



DIGITAL DESIGN

ASSIGNMENT REPORT

ASSIGNMENT ID : 1

Student Name: 杨钰城

Student ID: 12112323

PART 1: DIGITAL DESIGN THEORY

Provide your answers here:

12112323 杨钰城

第一次作业

Q1

$$234.5 = 234 + 0.5$$

① base 3:	$234/3$	1	%	$0.5 \times 3 = 1.5$
	$74/3$	7	4	$0.5 \times 3 = 1.5$
	$24/3$	2	4	...
	$8/3$	8	0	
	$2/3$	2	2	
	$0/3$	0	2	

$\therefore (234.5)_{10} = (22022.11)_3$

② base 5:	$234/5$	4	6	4	$0.5 \times 5 = 2.5$
	$46/5$	9		1	$0.5 \times 5 = 2.5$
	$9/5$	1		4	$\therefore (234.5)_{10} = (11414.22)_5$
	$1/5$	0		1	

③ base 6:	$234/6$	3	9	0	$0.5 \times 6 = 3$
	$39/6$	6		3	$\therefore (234.5)_{10} = (1030.3)_6$
	$6/6$	1		0	
	$1/6$	0		1	

④ base 12:	$234/12$	1	9	6	$0.5 \times 12 = 6$
	$19/12$	1		7	$(234.5)_{10} = (176.6)_{12}$
	$1/12$	0		1	

⑤ base 16:	$234/16$	1	4	10	$0.5 \times 16 = 8$
	$14/16$	0		14	$(234.5)_{10} = (EA.8)_{16}$

Q2:

$$(791)_{11} = (7 \times 11^2 + 9 \times 11^1 + 1 \times 11^0)_{10}$$

$$= (947)_{10}$$

$$10's \text{ Complement} = 1000 - 947 = 53$$

$53/11$	4	9
$9/11$	0	9

$$(53)_{10} = (49)_{11}$$

Q3:

$$\begin{aligned} \textcircled{1} (a+bt+c')(a'b'+tc) \\ = aa'b' + ac + a'b'b + bc + a'b'c' + cc' \\ = ac + bc + a'b'c' \end{aligned}$$

$$\begin{aligned} \textcircled{2} a'b'c + abc + abc + a'bc \\ = b'c(a'+a) + bc(a'+a) \\ = (b'+b)c = c \end{aligned}$$

$$\begin{aligned} \textcircled{3} (a+ca'a'+b+c)(a'+b+c) \\ = [aa' + a(b+c) + a'c + c(b+c)](a'+b+c) \\ = (ab+ac+a'c+bc+c)(a'+b+c) \\ = (ab+bc+c)(a'+b+c) \\ = abc + a'b'c + a'c + bc + c \\ = bc + a'c + b'c + c \\ = a'c + c = (a'+c)c \\ = c \end{aligned}$$

Q4:

$$\begin{aligned} F_1(A,B,C) &= \Sigma(1,2,6,7) \\ &= A'B'C + A'BC' + ABC' + ABC \\ &= A'B'C + BC' + AB \end{aligned}$$

$$\begin{aligned} F_2(A,B,C) &= \Sigma(0,1,2,3,5) \\ &= A'B'C' + A'BC' + A'BC' + A'BC + ABC' \\ &= A'B' + A'B + B'C \\ &= A' + B'C \end{aligned}$$

$$\begin{aligned} F_3(A,B,C) &= \Sigma(3,5,6,7) \\ &= A'BC + AB'C + ABC' + ABC \\ &= BC + AB + AC \end{aligned}$$

Q5:

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	1	1	1	0
10	0	0	1	1

$$\begin{aligned} \textcircled{1} F_1(A,B,C,D) &= \Sigma(0,2,3,6,7,10,11,12,13,15) \\ &= A'B'D' + A'C + ABC' + CD + AB' \end{aligned}$$



$$F_2(A, B, C, D) = \sum(1, 9, 10, 12, 13, 14) + d(4, 5, 8)$$

$$= C'D + AD' + AC'$$

AB \ CD	00	01	11	10
00	0	1	0	0
01	X	X	0	0
11	1	1	0	1
10	X	1	0	1

$$F_3(W, X, Y, Z) = \prod(0, 2, 6, 11, 13, 14, 15) + d(1, 3, 9, 10, 12)$$

$$g = \overline{a}b + d)(b' + c' + d)(a' + c + d')$$

$$= (W + X)(Y + Z)(W' + Z')$$

WZ \ XY	00	01	11	10
00	0	X	X	0
01	1	1	1	0
11	X	0	0	0
10	1	X	0	X

Q6:

$$f = abd' + c'd + a'cd' + b'cd'$$

$$g = (a + btd')(b' + c' + d)(a' + c + d')$$

ab \ cd	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	1	1	0	1
10	0	1	0	1

$$= (c' + d')(a + c + d)(b + c + d)$$

ab \ cd	00	01	11	10
00	1	0	0	1
01	1	1	1	0
11	1	0	1	0
10	1	0	1	1

$$fg = (c' + d')(a + c + d)(b + c + d)(a + btd')(b' + c' + d)(a' + c + d')$$

$$= b'cd' + a'b'c'd + abc'd'$$

ab \ cd	00	01	11	10
00	0	0	0	1
01	0	1	0	0
11	1	0	0	0
10	0	0	0	1

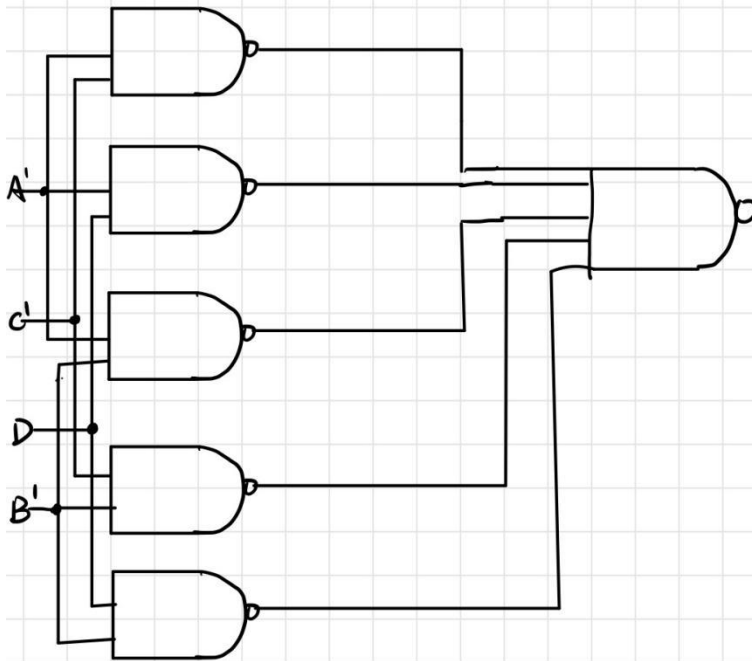
Q7: $F(A,B,C,D) = \sum(1,2,4,7,8,9,11) + d(0,3,5)$

AB\CD	00	01	11	10
00	X	1	X	1
01	1	X	1	0
11	0	0	0	0
10	1	1	1	0

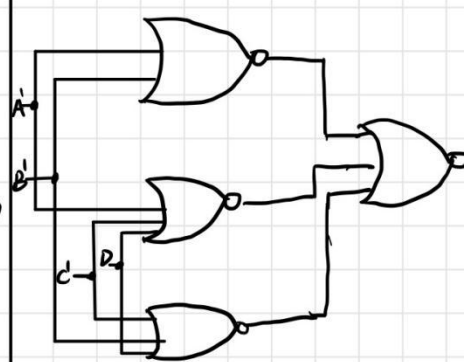
$$= A'B' + A'C' + A'D + B'C' + B'D$$

$$= (A'+B' \times B' + C' + D)(A' + C' + D)$$

① use NAND gates only



② use NOR gates only



Q8

① $F_1(A,B,C,D) = \sum(1,3,5,7) = \prod(0,2,4,6,8,9,10,11,12,13,14,15)$

$F_2(A,B,C,D) = \sum(2,3,6,7) = \prod(0,1,4,5,8,9,10,11,12,13,14,15)$

$F_1 F_2(A,B,C,D) = \prod(0,1,2,4,5,6,8,9,10,11,12,13,14,15) = \sum(3,7)$

$= A'CD$

AB\CD	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	0	0
10	0	0	0	0

$\textcircled{2} F_1(A,B,C,D) = \sum(1,3,5,6,8,10,11,12,13) = \prod(0,2,4,7,9,14,15)$
 $F_2(A,B,C,D) = \sum(0,3,5,8,9,11,13,15) = \prod(1,2,4,6,7,10,12,14)$
 $F_1 F_2 = \prod(0,1,2,4,6,7,9,10,12,14,15) = \sum(3,5,8,11,13)$

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	1	0	0
11	0	1	0	0
10	1	0	1	0

$$= AB'C'D' + BCD + B'CD$$

$\textcircled{3} F_1 F_2 = \prod(0,1,3,6,7) = \sum(2,4,5,8,9,10,11,12,13,14,15)$

AB \ CD	00	01	11	10
00	0	0	0	1
01	1	1	0	0
11	1	1	1	1
10	1	1	1	1

$$= A + BC' + ACD'$$



PART 2: DIGITAL DESIGN LAB (TASK1)

DESIGN

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

- Verilog design (provide the Verilog code)
- Truth-table

Lab 1

• Simplify the circuit by using karnaugh map.

Before the simplification, there are only ? not gate(s), ? and gate(s) and ? or gate(s) in the circuit.

a[1]	a[0]	b[1]	b[0]	sum[0]
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

After simplified by using karnaugh map, the circuit about sum[0] and a,b in Verilog is:
`assign sum[0] = ~a[0]&b[0] + ~b[0]&a[0];`

There are only ? not gate(s), ? and gate(s) and ? or gate(s).

a[1]	a[0]	b[1]	b[0]	P[0]	P[1]	P[2]	P[3]
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	0
1	1	0	0	0	0	1	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	0	0	1

Product[0]		b[1] b[0]			
		00	01	11	10
a[1] a[0]	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	0
	10	0	0	0	0

$Product[0] = a[0] \oplus b[0]$

Product[1]		b[1] b[0]			
		00	01	11	10
a[1] a[0]	00	0	0	0	0
	01	0	0	1	1
	11	0	1	0	1
	10	0	1	1	0

$Product[1] = a[0] \& b[1] \& \sim b[0] \mid a[0] \& \sim a[1] \& b[0] \mid a[1] \& \sim b[1] \& b[0] \mid a[1] \& a[0] \& b[1]$

Product[2]		b[1] b[0]			
		00	01	11	10
a[1] a[0]	00	0	0	0	0
	01	0	0	0	0
	11	0	0	0	1
	10	0	0	1	1

$Product[2] = a[1] \& a[0] \& b[1] \mid a[1] \& b[1] \& \sim b[0]$

Product[3]		b[1] b[0]			
		00	01	11	10
a[1] a[0]	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	0
	10	0	0	0	0

$Product[3] = a[1] \& a[0] \& b[1] \& b[0]$

```

module UnsignedMultiplier(
input [1: 0] in1,
input [1: 0] in2,
output [3: 0] product_led
);
assign product_led[0]=in1[0]&in2[0];
assign product_led[1]=(in1[0]&in2[1]&~in2[0])|(in1[0]&~in1[1]&in2[1])|(in1[1]&~in2[1]&in2[0])|(in1[1]&~in1[0]&in2[0]);
assign product_led[2]=(in1[1]&~in1[0]&in2[1])|(in1[1]&in2[1]&~in2[0]);
assign product_led[3]=in1[1]&in1[0]&in2[1]&in2[0];
endmodule

```

SIMULATION

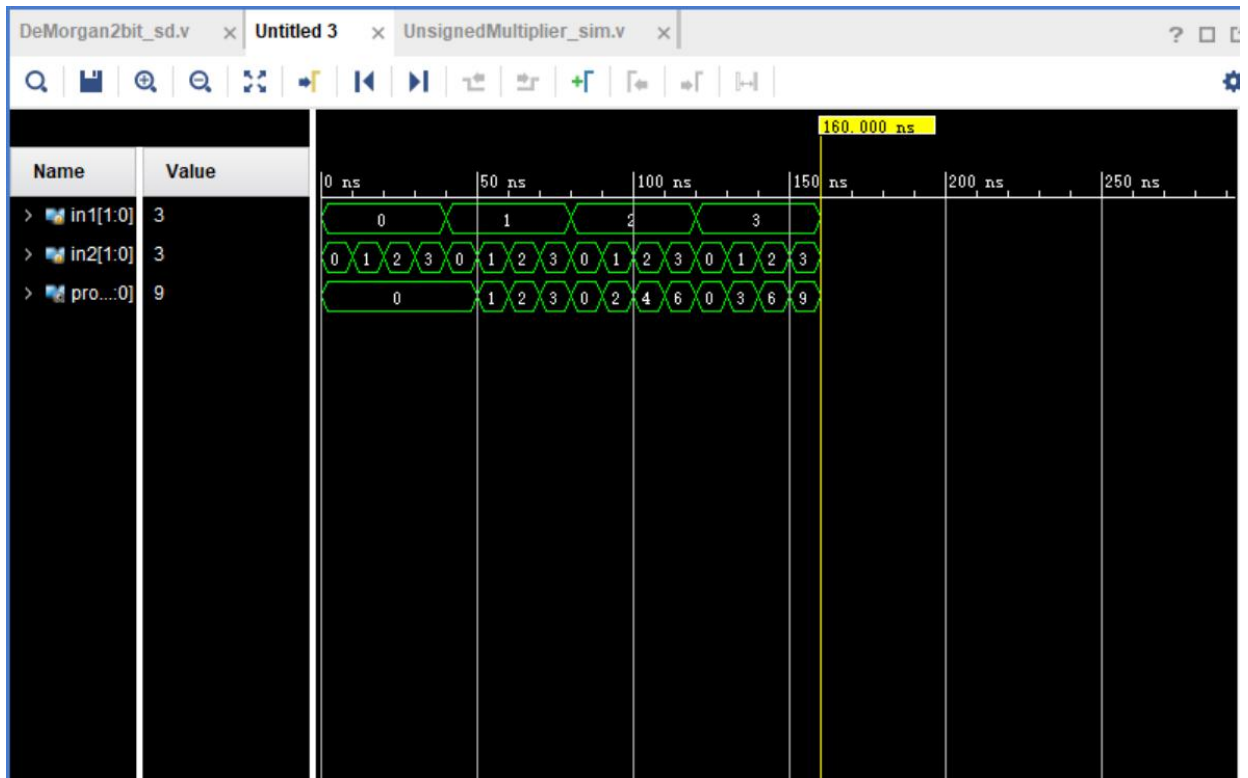
Describe how you build the test bench and do the simulation.

- *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.*

```

module UnsignedMultiplier_sim( );
reg [1:0] in1=2'b00;
reg [1:0] in2=2'b00;
//connect to output
wire [3:0] product_led;
UnsignedMultiplier fk(
.in1(in1), .in2(in2), .product_led(product_led)
);
initial begin
    in1=2'b00; in2 = 2'b00;
    repeat(15) #10 {in1,in2} = {in1,in2} + 1;
    #10 $finish();
end
endmodule

```

- Through the simulation result ,we can see the result is obviously right as the truth-table,and we use truth-table to design this logic gate,so we can think that the function of the design meet the expectation.

THE DESCRIPTION OF OPERATION

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

- Problems and solutions

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

Task 2.

$(A+B)' = A'B'$

$a[0]$	$b[0]$	$\sim(a[0] \mid b[0])$	$\sim a[0] \& \sim b[0]$
0	0	1	1
	1	0	0
1	0	0	0
	1	0	0

$a[1]$	$b[1]$	$\sim(a[1] \mid b[1])$	$\sim a[1] \& \sim b[1]$
0	0	1	1
	1	0	0
1	0	0	0
	1	0	0

$(AB)' = A' + B'$

$a[0]$	$b[0]$	$\sim(a[0] \& b[0])$	$\sim a[0] \mid \sim b[0]$
0	0	1	1
	1	1	1
1	0	1	1
	1	0	0

$a[1]$	$b[1]$	$\sim(a[1] \& b[1])$	$\sim a[1] \mid \sim b[1]$
0	0	1	1
	1	1	1
1	0	1	1
	1	0	0

```

23 module DeMorgan2bit_df(
24   input  [1: 0]in1,[1: 0]in2,
25   output  [1: 0]out11,[1: 0]out22,[1: 0]out33,[1: 0]out44
26 );
27   assign out11[0]=~(in1[0]||in2[0]);
28   assign out11[1]=~(in1[1]||in2[1]);
29   assign out22[0]=~in1[0]&&~in2[0];
30   assign out22[1]=~in1[1]&&~in2[1];
31   assign out33[0]=~(in1[0]&&in2[0]);
32   assign out33[1]=~(in1[1]&&in2[1]);
33   assign out44[0]=~in1[0]||~in2[0];
34   assign out44[1]=~in1[1]||~in2[1];
35 endmodule
36

```

```

module DeMorgan2bit_df2(
input  [1: 0]in1,[1: 0]in2,
output [1: 0]out1,[1: 0]out2,[1: 0]out3,[1: 0]out4
);
wire ou1,ou2,ou3,ou4,ou5,ou6,ou7,ou8,ou9,ou10,ou11,ou12,ou13;
or u1(ou1,in1[0],in2[0]);
not u2(out1[0],ou1);
or u3(ou2,in1[1],in2[1]);
not u4(out1[1],ou2);
not u5(ou3,in1[0]);
not u6(ou4,in2[0]);
and u7(out2[0],ou3,ou4);
not u8(ou5,in1[1]);
not u9(ou6,in2[1]);
and u10(out2[1],ou5,ou6);
and u11(ou7,in1[0],in2[0]);
not u12(out3[0],ou7);
not u13(ou8,in1[0]);
not u14(ou9,in2[0]);
or u15(out4[0],ou8,ou9);
and u16(ou10,in1[1],in2[1]);
not u17(out3[1],ou10);
not u18(ou11,in1[1]);
not u19(ou12,in2[1]);
or u20(out4[1],ou11,ou12);
endmodule

```

SIMULATION

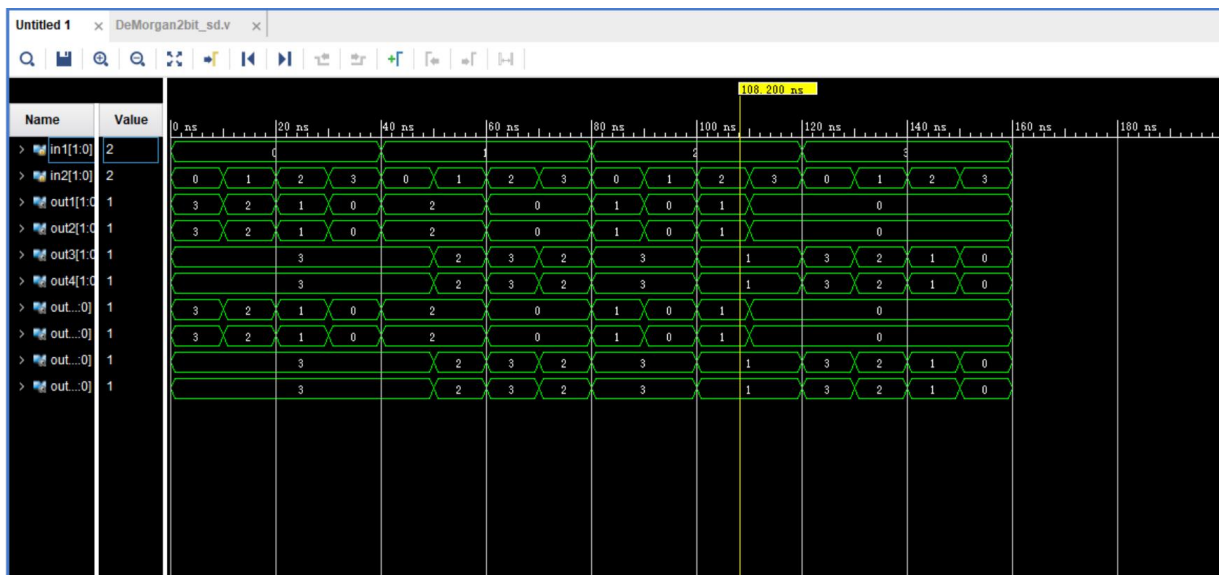
Describe how you build the test bench and do the simulation.

- *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*

```

module DeMorgan2bit_sd( );
reg [1:0]in1=2'b00;
reg [1:0]in2=2'b00;
wire [1:0]out1,out2,out3,out4,out11,out22,out33,out44;
DeMorgan2bit_df xixi(
.in1(in1),.in2(in2),.out11(out11),.out22(out22),.out33(out33),.out44(out44)
);
DeMorgan2bit_df2 haha(
.in1(in1),.in2(in2),.out1(out1),.out2(out2),.out3(out3),.out4(out4)
);
initial begin
    {in1,in2} = 2'b00;
    repeat(15) #10 {in1,in2} = {in1,in2} + 1;
    #10 $finish();
end
endmodule

```



- Through the simulation result ,we can see the result out1equals to out2 and out3 equals to out4,which proves DeMorgan of 2 bits is right and the result is right as the truth-table. we use truth-table to design this logic gate,so we can think that the function of the design meet the expectation.

THE DESCRIPTION OF OPERATION

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

- *Problems and solutions*

PART 2: DIGITAL DESIGN LAB (TASK3)

DESIGN

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

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

Task 3

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	0	1	1
11	1	1	1	0
10	0	0	1	1

USE Karnaugh map:

$$F(A,B,C,D) = \sum(0,2,3,6,7,10,11,12,13,15)$$

$$= A'B'D' + B'C + A'C + CD + ABC'$$

USE Sum of Minimum

$$F(A,B,C,D) = \sum(0,2,3,6,7,10,11,12,13,15)$$

$$= A'B'C'D' + A'B'CD' + A'B'CD + A'BCD' +$$

$$A'BCD + AB'CD' + AB'CD + ABC'D' +$$

$$ABC'D + ABCD$$

★ USE product of Maximum

$$= F(A,B,C,D) = \sum(0,2,3,6,7,10,11,12,13,15)$$

$$= \prod(1,4,5,8,9,14)$$

$$= (A+B+C+D')(A+B+C+D)(A+B+C+D)(A'+B'+C'+D)(A'+B'+C'+D)(A'+B'+C'+D)$$

```

module TASK(
input  A,B,C,D,
output out1,out2,out3,out4
);

assign out1=(~A&~B&~D)|(~B&C)|(~A&C)|(C&D)|(A&B&~C);
assign out2=(~A&~B&~C&~D)|(~A&~B&C&~D)|(~A&~B&C&D)|(~A&B&C&~D)|(~A&B&C&D)|(A&~B&C&~D)|(A&~B&C&D)|(A&B&~C&~D)|(A&B&~C&D);
assign out3=(A|B|C|~D)&(A|~B|C|D)&(A|~B|C|~D)&(A|B|C|D)&(A|~B|~C|D)&(A|B|C|~D);
endmodule

```

SIMULATION

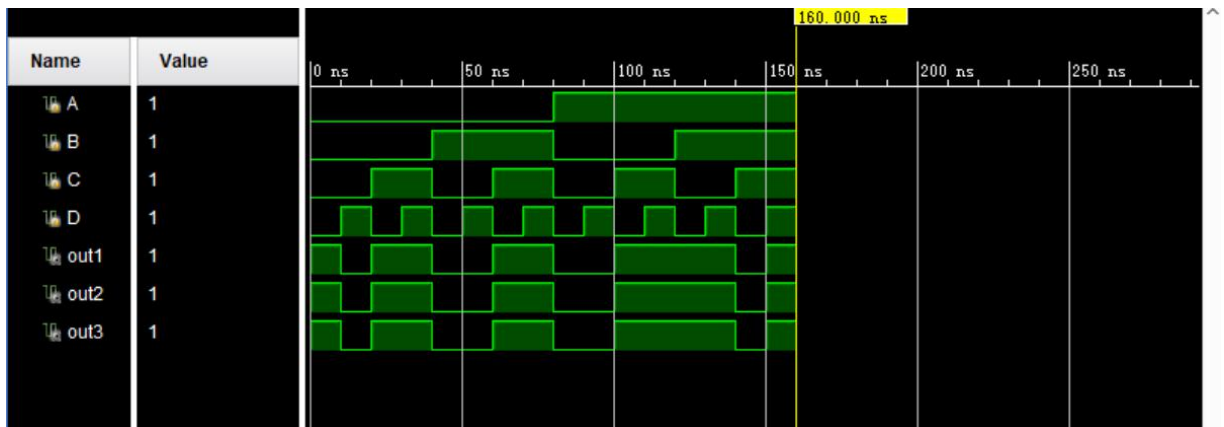
Describe how you build the test bench and do the simulation.

- 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


```

module TASK_sim();
reg A,B,C,D;
wire out1,out2,out3;
TASK ok(
.A(A),.B(B),.C(C),.D(D),.out1(out1),.out2(out2),.out3(out3)
);
initial begin
| {A,B,C,D} = 2'b0;
repeat(15) #10 {A,B,C,D} = {A,B,C,D} + 1;
#10 $finish();
end
endmodule

```



- Through the simulation result ,we can see the result is obviously right as the truth-table,and out1 equals to out2 and out3,which proves that three expressions are logical same. we use truth-table to design this logic gate,so we can think that the function of the design meet the expectation.

THE DESCRIPTION OF OPERATION

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

- Problems and solutions