DIGITAL DESIGN

ASSIGNMENT REPORT

ASSIGNMENT ID :1

PART 1: DIGITAL DESIGN THEORY

Provide your answers here:

Question1

1 KB = 2^10 Byte 1 MB = 2^20 Byte 1 GB = 2^30 Byte

a. 64 x 2^10 = 65,536 Byte

b.128 x 2^20 = 134,217,728 Byte

c. 6.4 x 2^30 = 6,871,947,673.6 Byte

Question2

a. 11111111111111

b. in decimal: 16383

c. in hexadecimal: 3FFF

Question3

a. Decimal number 184 to binary: 10111000

b. Decimal number 184 to hexadecimal: B8

hexadecimal B8 to binary 10111000

Method2 is faster, because 1bit of hexadecimal equals to 4bits of binary number

Question4

a. 10^8-27904836=72095164

b. 10^8-63325006=36674994

Question5

a. 10000-C6EF=3911

b. 1100 0110 1110 1111

c. 0011 1001 0001 0001

d. 3911, the answer is same as (a.

Question6

a. 10011.101

b. 1.01010101

convert from binary to decimal is 1.33203125, the difference is 0.00130208

c. 1.55

convert the result to decimal is 1.33203125, the answer is same.

When we do hexadecimal conversion, especially in binary and hexadecimal number, their negative powers are not infinite number. So, if we get a fix number, and do hexadecimal conversion among decimal, binary, and hexadecimal, the value will not change.

Question7

a. BCD: 0110 0101 0000 0011

b. excess-3:1001 1000 0011 0110

c. 8, 4, -2, -1: 1010 1011 0000 0101

d. 6311: 1000 0111 0000 0100

Question8

A = 10100101

B = 00011010

a. AND:00000000----00

b. OR:10111111----BF

c. XOR:10111111----BF

d. NOT A: 01011010----5A

e. NOT B: 11100101----E5

f. NAND: 11111111----FF

g. NOR: 01000000----40

PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

Verilog design code

```
module add(
        input signed [1:0] in1,
        input signed [1:0] in2,
        output signed [1:0] out1,
        output signed   [1:0] out2,
        output signed   [2:0] out3
        );
        assign out1 = in1;
        assign out2 = in2;
        assign out3 = in1 + in2;
endmodule
```

Truth-table

Input: Inn1, Inn2 Output: Out1 = Inn1

Out2 = Inn2, Out3 = Inn1 + Inn2(Signed)

Inn1	Inn2	Out1	Out2	Out3
00	**00**	**00**	**00**	**000**
00	**01**	**00**	**01**	**001**
00	**10**	**00**	**10**	**110**

00	11	00	11	101
01	00	01	00	001
01	01	01	01	010
01	10	01	10	101
01	11	01	11	000
10	00	10	00	110
10	01	10	01	101
10	10	10	10	100
10	11	10	11	111
11	00	11	00	101
11	01	11	01	000
11	10	11	10	111
11	11	11	11	110

SIMULATION

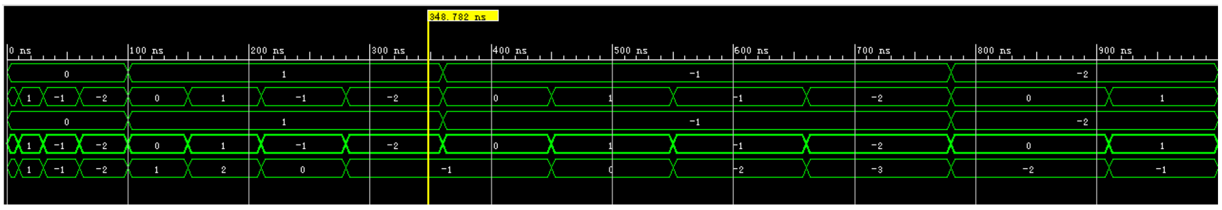
Simulation code

```
module add_tb();
```

```
reg signed [1:0] simin1;

reg signed [1:0] simin2;

wire signed [1:0] simout1;

wire signed [1:0] simout2;

wire signed [2:0] simout3;

add uu1(.in1(simin1), .in2(simin2),.out1(simout1),.out2(simout2),.out3(simout3));

initial

 begin

      simin1 = 0; simin2 = 0;

      #10 {simin1, simin2} = 4'b0001;

      #20 {simin1, simin2} = 4'b0011;

      #30 {simin1, simin2} = 4'b0010;

      #40 {simin1, simin2} = 4'b0100;

      #50 {simin1, simin2} = 4'b0101;

      #60 {simin1, simin2} = 4'b0111;

      #70 {simin1, simin2} = 4'b0110;

      #80 {simin1, simin2} = 4'b1100;

      #90 {simin1, simin2} = 4'b1101;

      #100 {simin1, simin2} = 4'b1111;

      #110 {simin1, simin2} = 4'b1110;

      #120 {simin1, simin2} = 4'b1000;

      #130 {simin1, simin2} = 4'b1001;

      #140 {simin1, simin2} = 4'b1011;

      #150 {simin1, simin2} = 4'b1010;

end
```

endmodule

Wave form simulation



from top to down is in1, in2, out1, out2, out3

description

The simulation result is same as the truth-table.

In out3, we have 3-bits. The first bit represents sign, the last two bits represents size, so generate simulation in signed decimal we get the answer equals to expectation.

So, the function of the design meets the expectation.

THE DESCRIPTION OF OPERATION

How to represent sign

During the coding, I find that in order to represent the sign of the sum of two 2-bits variables, we must use another 3-bits variable in order to use another bit to represent the sign.

How to display signed decimal

In vivado, after we simulation the file, in the test bench, we must choose “signed decimal” selection in order to get the signed decimal view.

PART 2: DIGITAL DESIGN LAB (TASK2)

TRUTH TABLE

Input: X,Y

Output: Out

	X	Y	Out
(x+y)’	0	0	1
	0	1	0
	1	0	0
	1	1	0
x’y’	0	0	1
	0	1	0
	1	0	0
	1	1	0
(xy)’	0	0	1
	0	1	1
	1	0	1
	1	1	0
x’+y’	0	0	1
	0	1	1
	1	0	1
	1	1	0