DIGITAL DESIGN

ASSIGNMENT REPORT

ASSIGNMENT ID : I

Student Name: 郑鑫颖

Student ID: 11912039

PART 1: DIGITAL DESIGN THEORY

Provide your answers here:

(a) 64k byte = 64×1024 byte = 65536 byte

(b) 128 M byte = 128×1024^2 byte = 134217728 byte

(c) 6.4 G byte = ~~128~~ 6.4 $\times 1024^3$ byte = 12202396.6 byte

Digital design assignment 1 郑鑫颖 11912039

~~1 (a) 64×10^3 (b) 128×10^6 (c) 6.4×10^9~~

2. $(\underline{11}\,\underline{1111}\,\underline{1111}\,\underline{1111})_2 = (16383)_{10} = (3FFF)_{16}$

3.
$$\begin{array}{r|l}
2 & 184 \quad 0 \\
2 & 92 \quad 0 \\
2 & 46 \quad 0 \\
2 & 23 \quad 1 \\
2 & 11 \quad 1 \\
2 & 5 \quad 1 \\
2 & 2 \quad 0 \\
 & 1
\end{array}
\qquad
\begin{array}{r|l}
16 & 184 \quad 8 \\
 & 11
\end{array}$$

$(184)_{10} = (B8)_{16} = (1011\,1000)_2 \rightarrow$ faster

$(184)_{10} = (10111000)_2$

4. (a) 27904836 — 9's complement = 72095163; 10's complement = 72095164

(b) 63325006 — 9's complement = 36674993; 10's complement = 36674994

5. (a)
$$\begin{array}{r}
FFFF \\
-\,C6EF \\
\hline
3910 \\
+\quad\;\; 1 \\
\hline
(3911)_{16}
\end{array}$$

(b) $C \rightarrow 1100$
$6 \rightarrow 0110$
$E \rightarrow 1110$
$F \rightarrow 1111$

$(C6EF)_{16} = (\underline{1100}\;\underline{0110}\;\underline{1110}\;\underline{1111})_2$

(c)
$$\begin{array}{r}
1111\;1111\;1111\;1112 \\
-\,1100\;0110\;1110\;1111 \\
\hline
0011\;1001\;0001\;0001
\end{array}$$

(d) $(0011\,1001\,0001\,0001)_2 = (3911)_{16}$ the same in a.

6. (a) $(19.625)_{10} = 19 + 0.625 = (10011.101)_2$.

$$\begin{array}{r|l}
2 & 19 \quad 1 \\
2 & 9 \quad 1 \\
2 & 4 \quad 0 \\
2 & 2 \quad 0 \\
 & 1
\end{array}$$

$(19)_{10} = (10011)_2$

$0.625 \times 2 = 1.25 \quad 1$

$0.25 \times 2 = 0.5 \quad 0$

$0.5 \times 2 = 1.0 \quad 1$

$(0.625)_{10} = (0.101)_2$

(b) $\frac{4}{3} = 1 + \frac{1}{3}$

$\frac{1}{3} \times 2 = \frac{2}{3}$ 0
$\frac{2}{3} \times 2 = \frac{4}{3}$ 1
$\frac{1}{3} \times 2 = \frac{2}{3}$ 0
⋮

$(\frac{4}{3})_{10} \approx (1.0101010)_2 = (1+2^{-2}+2^{-4}+2^{-6})_{10} = \frac{85}{64}$

$\frac{85}{64} - \frac{4}{3} = -\frac{1}{192}$

(c) $(1.01010100)_2 = (1.54)_{16}$

$(1.54)_{16} = 1 + 5\times16^{-1} + 4\times16^{-2} = \frac{85}{64}$

because conversion between bases does not affect value

7. 6503 (a) BCD 0110 0101 0000 0011
(b) Excess-3: 1001 1000 0011 0110
(c) 8,4,-2,-1: 1010 0100 0000 0101
(d) 6311: 1000 0111 0000 0100

8. (a) A AND B = $(00000000)_2 = (00)_{16}$
(b) A OR B = $(1011\ 1111)_2 = (BF)_{16}$
(c) A XOR B = $(1011\ 1111)_2 = (BF)_{16}$
(d) NOT A = $(0101\ 1010)_2 = (5A)_{16}$
(e) NOT B = $(1110\ 0101)_2 = (E5)_{16}$
(f) A NAND B = $(1111\ 1111)_2 = (FF)_{16}$
(g) NOR = $(0100\ 0000)_2 = (\cancel{5A}\ 40)_{16}$

PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

Describe the design of your system by providing the following information:

- *Verilog design (provide the Verilog code)*

```
module Signed_Addition(
    input signed[1:0] in1,
    input signed[1:0] in2,
    output signed[1:0] o1,
    output signed[1:0] o2,
    output signed[2:0] sum
    );
    assign o1=in1,
           o2=in2,
           sum=in1+in2;

endmodule
```

- *Truth-table*

x	y	F=x+y
00	00	000
00	01	001
00	10	110
00	11	111
01	00	001
01	01	010
01	10	111
01	11	000
10	00	110
10	01	111
10	10	100
10	11	101
11	00	111
11	01	000
11	10	101
11	11	110

SIMULATION

Describe how you build the test bench and do the simulation.

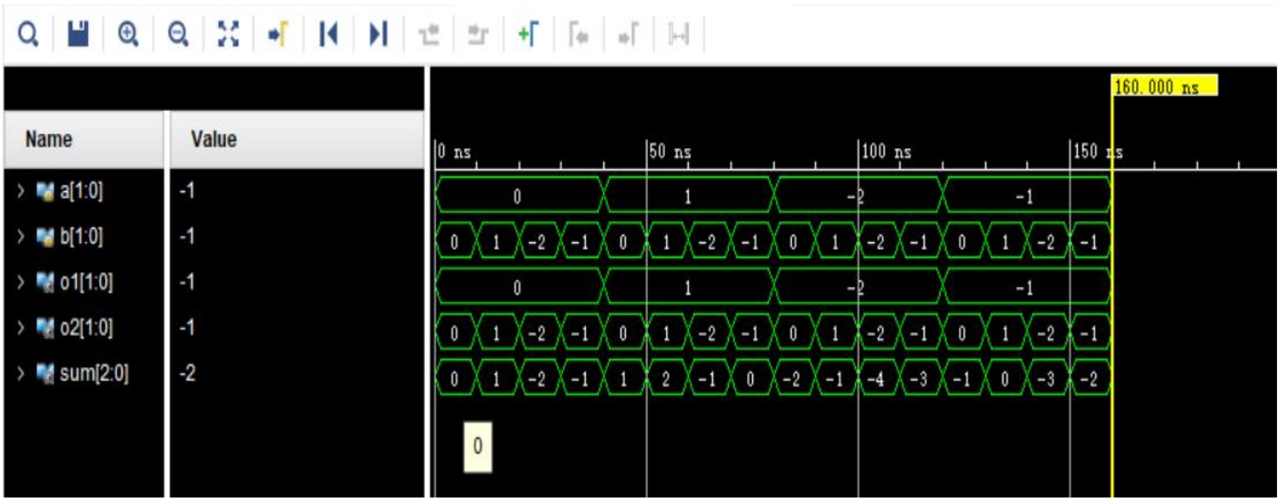
- *Using Verilog(provide the Verilog code)*

```
module Signed_Addition_sim();
  reg signed[1:0] a,b;
  wire signed[1:0] o1,o2;
  wire signed[2:0] sum;
Signed_Addition add(a,b,o1,o2,sum);

initial begin
    a=2'b0;b=2'b0;
#160 $finish;
end

always begin
    #10{a,b}={a,b}+1;
end
endmodule
```

- *Wave form of simulation result (provide screen shots)*



- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation.*

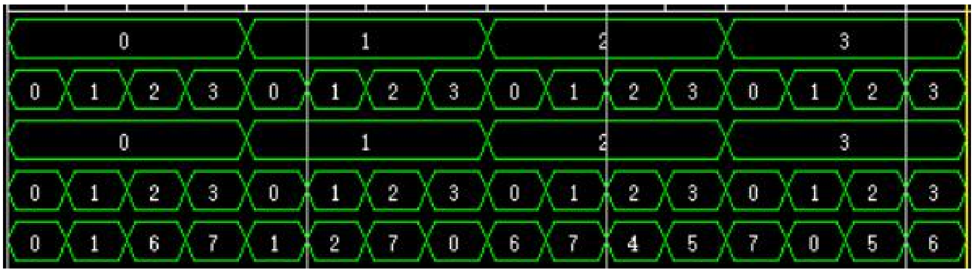
By check the truth table and the result of the simulation result, I can fine that they are the same.

THE DESCRIPTION OF OPERATION

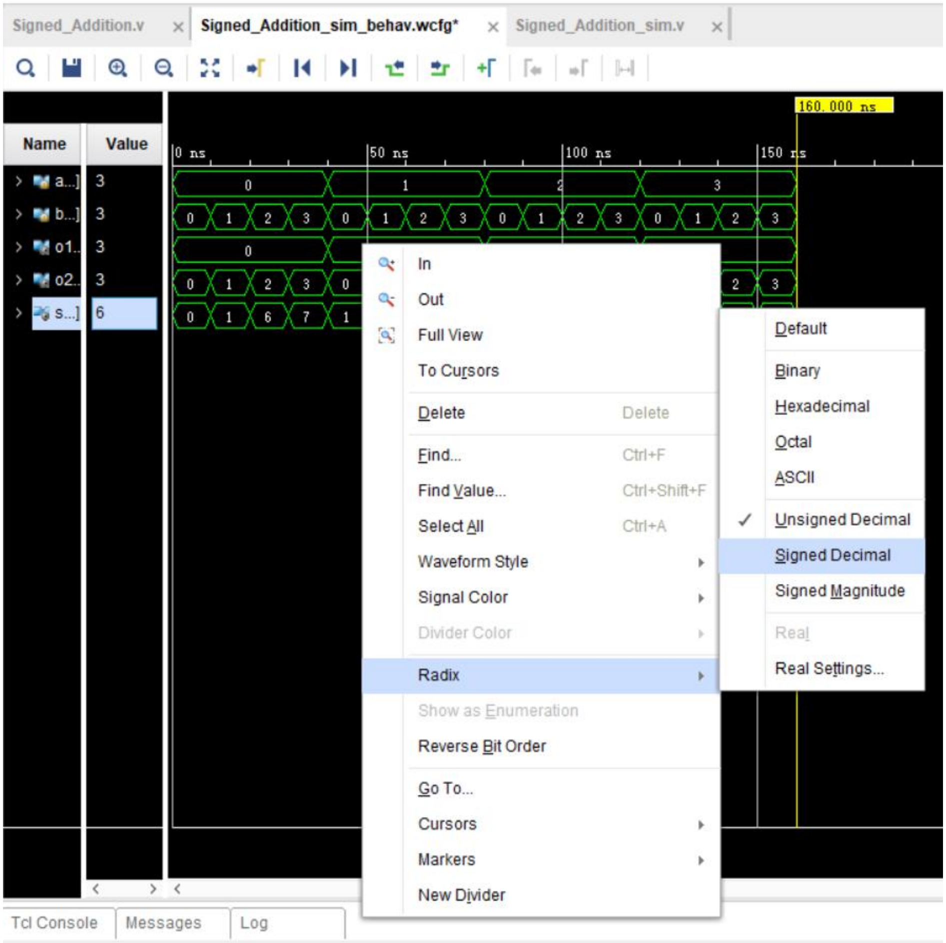
Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*

Problem1:while doing the simulation, I find that although I have made my input and output “signed”, it still displays unsigned data. Like the picture shows below.



Solution:



Problem2：I was wondering whether I can omit the “signed” before the output “sum”, since it is made by adding two signed numbers. it's a signed number by default

Solution:TA told me it's better to add that "signed" to avoid mistakes, and I also find in the PPT that :

Attention: Only when both operands are signed numbers can both operands be considered as signed numbers, otherwise both signed and unsigned numbers will be calculated as unsigned numbers.

PART 2: DIGITAL DESIGN LAB (TASK2)

DESIGN

Describe the design of your system by providing the following information:

- *Verilog design while using data flow (provide the Verilog code)*
- *Verilog design while using structured design (provide the Verilog code.*
- *Truth-table*

x	y	(x+y)'	x' y'	(xy)'	x' +y'
0	0	1	1	1	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	0	0	0	0

SIMULATION

Describe how you build the test bench and do the simulation.

- *Using Verilog (provide the Verilog code)*
- *Wave form of simulation result (provide screen shots)*
- *The description on whether the simulation result is same as the truth-table, is the function of the design meet the expectation*

CONSTRAINT FILE AND THE TESTING

Describe how you test your design on the Minisys Practice platform.

- *Constraint file (provide the screen shots on the feature of a pin and the binding info between pins and the input /output ports)*
- *The testing result (provide the screen shots (at least 3 testing scene)) to show state of inputs and outputs along with the related descriptions.*

THE DESCRIPTION OF OPERATION

Describe the problem occurred while in the lab and your solution. Any suggestions are welcomed.

- *Problems and solutions*

Problem1:Firstly, I did the structure design in the way as the following picture shows,

To use ~x and~y rather the not gate. And teacher told me the result is right but somehow don't satisfy the requirement.

```
nor   nor1(out5,x,y);
and   and1(out6,~x,~y);
nand  not1(out7,x,y);
or    or1(out8,~x,~y);
```

Solution: Then I modified my design.

```
nor nor1(out5,x,y);

not not00(notx,x);
not not01(noty,y);
and and1(out6,notx,noty);

nand not1(out7,x,y);

or  or1(out8,notx,noty);
```

Problem2: while editing a constraints file, I using out1[0] to represent the first bit of out1(out1 and out2 are both 1 bit long), as the following picture shows Then my implementation failed.

```
set_property IOSTANDARD LVCMOS33 [get_ports {out1[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {out2[0]}]
```

- Implementation (2 errors)
  - Place Design (2 errors)
    - DRC (1 error)
      - Pin Planning (1 error)
        - IO Standard (1 error)
          - [DRC BIVC-1] Bank IO standard Vcc: Conflicting Vcc voltages in bank 15. For example, the following two ports in this bank have conflicting VCCOs: out1 (LVCMOS18, requiring VCCO=1.800) and out3 (LVCMOS33, requiring VCCO=3.300)
    - [Vivado_Tcl 4-23] Error(s) found during DRC. Placer not run.

Solution: Then I ask teacher, she told me that if the port is one bit long , there's no need to add that[0].