Computer-Aided VLSI System Design
Midterm Examination
Nov. 22, 2007

Your Name _____

Student ID Number _____

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 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 15 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 aid on this examination.

Your signature _____

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

	Total points	Your points
Problem 1	15	
Problem 2	10	
Problem 3	20	
Problem 4	10	
Problem 5	10	
Problem 6	10	
Problem 7	25	
Total	100	

Problem 1 <Basic Concept>

(15%) Briefly explain the following terms:

(a) Full scan with ATPG

(b) Memory BIST Build in self test.

(c) Fault simulation

(d) RTL code

(e) Multi-cycle path

Ans:

(a) Full scan 即使用 ATPG 產生的 test vector 來找出 fault.

(b) ?

(c) 用一組 test vector 來找出它所能測出的全部 fault.

(d) RTL : register transfer level. coding 是一种可合成的电路語法.

(e) 將一下 cycle 內完成的動作改為由許多 cycle 來完成. 特別為完成各指令 (如 sw, add ... lw) 的 cycle 數不同

Problem 2 (10 points) <Procedural Block>

Given below is a testbench with sequential and parallel blocks as well as blocking and non-blocking assignments. Please show the outputs if this testbench is simulated via Verilog-XL.

```
`timescale 1ns/1ns
module test;
reg x,y,a,b;

initial begin
 x=1'b0;
 #5 y=1'b1;
 fork
  #15 a=y;
  #5 b=x;
  begin
   #10 y <= x;
       x <= y;
  end
 join
 #20 x=b;
 #5 $finish;
end

initial begin
 $monitor("%t %b %b %b %b",$time,x,y,a,b);
end
endmodule
```

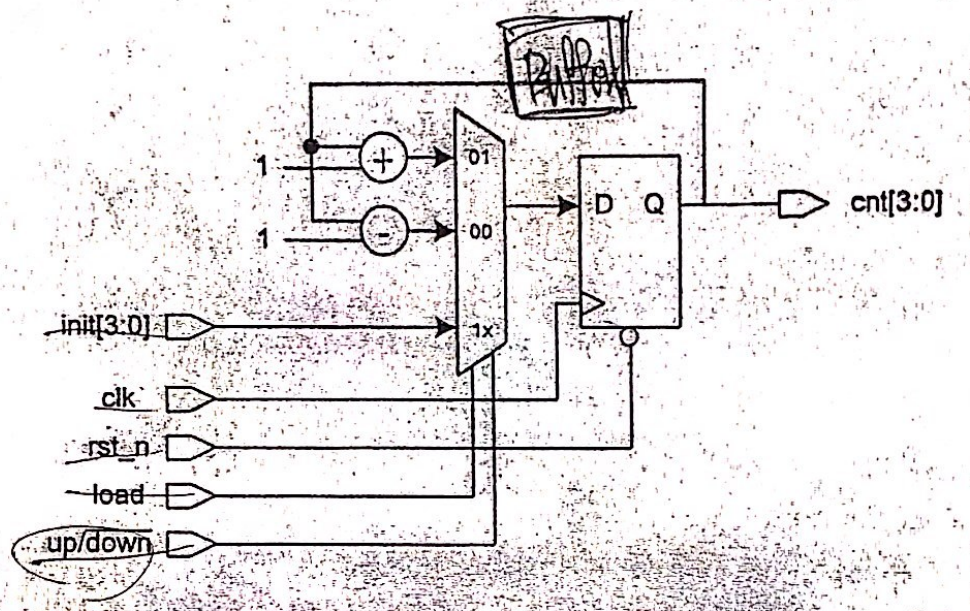
t	x	y	a	b
0	0	X	X	X
5	0	1	X	X
10	0	1	X	0
15	1	0	X	0
20	1	0	0	0
40	0	0	0	0

Ans:

	x	y	a	b
0	0	X	X	X
5	0	1	X	X
10	0	1	X	0
15	1	0	X	0
20	1	0	0	0
40	0	0	0	0

Problem 3 <Verilog Debug and Coding>

A counter with synchronized initial value loading and up count/down count function is shown below.



A. (10%) The following Verilog-RTL code for the counter is not correct. Please write the correct version in the right column.

Verilog code with bugs

```
module counter(clk, rst_n, cnt,
init, load, updown);

input clk, rst_n;
output [3:0] cnt;
input [3:0] init;
input load;
wire [3:0] cnt_w;
reg [3:0] cnt;
reg [3:0] init;

always@(cnt or init)
begin
  case({load, updown})
    2'b00: cnt_w = cnt - 1'b1;
    2'b01: cnt_w = cnt + 1'b1;
    2'b10: cnt_w = init;
    2'b11: cnt_w = init;
  endcase
end

always@(posedge clk or posedge
rst_n)
begin
  if(!rst_n)
    cnt <= 4'd0;
  else
    cnt <= cnt_w;
end

endmodule
```

Corrected Code

There are totally 9 bugs.

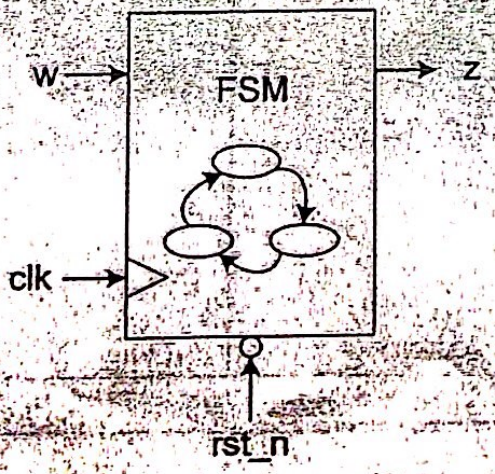
B. (10%) After synthesis, if the simulation of the netlist shows there are some hold time violations on the output register. (a) What does it mean? (b) If we want to fix it manually, please draw the corresponding circuits.

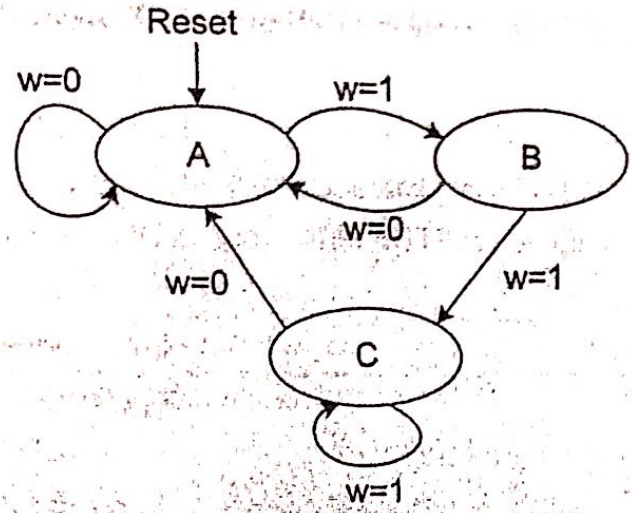
Ans:

(a).

Problem 4 <Finite State Machine>

(10%) A state diagram of a finite state machine is shown below.





State	Output z
A	0
B	0
C	1

Please complete the following Verilog code:

Ans:

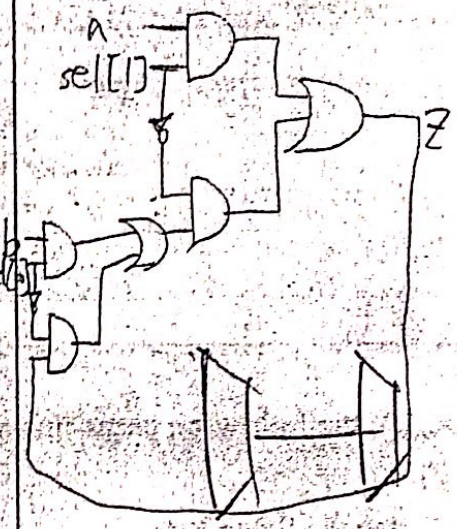
```
module FSM(clk, rst_n, z, w);

input clk, rst_n;
output z;
input w;

parameter A=2'b01, B=2'b10; C=2'b11;

reg [1:0] state; //current state
reg [1:0] next_state; //next state logic output
reg z;

//next state logic
always@( posedge clk or negedge rst_n )
begin
  If(! rst_n)
     next_state <= A;
  else if(state == A && w == 0)
     next_state <= A;
  else if(state == A && w == 1)
     next_state <= B;
  else if(state == B && w == 0)
     next_state <= A;
  else if(state == B && w == 1)
     next_state <= C;
  else if(state == C && w == 0)
```

```
end

//output logic
always@(posedge clk or negedge rst_n or state)
begin
    if(!rst_n)
        z<=0;
    else if(state==A)
        z<=0;
    else if
end

//state
always@(posedge clk or negedge rst_n)
begin
    if(!rst_n)
    state<=A;
    else

end

endmodule
```

Problem 5 <Manually Synthesis from Verilog RTL Code>

(10%) In the following table, the left column show some pieces of Verilog RTL code. Please draw the corresponding circuits in the right column. You can use AND, OR, NAND, NOR, XOR, XNOR, NOT, MUX in the circuit diagram.

Verilog Code	Circuit Diagram

always @(a or b or sel)
begin
if (sel[1])
z = a;
else if (sel[0])
z = b;
end
	

always @(a or b or c)
begin
 	
for(k=0; k<=2; k=k+1)
begin
out[k]=a[k]^b[k];
c=(a[k]|b[k])&c;
end
 	
end	

Problem 6 (16%) <Synthesis with Design Vision>

A. (6%) In Design Vision, the warning message "multiple design instance" results from that you use the same HDL description to represent more than one design instance, how do we to handle it? There maybe several methods to handle it, what is the property of each method?

Ans:

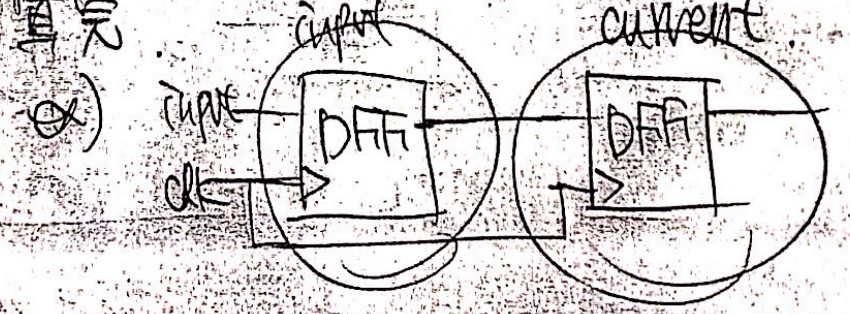
① dont touch hierarchy will be maintained
② Ungroup does not preserve hierarchy
③ Uniquify create a unique design file for each instance

B. (4%) Describe the following terms in synthesis (you can draw the diagram to explain it)
(a)Input Delay (b) False Path

Ans:

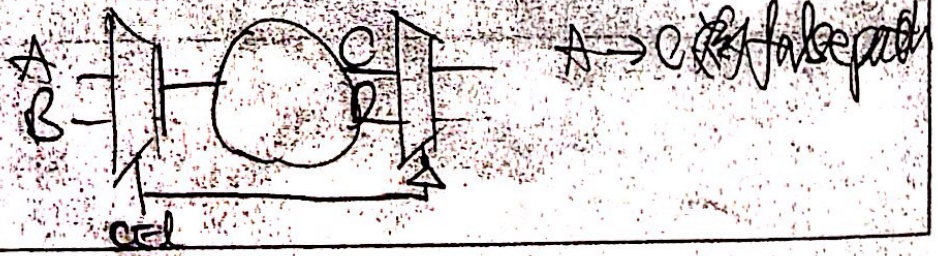
(a) 目前 BLOCK 要直時 前一級 input block 還未算完

ex)



(b) false path 即表示這不可能經過的路徑.

ex)



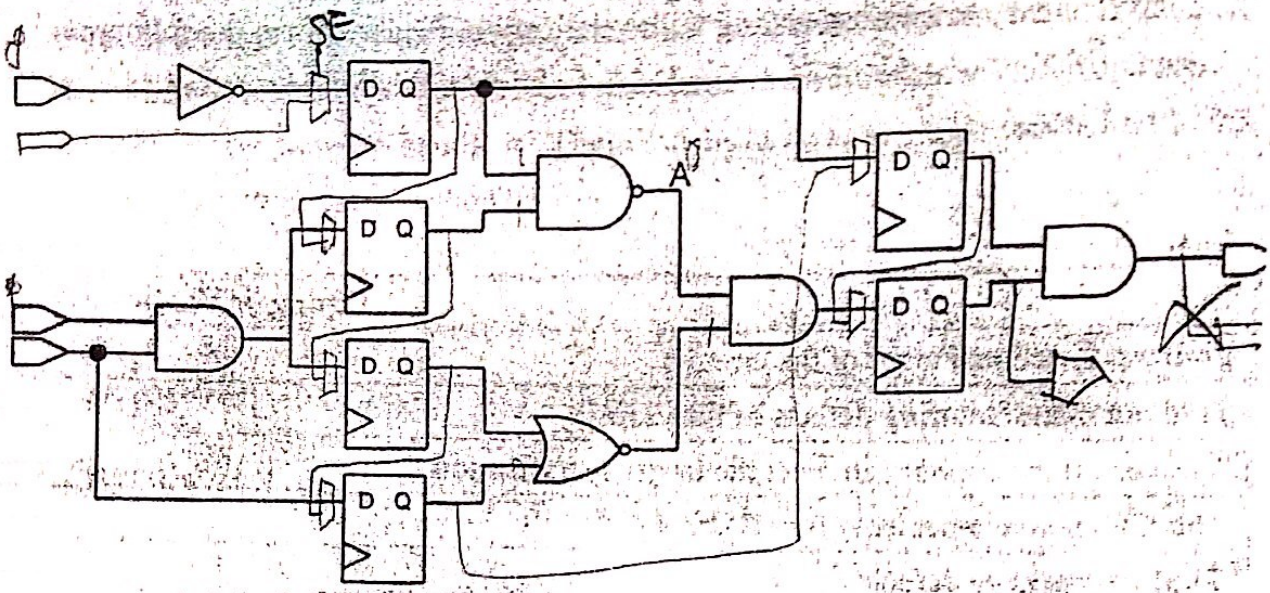
Problem 7 <Design for Testing>

A. (5%) It is easy to confuse testing with verification. Please fill out the following table to show whether the left-hand-side tasks should belong to testing or verification.

Ans.

Task	Testing or Verification
Find out functional error of the circuits	V
Find out manufacturing defects of the chip	T
To check if the design can meet the target specification	V
Should be done repeatedly for all chips	T
Find out package defects	T
Find out timing error of the circuits	V

B. (10%) For the following circuits, please draw the associated circuits with full-scan with multiplexed flip-flop. What input and output pins should be added?



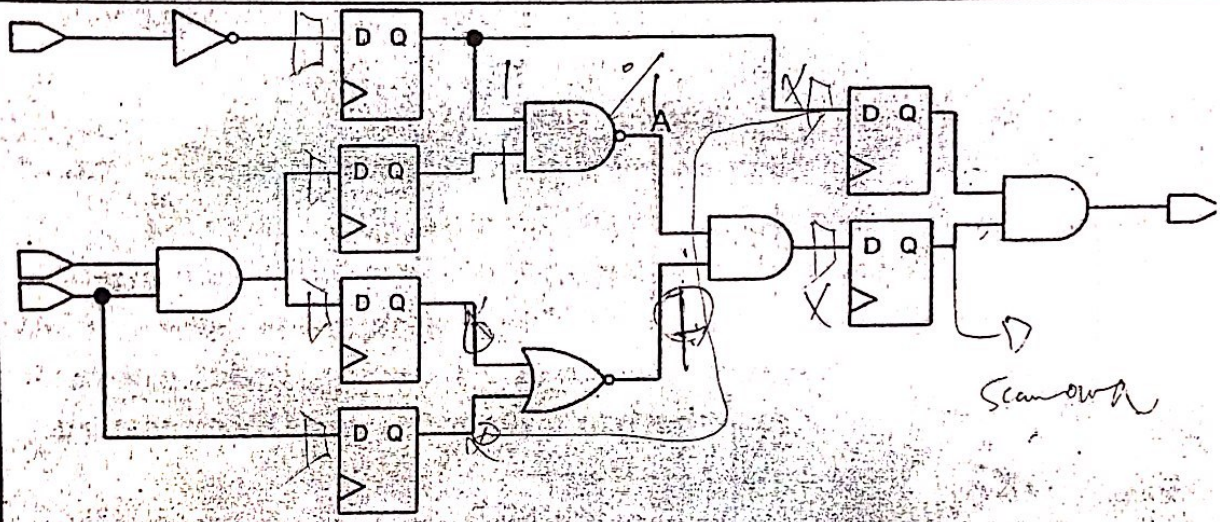
Ans.

The added input and output pins are:

C. (10%) Employ D-algorithm to find out the test pattern for the "stuck-at-1" fault at node A, include the test input vector to be shifted into the scan chain and the expected output to be shifted out from the scan chain.

How many clock cycles do we need to execute the above test pattern? Why?

Ans.



Input: XXXX 1 1

Output: 0/1 XXXXX

How many clock cycles do we need to test the above test pattern? Why?

Scan-in - 6 cycle

Capture - output - 1 cycle

Scan out - 6 cycle

7 cycle