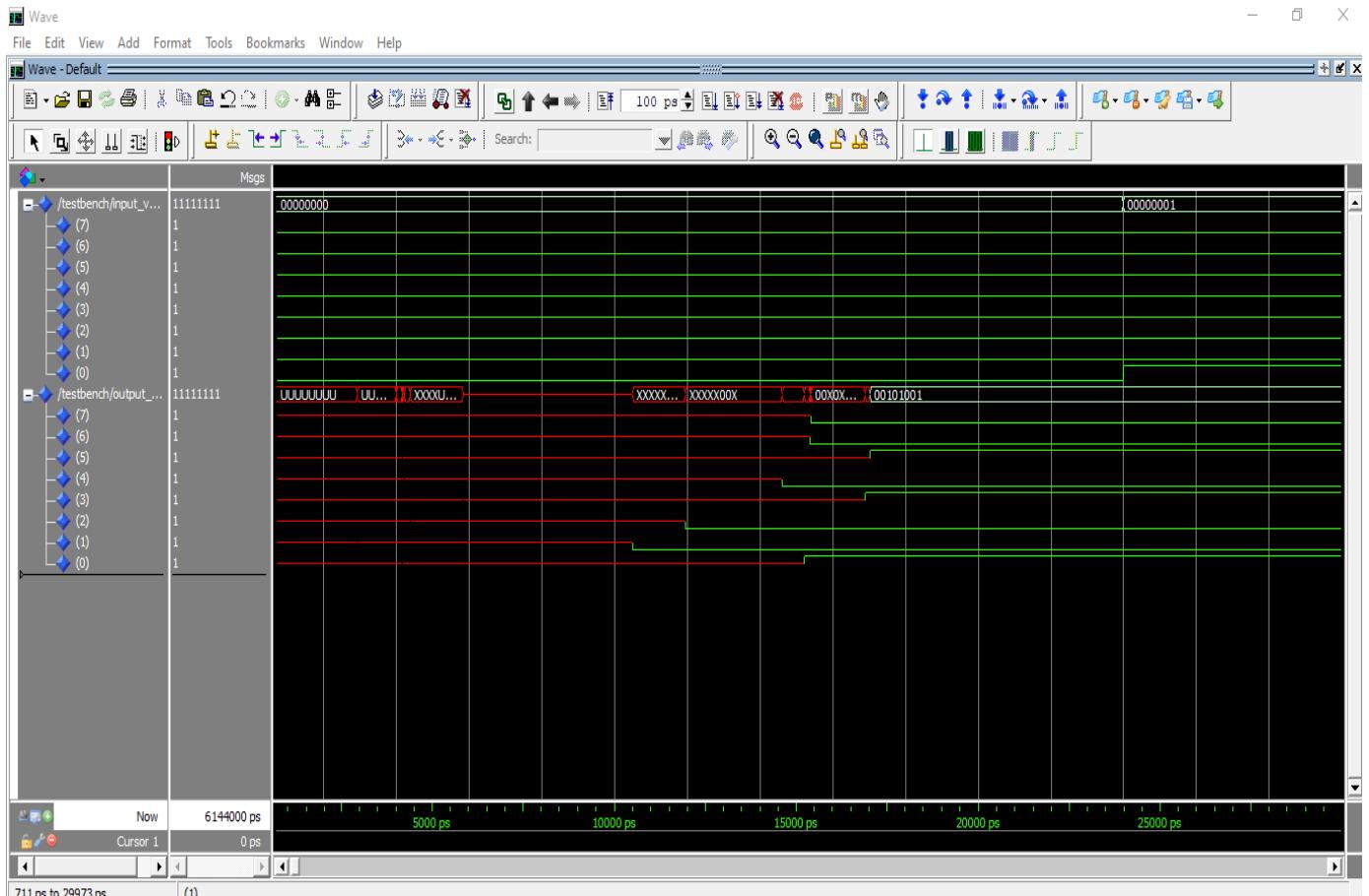
`X : in std_logic_vector(7 downto 0)`
`Z : out std_logic_vector(7 downto 0)`
00101000 01010111 11111111

- Input
- Output
- Input Bits, Output Bits, Mask Bits

RTL Simulation



Gate-level Simulation



Krypton board

Pin Planner - P:/EE214/Part3/Part3 - DUT

File Edit View Processing Tools Window Help

Report

Report not available

Groups Report

Tasks

Early Pin Planning

Early Pin Planning...

Run I/O Assignment Analysis

Top View Wire Bond

MAX V

5M1270ZT144CS

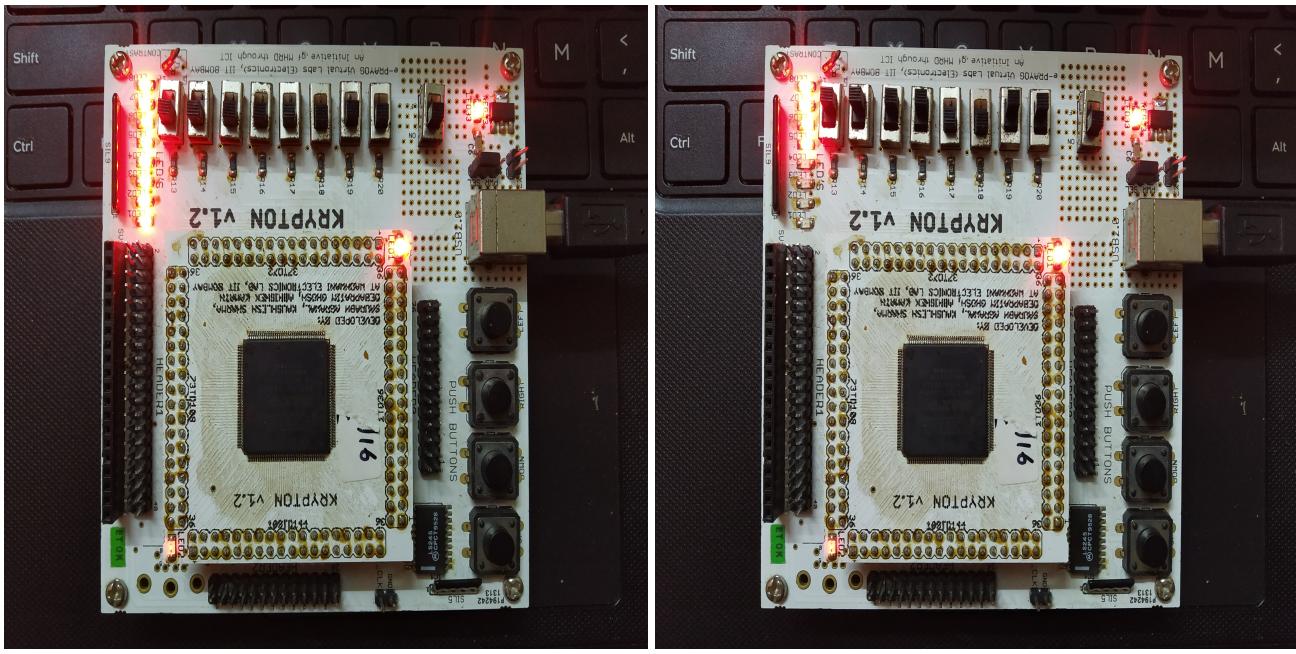
Named: * Edit Filter: Pins: all

Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair	Strict Preservation
in_input_vector[7]	Input	PIN_48	4	PIN_48	3.3-V LVTTL		16mA (default)		
in_input_vector[6]	Input	PIN_45	4	PIN_45	3.3-V LVTTL		16mA (default)		
in_input_vector[5]	Input	PIN_44	4	PIN_44	3.3-V LVTTL		16mA (default)		
in_input_vector[4]	Input	PIN_43	4	PIN_43	3.3-V LVTTL		16mA (default)		
in_input_vector[3]	Input	PIN_42	4	PIN_42	3.3-V LVTTL		16mA (default)		
in_input_vector[2]	Input	PIN_41	4	PIN_41	3.3-V LVTTL		16mA (default)		
in_input_vector[1]	Input	PIN_40	4	PIN_40	3.3-V LVTTL		16mA (default)		
in_input_vector[0]	Input	PIN_39	4	PIN_39	3.3-V LVTTL		16mA (default)		
out_output_vector[7]	Output	PIN_58	4	PIN_58	3.3-V LVTTL		16mA (default)		
out_output_vector[6]	Output	PIN_57	4	PIN_57	3.3-V LVTTL		16mA (default)		
out_output_vector[5]	Output	PIN_55	4	PIN_55	3.3-V LVTTL		16mA (default)		
out_output_vector[4]	Output	PIN_53	4	PIN_53	3.3-V LVTTL		16mA (default)		
out_output_vector[3]	Output	PIN_52	4	PIN_52	3.3-V LVTTL		16mA (default)		
out_output_vector[2]	Output	PIN_51	4	PIN_51	3.3-V LVTTL		16mA (default)		
out_output_vector[1]	Output	PIN_50	4	PIN_50	3.3-V LVTTL		16mA (default)		
out_output_vector[0]	Output	PIN_49	4	PIN_49	3.3-V LVTTL		16mA (default)		

All Pins

0% 00:00:00

Pin Planner



Input : 11111111 Output : 11111111

Input : 00001100 Output : 00001111

Observations

Some of the observations are :

Digit A	Digit B	Expected Output	Observed Output
0100	0100	AB + 29	01110011
0110	1101	A 1111	01101111
1100	0011	1111 B	11110011
1011	1100	1111 1111	11111111

Tiva C

After generating all 256 testcases in in.txt, Tiva-C compiled outputs in output.txt.
Image attached below shows the successful compilation.



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window contains the following text output from a Tiva-C compilation:

```
#----- Command - 509 : SDR 8 TDI(FE) 8 TDO(FF) MASK(FF) -----#
Successfully entered the input..
Sampling out data..
FF
:Output Comparison : Success

#----- Command - 510 : RUNTEST 1 MSEC -----#
Successfully entered the input..
Sampling out data..
FF
Output Comparison : Success

#----- Command - 511 : SDR 8 TDI(FF) 8 TDO(FF) MASK(FF) -----#
Successfully entered the input..
Sampling out data..
FF
Output Comparison : Success
:OK. All Test Cases Passed.
:Transaction Complete.

P:\EE214\Expt1\Part3>■
```

References

EE 309 : Microprocessors by Prof. Dinesh Sharma Video Lectures for understanding the logic and design of BCD.