

1

Simplify the following functions using Karnaugh maps

a

$$F = \sum_{w,x,y,z} (0, 1, 4, 5, 7, 10, 11, 14, 15)$$

✓ Answer ✓

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

$$A' \cdot C' + A \cdot C + B \cdot A' \cdot D + B \cdot C \cdot D$$

b

$$F = \sum_{w,x,y,z} (0, 2, 3, 4, 6, 9, 10, 11, 15)$$

✓ Answer

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

$$A' \cdot D' + B' \cdot C + D \cdot A \cdot B' + C \cdot D \cdot A$$

2

Simplify the following functions using Karnaugh maps

a

$$F = \sum_{x,y,z} (0, 1, 3, 4, 5, 6)$$

✓ Answer

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

$$B' + A \cdot C' + A' \cdot C$$

b

$$F = \sum_{x,y,z} (1, 2, 5, 6, 7)$$

✓ Answer

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

$$A \cdot B + B \cdot C' + A \cdot C + B' \cdot C$$

3

For the logic expressions given below, find all of the static hazards and design a hazard-free circuit that realizes the same logic function. Write the functions that are hazard free, you do not need to draw the circuit. (Hint: Use Karnaugh maps to find the timing hazards.)

a

$$F = W \cdot X + W' \cdot Y'$$

✓ Answer

Y\WX	00	01	11	10
0	0	0	1	0
1	1	1	1	0

$$W \cdot X + W' \cdot Y' + X \cdot Y$$

b

$$F = W \cdot Y + W' \cdot Z' + X \cdot Y' \cdot Z$$

✓ Answer

YZ\WX	00	01	11	10
00	1	1	0	0
01	0	1	1	0
11	0	0	1	1
10	1	1	1	1

$$F = W \cdot Y + W' \cdot Z' + Y \cdot Z' + X \cdot Y' \cdot Z + W' \cdot X \cdot Y' + W \cdot X \cdot Z$$

4

Draw the circuit for the following function using only two 2-input and one 3-input NOR gates. The complements of the inputs are also available.

$$F = a \cdot b + a' \cdot b' + b' \cdot c$$

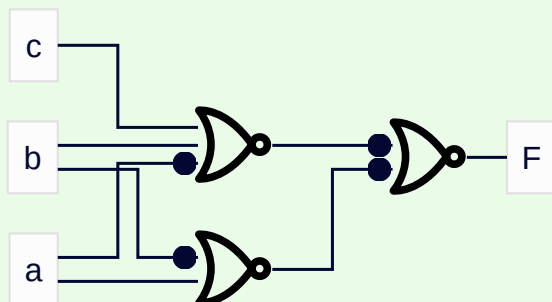
✓ Answer

$$F = a \cdot b + a' \cdot b' + b' \cdot c$$

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

$$F = (a + b') \cdot (a' + b + c)$$

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

**5**

Design a three-input logic circuit that will produce a 1-output when there are more zeros in the input combination than the ones. For example, 001 will produce a 1-output whereas 110 will produce a 0-output. First write the truth table for this problem and then find the minimal sum using a Karnaugh map. Draw the circuit using only NAND gates. The complements of the inputs are available.

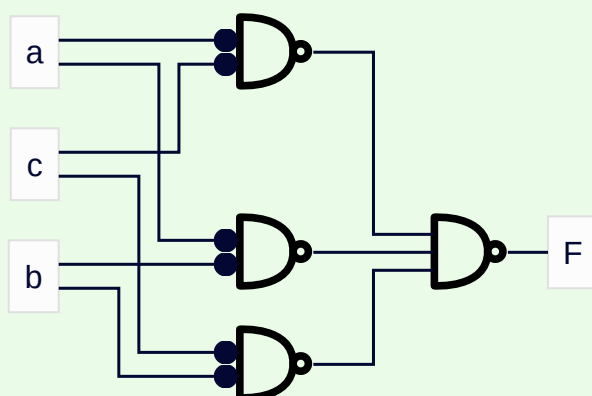
✓ Answer

a	b	c	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

c ab	00	01	11	10
0	1	1	0	1
1	1	0	0	0

$$F = a'b' + a'c' + b'c'$$

$$F = ((a'b')'(a'c')'(b'c'))'$$



Lab

Code

rc_adder_slice.sv

```

module rc_adder_slice (
    input logic a, b, c_in,
    output logic s, c_out
);

    logic p, g;

    assign p = a ^ b;
    assign g = a & b;

    assign s      = p ^ c_in;
    assign c_out = (p & c_in) | (a & b);

endmodule

```

rc_adder_parm.sv

```

module adder_parm #(
    parameter N = 4
) (
    input logic [N-1:0] a, b,
    input logic c_in,
    output logic [N-1:0] s,
    output logic c_out
);
    logic [N-1:0] p, g;
    logic [N:0] c;

    always_comb begin
        p = a ^ b;
        g = a & b;
        c[0] = c_in;
        for (int i = 0 ; i < N; i++) begin
            s[i] = p[i] ^ c[i];
            c[i+1] = (p[i] & c[i]) | (a[i] & b[i]);
        end
        c_out = c [N];
    end
endmodule

```

rc_adder4.sv

```

module rc_adder4 (
    input logic [2:0] a, b,
    input logic c_in,
    output logic [2:0] s,
    output logic c_out

```

```

);
    adder_parm #(
        3
    ) parmT (
        a, b, c_in, s, c_out
    );
endmodule

```

testbench_lab3.sv

```

`timescale 1ns/10ps

module testbench_lab3 ();

    logic [2:0] a=3'b000, b=3'b000, s;
    logic co;

    rc_adder4 UUT (
        a, b, 0, s, co
    );

    int t = 0;

    always begin
        #1 t++;
        a = t % 8;
        b = (t/8) % 8;
    end

    initial begin
        $display("TIME | A B | S CO");
        $display("-----");
        $monitor("  %2d | %d %d | %d %b",
            $time, a, b, s, co);

        #128;
        $finish();
    end

endmodule

```

Deliverables

100ns

Layout Simulate ColumnLayout AllColumns

sim - Default

Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage
testbench_lab3	testbench_...	Module	DU Instance	+ACC...	
UUT	rc_adder4	Module	DU Instance	+ACC...	
always#14	testbench_...	Process	-	+ACC...	
initial#20	testbench_...	Process	-	+ACC...	
std	std	VPackage	Package	+ACC...	
vsim_capacity#	Capacity	Statistics		+ACC...	

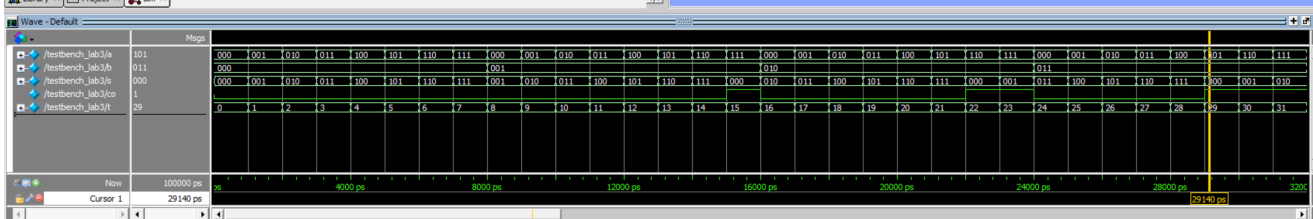
Objects

Name	Value	Kind	Mode
a	101	Pack...	Internal
b	011	Pack...	Internal
s	000	Pack...	Internal
co	1	Reg...	Internal

Processes (Active)

Name	Type (filtered)	State	Order	Parent Path	Class Info
------	-----------------	-------	-------	-------------	------------

Library Project sim



Transcript

```

# 92 | 4 | 3 | 7 | 0
# 93 | 5 | 3 | 1 | 0 | 1
# 94 | 6 | 3 | 1 | 1 | 1
# 95 | 7 | 3 | 1 | 2 | 1
# 96 | 0 | 4 | 1 | 4 | 0
# 97 | 1 | 4 | 1 | 5 | 0
# 98 | 1 | 3 | 4 | 1 | 6 | 0
# 99 | 1 | 3 | 4 | 1 | 7 | 0

V$M84> Ierror Nicholas tln32

```

Now: 100 ns Path: 4 sim:/testbench_lab3