

Problem 1 (10 points) <Basic Concepts>

A. Please explain the meaning of the following terminologies and where to use them:

1. (2 points) DesignWare Library
2. (2 points) Synchronous Reset

B. Please tell the meanings and the difference between each item:

1. (2 points) set_wire_load-mode TOP/Enclosed/Segment
2. (2 points) Blocking Assignment / Non-blocking Assignment
3. (2 points) Mealy Machine / Moore Machine

ANS:

combinational

sequential

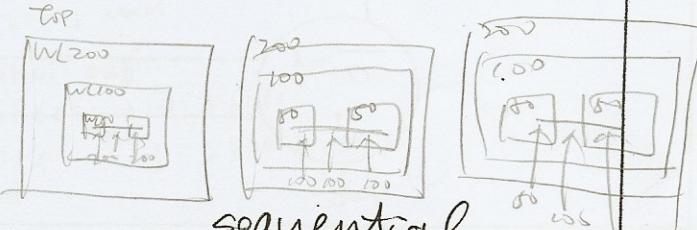
A. 1. DesignWare is technology-independent soft macros such as adders, comparators, etc., which can be synthesized into gates from your target library. If you want to use the Designware library, you must set the "synthetic library" and "search-path" in the .synopsys_dc.setup

A. 2. Use CLK as trigger signal. Use register for reset D

-2

B. 1. set_wire_load-mode TOP : Use -mode option, you can specify which wire load mode to use for nets that cross hierarchical boundaries

-2



sequential

B. 2. use non-blocking assignments within sequential always block. use blocking assignments within combinational always block

B. 3. Mealy Machine: Output is function of both input and current state.

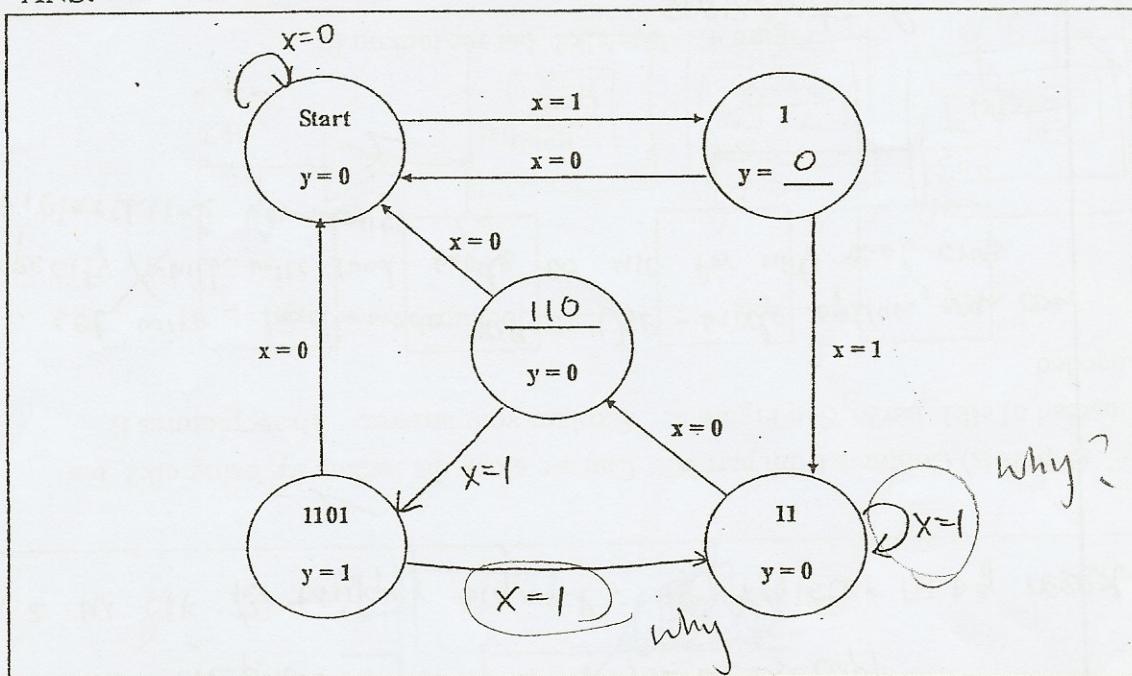
Moore Machine: Output is function of current state only not function of inputs.

Problem 2 (25 points) <Finite State Machine>

The following FSM is a Moore machine which detects if the input sequence x is 1101. The output y is '1' if this machine recognizes the input sequence 1101; otherwise it is '0'. There are five states in this FSM.

A. (10 points) Please complete the following state diagram. Add branches if needed.

ANS:



B. (15 points) Please complete the Verilog code with asynchronous negative edge reset that describes the above Moore machine.

ANS:

```

9 module seq1101(x, y, CLK, RESET);
  input x;
  input CLK;
  input RESET;
  output y;
  reg y;
  reg [2:0] y;
  parameter start = 3'b000, got1 = 3'b001, got11 = 3'b011,
            got110 = 3'b010, got1101 = 3'b110;
  reg [2:0] Q; // state variables

```

$\overline{X} \oplus \overline{C/L} = Y$

```
reg [2:0] D; // next state logic output
```

// next state logic

always @ (posedge CLK or negedge RESET)

begin

if (!RESET)

begin *Reset*

Q <= 3'b000;

~~y <= 0;~~

end

else

begin

Q <= D;

end end

(posedge CLK negedge RESET)

case

~~begin case (Q);~~

~~default:~~ ;

~~endcase~~

out put : *y* = *Q* & *x* > 7'b10;

寫反

-3

// state variables

always @ (Q or X)

begin

case (Q)

3'b000: begin

if (X == 1)

else

3'b001: begin

if (X == 1)

else

3'b011: begin

if (X == 0)

else

3'b110: begin

if (X == 1)

else

// output logic

always @ (Q)

y = Q[2];

if (*y* == 3'b110)

y = 1;

else *y* = 0;

if (RESET == 0)

begin

D = 3'b000;

end

D = 3'b011;

D = 3'b000;

end

D = 3'b010;

D = 3'b011;

end

3'b110: begin

if (X == 1) D = 3'b011;

else D = 3'b000;

end

-3

else

begin & end

start : 0

default?

end case?

Problem 1 (10 points) <Basic Concepts>

A. Please explain the meaning of the following terminologies and where to use them:

1. (2 points) DesignWare Library
2. (2 points) Synchronous Reset

B. Please tell the meanings and the difference between each item:

1. (2 points) set_wire_load-mode TOP/Enclosed/Segment
2. (2 points) Blocking Assignment / Non-blocking Assignment
3. (2 points) Mealy Machine / Moore Machine

ANS:

combinational

sequential

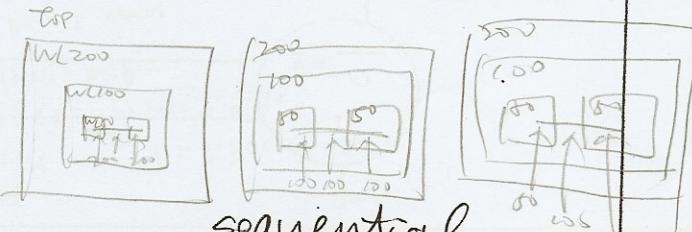
A. 1. DesignWare is technology-independent soft macros such as adders, comparators, etc., which can be synthesized into gates from your target library. If you want to use the Designware library, you must set the "synthetic library" and "search-path" in the .synopsys_dc.setup

A. 2. Use CLK as trigger signal. Use register for reset D

-2

B. 1. set_wire_load-mode TOP : Use -mode option, you can specify which wire load mode to use for nets that cross hierarchical boundaries

-2



sequential

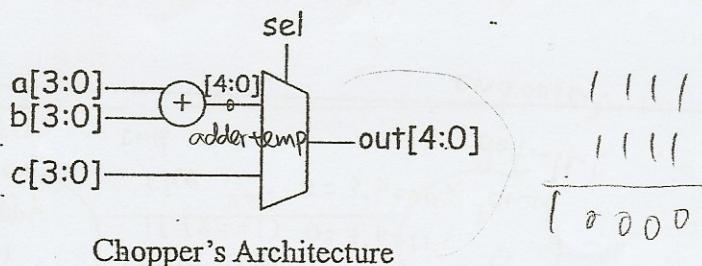
B. 2. use non-blocking assignments within sequential always block. use blocking assignments within combinational always block

B. 3. Mealy Machine: Output is function of both input and current state.

Moore Machine: Output is function of current state only not function of inputs.

Problem 3 (15 points) <Verilog Debug>

Chopper wants to be the greatest IC designer in the world. One day he found there are some errors in his code that does not match with his architecture. Now it's your turn to save Chopper from this painful situation. Please write your corrections to his code in the right column. Totally, how many bugs are there?



13

ANS:

Original Code

```

module Chopper_Code(out, a, b, c, sel);
output [4:0] out;
input [3:0] a, b;


reg [4:0] out;

always @((a|b)) b;
addertemp = a+b;

always@(sel) begin
    or addertemp or c
    case(sel)
        1'b0: out=addertemp;
        1'b1: out={b, c};
    endcase
end
endmodule

```

[[3]]

Corrected Code

```

module Chopper_Code (out, a, b, c, sel);
output [4:0] out;
input [3:0] a;



input sel;
reg [4:0] out;

always @ (sel or a or b or c)
begin
    case (sel)
        1'b0: out = addertemp;
        1'b1: out = {1'b0, c};
    endcase
end

```

-2

Problem 4 (15 points) <Synthesis>

Johnson is in charge of integrating the chip in Figure 1. In this chip, two blocks (FUNC_A and FUNC_B) are designed by Johnson himself and the other one block (CUSTOM_IP) is purchased from some other company. Johnson's mission is to integrate these three blocks.

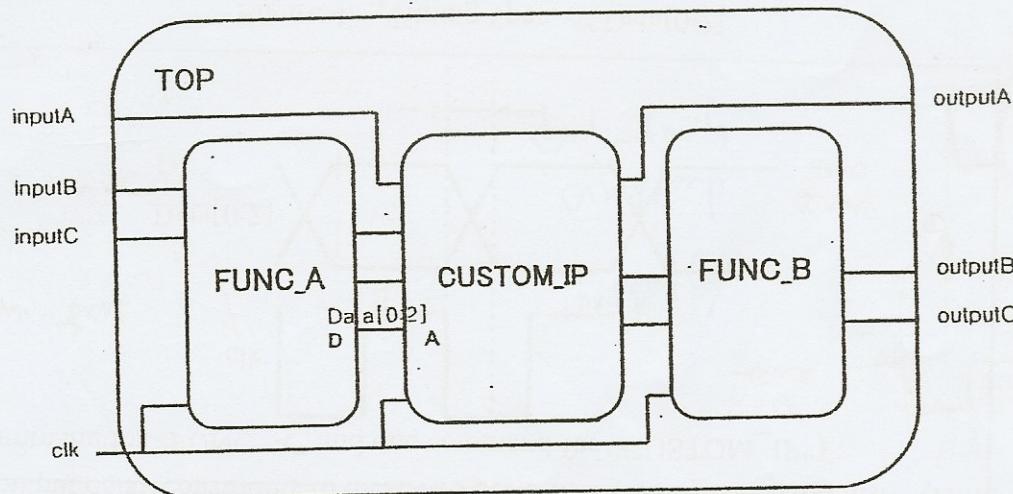


Figure 1

A. (5points) If we use PIC2A as the input pad and POC2A as the output pad as described in Figure 2, Figure 3 and Table 1, please write the commands for the port constraints of TOP related to pad explicitly including the exact numbers. (Hint: 2 commands for input and 2 commands for output, please use set_drive, set_load, set_input_delay, and set_output_delay commands)

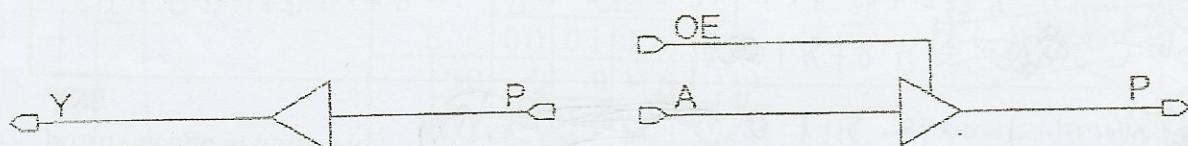


Figure 2. PIC2A

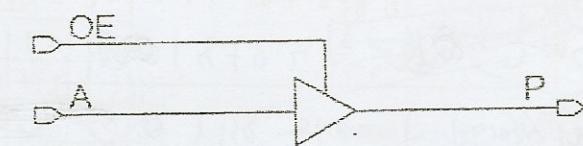


Figure 3. POC2A

Table 1 Data Sheet for Pad

Cell Name	Description	Intrinsic Delay (ns)	Kload (ns/pF)	Capacitance (pF)
PIC2A	P → Y ↑	0.714	0.108	port P 2.902
	P → Y ↓	0.543	0.106	
POC2A	A → P ↑	1.63	0.172	port A 0.038
	A → P ↓	1.839	0.193	
	OE → P ↑	1.631	0.172	port OE 0.109
	OE → P ↓	1.88	0.193	

ANS:

set_input_delay - clock clk - max 0.714	Y	port
set_input_delay - clock clk - min 0.543	Y	
set_output_delay - clock clk - max 1.839	A	
set_output_delay - clock clk - min 1.63	A	
set_output_delay - clock clk - max 1.88	OE	
set_output_delay - clock clk - min 1.63	OE	
set_output_delay - clock clk - min 1.63	OE	

- 3.5

B. (5points) In design FUNC_A, the clock period = 35ns with rising edge at 0ns, rise time network latency = 3ns, fall time network latency = 1ns, clock rise time skew = 1.8ns, and clock fall time skew = 1.0ns. In this design, positive edge triggered flip-flops of library hold time = 0.2ns and setup time = 0.3ns is used. Please derive the data required time for hold time and data arrive time for setup time at clock positive edge at 35ns with.

ANS:

~~RT = Dclk_p + Dclk_s + Dclk_h - Dclk_n - T_h~~ ~~+ rise skew + library hold time~~

$$= 1.8 + 0.2 = 2 \text{ ns}$$

$$RT : Dclk_p + Dclk_s + Dclk_h - Dclk_n - T_h = 35 + 3 + 1 - 0.3 = 38.7$$

$$RT : Dclk_s + Dclk_h + T_h = 3 + 1 + 0.2 = 4.2$$

$$\text{Setup} = \text{clock period} - \text{rise skew} - \text{setup time} = 35 - 1.8 - 0.3$$

C. (5points) Suppose we have a data "Data[0:2]" transmitted from port "D" of "FUNC_A" to port "A" of "CUSTOM_IP". The data bus of "Data[0:2]" will be stable after 2ns after every positive edge of clock "clk". Please write down two input/output delay constraints to these two ports with clock period of 35ns. (Hint: one command for "FUNC_A" and one command for "CUSTOM_IP")

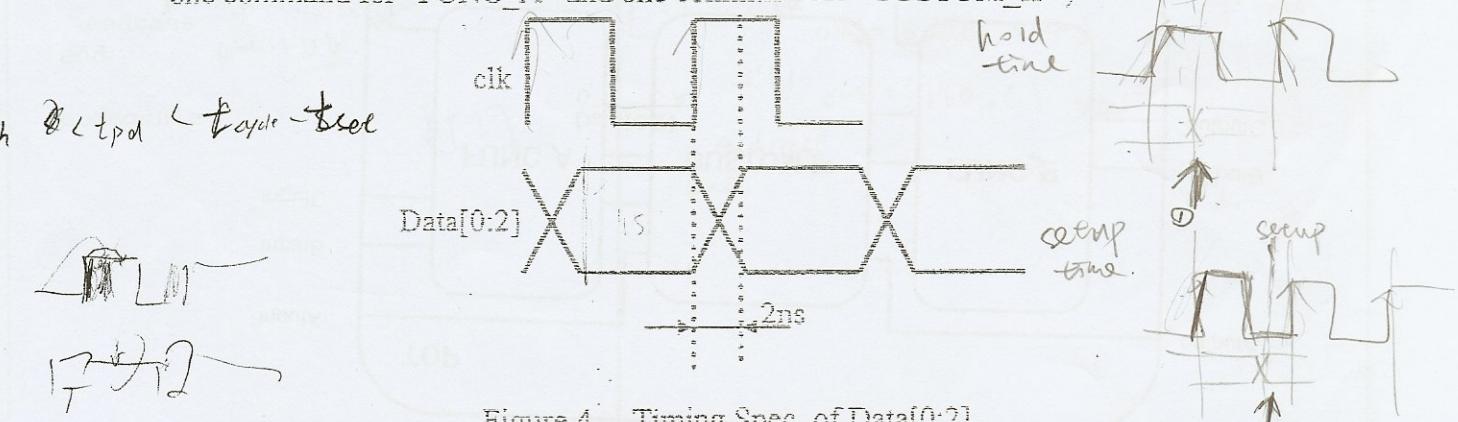


Figure 4: Timing Spec. of Data[0:2]

ANS:

- 5

Problem 5 (15 points) <Synthesis> Ppt. p.203

A. (5 points) While compiling the design, we found a warning message "multiple design instance". Please describe the reason why this happens. In addition, please describe the three recommended solutions to fix this problem and the differences of these three methods.

ANS:

~~Joint-touch
un group
uniquely~~ → ~~Joint-touch
un group
uniquely~~

B. (5points) If we have a block with the following function. Please write down one Boolean expression using structure method and one Boolean expression using the flatten method. (Hint: use 4 terms in flatten method and only one common factor $XOR(C,D)$ in structure method)

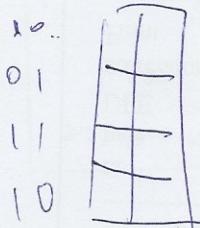


Table 2. Function Table

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

ANS:

$$E = BC'D' + ABC'C + AB'D + AB'C' + A'CD'$$

~~Structure~~ ~~E = BC'D' + ABC'C + AB'D + AB'C' + A'CD'~~

~~Flattening~~ ~~E = BC'D' + ABC'C + AB'D + AB'C' + A'CD'~~

$E = (B \oplus C) \cdot D' + (A \oplus C) \cdot D$

~~Structure~~ ~~$E = BC'D' + ABC'C + AB'D + AB'C' + A'CD'$~~

~~Flattening~~ ~~$E = BC'D' + ABC'C + AB'D + AB'C' + A'CD'$~~

Final: $E = BC'D' + ACD + B'CD + A'CD'$

$XOR(C,D) = (0,0) \quad (0,1) \quad (1,0) \quad (1,1)$

C. (5points) After reporting the violations, we found a violation described as follows.
 Please indicate what type of violation this is and how to fix this problem by using the
 "compile command".

Report : constraint -all_violators
 -verbose

Design : rx

Version: 2002.05-SP1

Date : Thu Aug 28 11:20:39 2003

Startpoint: reset (input port)

Endpoint: rx_gain_ctrl/state_reg[0]
 (rising edge-triggered flip-flop clocked by clk_1M)

Path Group: clk_1M

Path Type: min

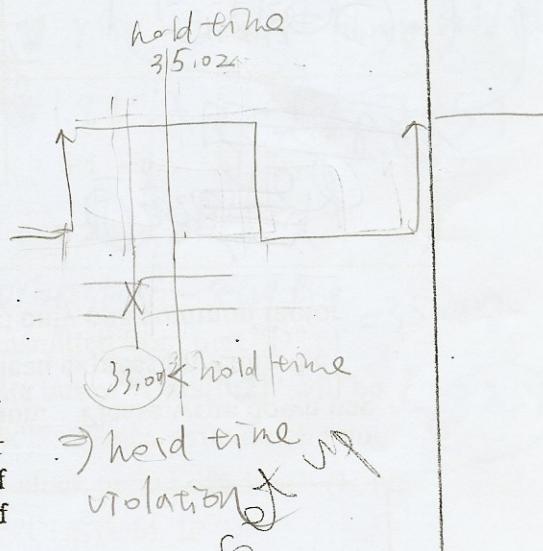
hold time check.

MAP

Des/Clust/Port Wire Load Model Library

Point	Incr	Path
rx	TSMC25_Conservative	typical
gain	TSMC25_Conservative	typical
DEE	TSMC25_Conservative	typical
correlator_0	TSMC25_Conservative	typical

clock (input port clock) (rise edge)	32.00	32.00
clock network delay (ideal)	0.00	32.00
input external delay	0.00	32.00 r
reset (in)	0.00	32.00 r
U9/Y (CLKBUFX1)	0.47	32.47 r
rx_gain_ctrl/U282/Y (NOR2XL)	0.17	32.64 f
rx_gain_ctrl/U255/Y (NAND2X1)	0.23	32.87 r
rx_gain_ctrl/U226/Y (OAI21XL)	0.12	33.00 f
rx_gain_ctrl/state_reg[0]/D (DFFXL)	0.00	33.00 f
data arrival time		33.00
clock clk_1M (rise edge)	35.00	35.00
clock network delay (ideal)	0.00	35.00
rx_gain_ctrl/state_reg[0]/CK (DFFXL)	0.00	35.00 r
library min-time <i>hold</i>	0.02	35.02
data required time		35.02
data required time		35.02
data arrival time		-33.00
slack (VIOLATED)		-2.02



-2.02 32 clk ?

ANS:

hold time violated

Compile -exact -map ✓ *-only_hold_time*

Problem 6 (20 points) <Timing Analysis and DfT>

A. (5 points) Figure 1 is a picture of a scan chain of two clock domains. F1 and F2 belong to the clk1 domain. F3 belongs to the clk2 domain. We insert a lockup FF between F2 and F3. Please explain why we want to insert a lockup FF in our scan chain. (assume all FF are positive edge triggered) Draw pictures if needed.

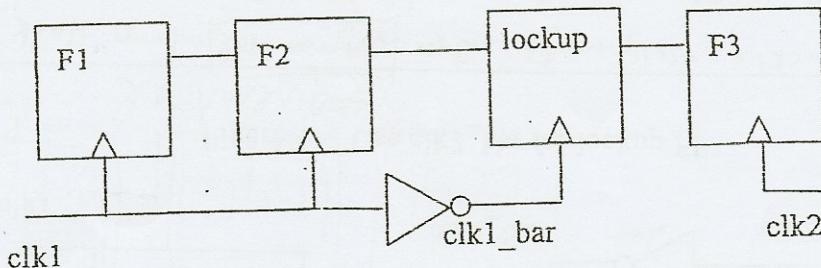
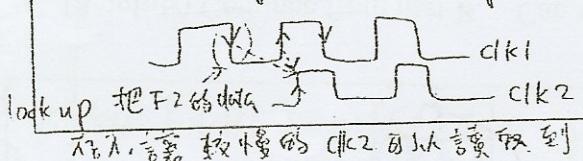


Figure 1. Lockup FF in a Scan Chain.

ANS:

if clk2 rising edge is slower than clk1's rising edge, then the scan data will be lost in the chain. therefore we need this lockup ff to keep the chain works correctly.



B. (10 points) Continue from part A. Suppose all FFs are identical. The timing of FF is shown in Figure 2. The setup time and hold time are given by $t_s (>0)$ and $t_h (>0)$ respectively. In addition, the time interval from positive clock edge to data output is t_Q , and the clk1 inverter has a delay of t_{INV} .

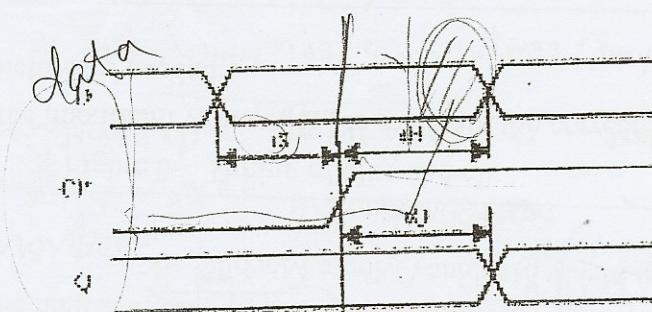


Figure 2. Definition of FF Timing

Figure 3 shows the timing of clk1 and clk2 assuming that the cycle time of both clocks are t_{cycle} and their duty cycles are both 50%. As specified in the figure, clk 2 is identical to clk 1 except that clk2 is t_{late} later than clk1 under the following condition: $0 < t_{late} < 50\% t_{cycle}$.

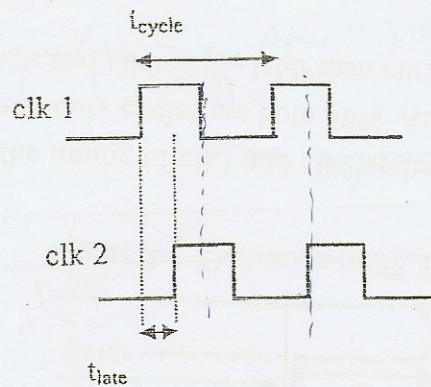
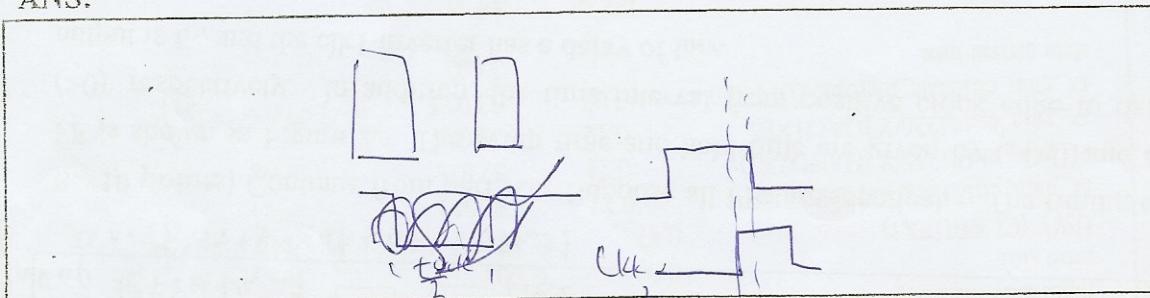


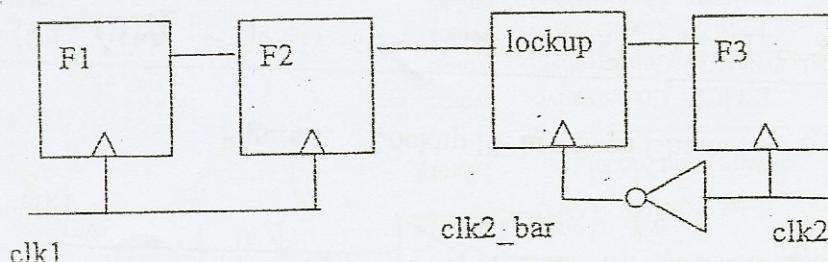
Figure 3 Timing of clk1 and clk2

Please find out the maximum value of t_{late} to maintain the correct function of the scan chain. Draw pictures if needed.

ANS:



C. (5 points) Continue from part B. Can we clock the lockup FF using clk2_bar instead of clk1_bar? See Figure 4. Explain your answer. Draw pictures if needed.

Figure 4. Use clk2_bar for lockup FF

ANS:

