Computer-Aided VLSI System Design
Midterm Examination
2011

Name ______________________________

Student ID ______________________________

Instructions

Exams: Consultation during the exam is not permitted. This is not an open book exam. The exam is to be completed in one and a half hours. If you need scratch paper, just use the blank parts of these pages; show all of your work on these pages. Before you start writing, please check if you have all 14 pages of the exam.

Regrading Policy: Exams will be accepted for regrading up to two weeks after you get the graded exam. No regrades after two weeks.

Please sign the following statement upon completing this exam:

I certify that I will follow the above instructions. I have neither received nor given unpermitted aids on this examination.

Your signature ______________________________

******************** **Score Board (to be filled by graders)** ******************

	Total Points	Score
Problem 1	15	
Problem 2	12	
Problem 3	12	
Problem 4	15	
Problem 5	16	
Problem 6	10	
Problem 7	20	
Total	100	

Problem 1 <Basic Concept> (15%)

Briefly explain the following terms:

(a) What is "setup time"? (3%)

(b) What is a "critical path"? (3%)

(c) Compare synchronous and asynchronous reset (3%)

(d) What is a "fault model"? (3%)

(e) Compare full-custom based and standard-cell based designs (3%)

(a) min time before the clock event by which. the input must be stable

(b) ~~MM~~ the longest path b/w two ~~register~~ .

(c) synchronous, reset happens on positive edge of clock

asynchronous, reset happens on negative edge of reset.

(d) To represent defects abstracted logic level.

Model logic behavior of defects.

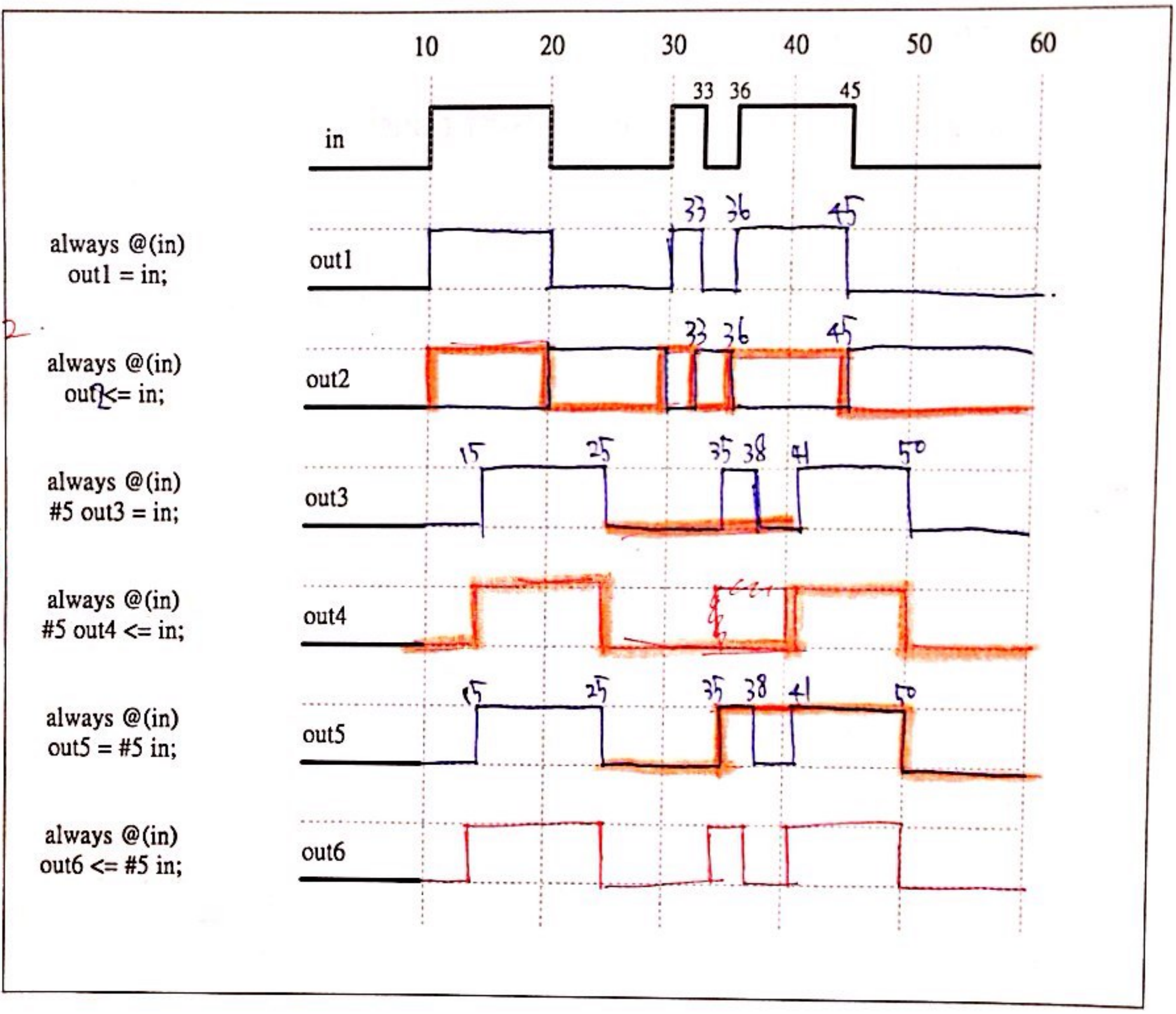
(e)

Problem 2 <Basic Concept> (12%)

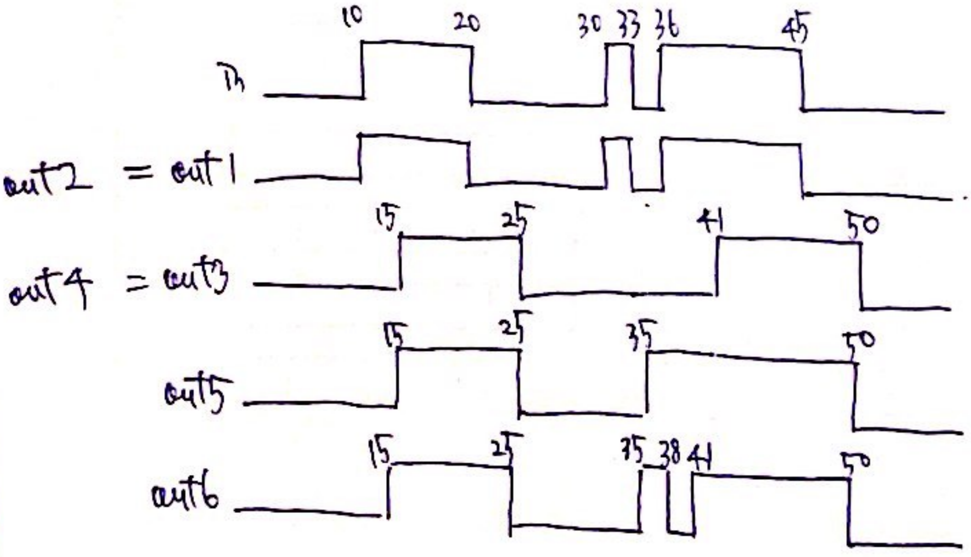
The left hand side of the table below shows some Verilog code. The top of table shows the waveform of "in" signal. Please complete the rest waveforms with respect to the "in" signal in the following table.

out 1 = out2 .

out3 = out4



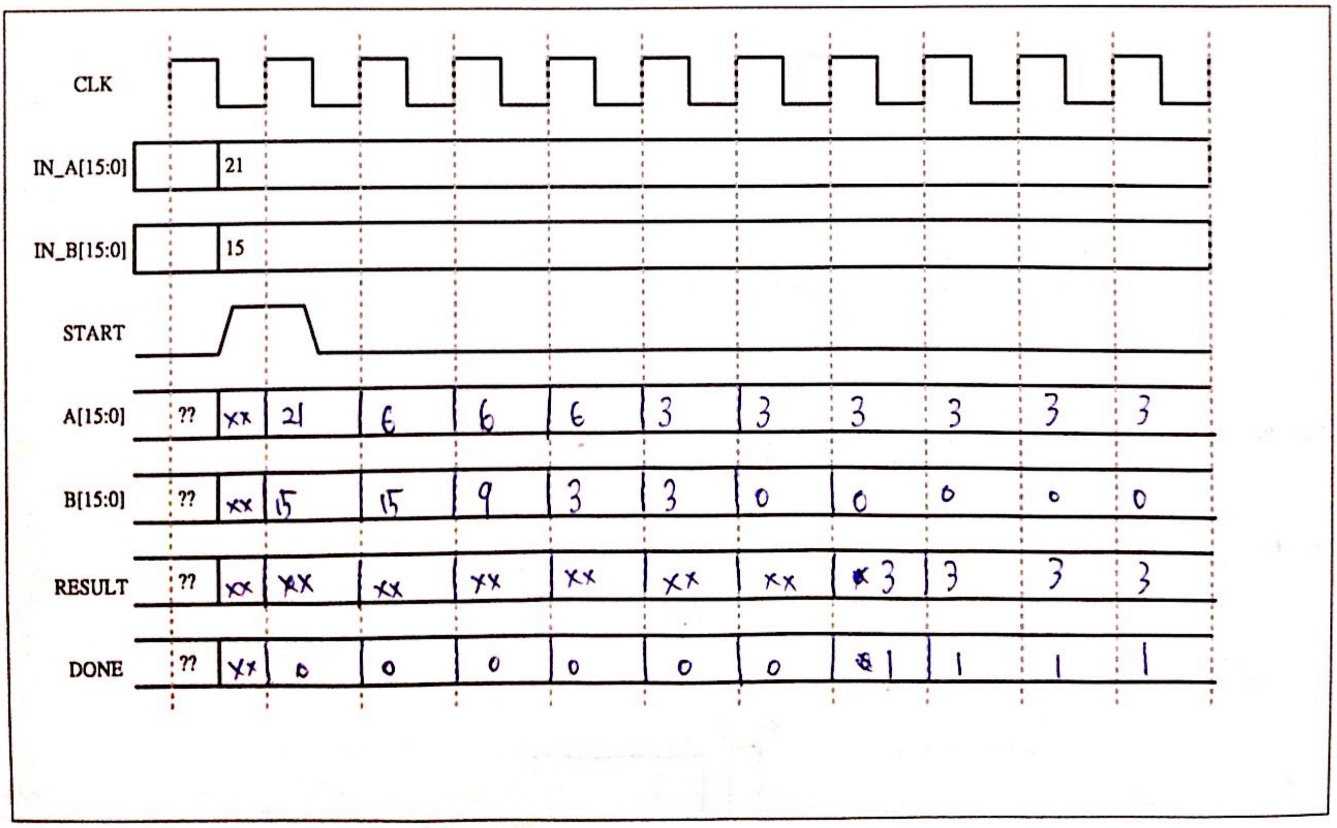
```
always @(in)
out1 = in;

always @(in)
out2 <= in;

always @(in)
#5 out3 = in;

always @(in)
#5 out4 <= in;

always @(in)
out5 = #5 in;

always @(in)
out6 <= #5 in;
```



Problem 3 <Basic Concept > (12%)

Consider the following Verilog module that computes the greatest common divisor of two 16-bit unsigned integer values IN_A and IN_B where IN_A ≥ IN_B. Please complete the timing diagram below as the module computes the GCD of 21 and 15. **Use "xx" to indicate values that can not be determined from the information given.**

```
module GCD(CLK, START, IN_A, IN_B, RESULT, DONE)
     input CLK, START;
     input [15:0] IN_A, IN_B;
     output reg [15:0] RESULT;
     output reg DONE;

     reg [15:0] A, B;
     always@(posedge CLK)
     begin
          if (START) begin
               A <= IN_A; B <=IN_B; DONE <= 0;   end
          else if (B==0) begin RESULT <= A; DONE <= 1; end
          else if (A > B) begin A <= A - B;
          else B <= B-A;
     end
endmodule
```


CLK
IN_A[15:0] 21
IN_B[15:0] 15
START
A[15:0] ?? xx 21 6 6 6 3 3 3 3 3 3
B[15:0] ?? xx 15 15 9 3 3 0 0 0 0 0
RESULT ?? xx xx xx xx xx xx xx 3 3 3 3
DONE ?? xx 0 0 0 0 0 0 1 1 1 1

Problem 4 <Verilog and Testbench> (15%)

The following Verilog test initially sets A to 1, b to 0 and CLK to 0, waits 10 time units, sets CLK to 1, waits another 10 time units and then prints out the values of A, B and C.

```
module test();
  reg CLK, A, B;
  wire C;

  assign C = ~A;


   // Additional Verilog code is insterted here


   initial begin
      A = 1;
      B = 0;
      CLK = 0;
      #10
      CLK = 1;
      #10
      $display ("A = %d, B = %d, C = %d\n", A, B, C);
      #stop;
   end
  endmodule
```

In these tests, a series of additional Verilog codes below substituted into the test module at the above indicated location. For each additional Verilog code, please print out the final values of A, B and C. **Write "unknow" for a value if it can not be determined.**

a. always @ (posedge CLK) begin a = b; b = c; end

unknown: . Values: A = 0, B = 1. , C = 1

b. always @ (posedge CLK) begin a <= b; b <= c; end

Values: A = 0, B = 0 , C = 1.

c. always @ (posedge CLK) begin a = b; end

always @ (posedge CLK) begin b = c; end

unknown Values: A = 0, B = 0 , C = 1.

d. always @ (posedge CLK) begin a <= b; end

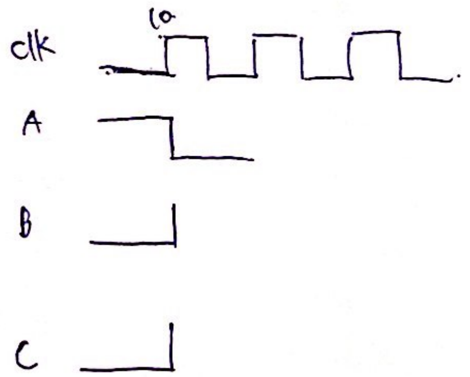
always @ (posedge CLK) begin b <= c; end

Values: A = 0, B = 0 , C = 1.

e. always @ (posedge CLK) begin a <= b; b = c; end

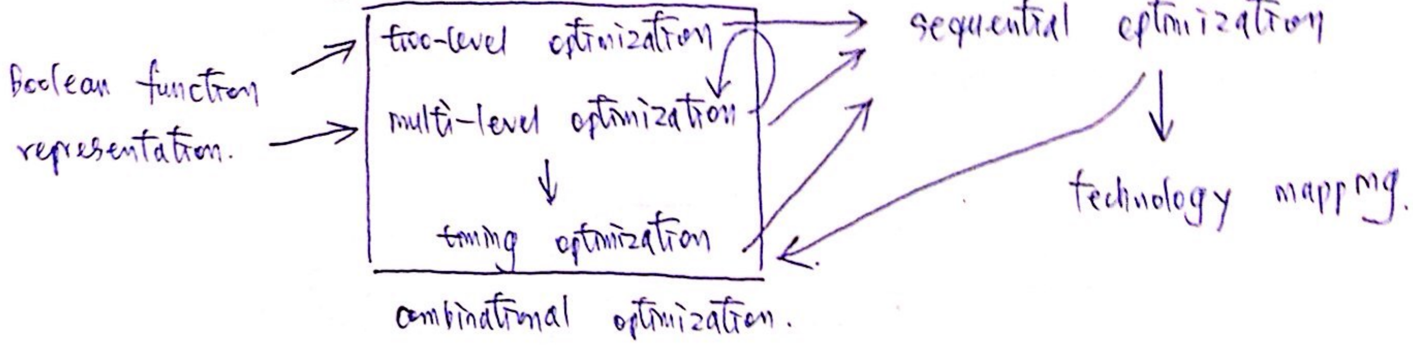
Values: A = 0., B = 1. , C = 1.

(a)



Problem 5 <Logic Synthesis> (16%)

A. Briefly describe the logic synthesis flow.



B. Consider the following Boolean matrix in the Quine-McCluskey two-level logic minimization method.

If the minterm is covered by the prime, entry is 1

1. What do M_i's and P_j's mean in this matrix? For an entry in the ith row and jth column, if the entry is 1, what does it mean? If the entry is 0, what does it mean? M: all onset minterm. P: primes computed.
2. Simplify the matrix by reduction using row dominance and column dominance. (Show immediate steps and state which rules you apply to simplify this matrix.)
3. What is your logic minimization result? P2 P5 P6 P8

	P1	P2 ✓	P3	P4 ✓	P5 ✓	P6 ✓	P7	P8 ✓	P9
~~M1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~
~~M2~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~
~~M3~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~
~~M4~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~1~~
~~M5~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
~~M6~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~
~~M7~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~
M8	1	0	0	0	0	0	0	1	1
M9	0	0	1	0	0	0	1	1	0
~~M10~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~

	1	2	3	~~4~~	6	7	8	9
1	1	1	0	0	1	1	1	0
2	0	0	1	0	1	0	0	0
3	0	1	0	0	0	1	0	1
5	0	1	0	0	0	0	0	0
7	0	0	0	1	1	0	0	1
8	1	0	0	0	0	0	1	1
9	0	0	1	0	0	1	1	0

P2 ~~P4~~ P5 P6 P8.

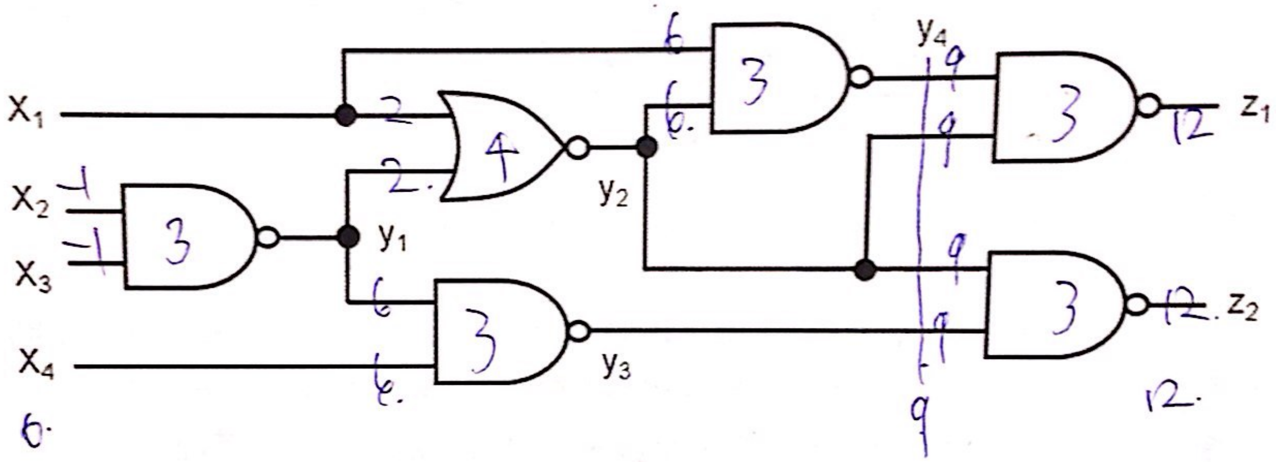
~~M1~~ ~~M2~~ ~~M3~~ ~~M4~~ ~~M5~~ ~~M6~~ ~~M7~~. ~~M8~~ ~~M9~~ ~~M10~~

P2 P5 P6 P8

~~M1~~ ~~M2~~ ~~M3~~ ~~M4~~ ~~M5~~ ~~M6~~ ~~M7~~. ~~M8~~ ~~M9~~ ~~M10~~

Problem 6 <Timing Analysis> (10%)

Calculate the arrival time, required time, and slack at each gate output. Assume the NAND gates have delay 3ns, NOR gate 4ns, arrival times at primary inputs are 0, and required times at primary outputs are 12ns.



X1: Arrival 0 ; Required 2 ; Slack 2

X2: Arrival 0 ; Required -1 ; Slack -1

X3: Arrival 0 ; Required -1. ; Slack -1

X4: Arrival 0. ; Required 6. ; Slack 6

Y1: Arrival 3 ; Required 2. ; Slack -1

Y2: Arrival 7 ; Required 6 ; Slack -1

Y3: Arrival 6 ; Required 9 ; Slack 3

Y4: Arrival 10 ; Required 9 ; Slack -1.

Z1: Arrival 13 ; Required 12 ; Slack -1

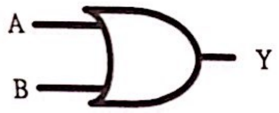
Z2: Arrival 10 ; Required 12. ; Slack 2.

Please identify the critical paths (with slack less than 0).

X2
X3 > Y1 - Y2. - Z1

Problem 7 <Design for Testability> (20%)

A. Given one OR gate for your reference below. Answer the following questions.



1. Complete the single stuck-at fault (SSF) table of two input OR gate below for the output value with stuck-at fault. **And also identify the output value at Y with stuck-at fault using "*" character (such as "1*").**

Input		Fault-free output	Output value with stuck-at fault					
A	**B**	**Y**	**A/0**	**A/1**	**B/0**	**B/1**	**Y/0**	**Y/1**
0	**0**	0	0	1*	0	1*	0	1*
0	**1**	1	1	1	0*	1	0*	1
1	**1**	1	1	1	1	1	0*	1
1	**0**	1	0*	1	1	1	0*	1

2. Find out the equivalent faults based on above SSF table.

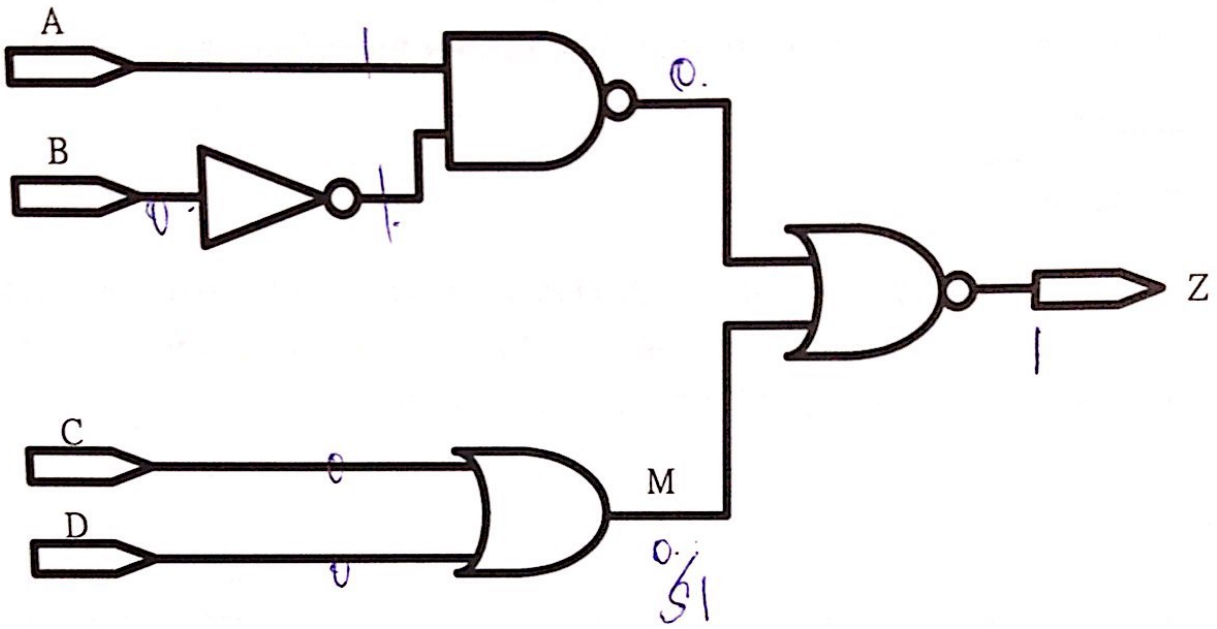
A/1 B/1 Y/1 ~~A/0~~ Y/0.

~~B/0~~ Y/0

3. How many percentage of fault coverage could be achieved If the test pattern (A, B) is {(0, 1),(1, 1)}?

2 : B/0 , Y/0.

B. Given the circuit below, please identify the test pattern for the "stuck-at-1" fault at signal M by using D-Algorithm. The fault model is based single stuck-at fault (SSF) model.



A B CD 1 000