

Computer-Aided VLSI System Design
Midterm Examination
2014. 11. 19

Name

林柏宏

Student ID

R03943099

Instructions

This is a **CLOSED** book exam. The exam is to be completed in 90 minutes. 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 18 pages of the exam.

Score Board (to be filled by TAs)

	Points	Score		Points	Score
Problem 1	20	20	Problem 5	20	20
Problem 2	10	8	Problem 6	10	9
Problem 3	10	9	Problem 7	20	19
Problem 4	10	10	Total	100	95

1. <General Concept> (20%)

A. (8 points) <Synchronous and Asynchronous Reset>

There are two kinds of reset strategies for clock-triggered registers: synchronous and asynchronous.

(a) (2 points) Explain what differences between them.

(b) (6 points) Write a template code for them below.

ANS

(a)

synchronous reset: It will reset the flip-flop only on the active edge of clock.

asynchronous reset: reset the flip-flop asynchronously if, no matter what signal the clock is.

(b)

i. synchronous reset 需等 clock event 以重置, asynchronous

(i) Template code for a register with Active Low Asynchronous triggered reset:

```
module Register_a(data_out,data_in,reset_b,clk);  
output [7:0] data_out;
```

```
input [7:0] data_in;  
input reset_b,clk;  
reg [7:0] data_out;
```

//----complete below

```
always @ (posedge clk or negedge reset-b) begin
```

```
if (!reset_b) begin
```

```
    data_out <= 0;
```

```
end
```

```
else begin
```

```
    data_out <= data_in;
```

```
end
```

```
end  
endmodule
```

(ii) Template code for a register with **Active Low Synchronous** reset:

```
module Register_s(data_out,data_in,reset_b,clk);  
output [7:0] data_out;
```

```
input [7:0] data_in;  
input reset_b,clk;
```

```
reg [7:0] data_out;
```

```
//----complete below
```

```
always @ (posedge clk) begin  
    if (!reset_b) begin  
        data_out <= 0;  
    end  
    else begin  
        data_out <= data_in;  
    end  
end  
endmodule
```

B. (4 points) <Design Flow>

Please draw a design flow chart with correct connection between each step

(1) Placement (2) Logic Synthesis (3) Routing (4) Floorpanning (5) HDL Coding (6) DFT Insertion

ANS

(5) HDL Coding → (2) Logic Synthesis → (6) DFT insertion
→ (4) Floorpanning → (1) Placement → (3) Routing

C. (8 points) < Basics - True or False >

Answer T for true, F for false. **Give brief explanation** if the answer is false.

- (a) F Verilog is a hardware description language only for RT-level, gate-level, and transistor level modeling and simulation. (2 points)

It also supports behavior-level.

- (b) F Task always return a single value, it cannot have output or inout arguments. It is similar to statement, which can only be used to describe combinational circuits. (2 points)

Task can return not only a single value.

- (c) F Many Verilog HDL simulators, such as Cadence NCVerilog and Synopsys VCS, are cycle-based simulators. (2 points)

event-based

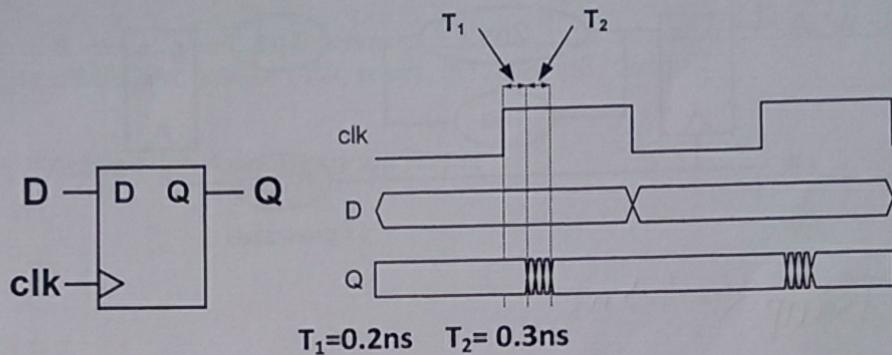
- (d) T The \$display system task displays the values of the listed arguments at the end of any time unit in which any of the arguments change value (2 points)

display 只在執行到 display 時才顯示,

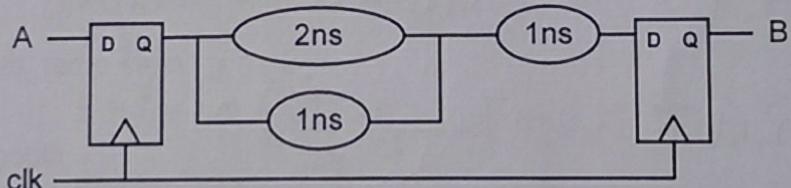
Monitor 才有上述功能,

2. <Important Timing Parameters> (10%)

Assume that the timing characteristics of the two flip-flops in the circuit are the same. Their timing diagrams can be described as follows:



If the circuit in the below operates at the clock frequency of **200MHz** without any timing violation.



(a) (5 points) Write the timing inequality for setup time and hold time.

① For setup time.

$$T - (T_{cq} + T_{logic}) > T_{setup}$$

$$T = \frac{1}{200\text{MHz}} = \frac{1}{200 \times 10^6} = \frac{1 \times 10^{-9}}{200} = \frac{1000}{200} \text{ ns} = 5 \text{ ns}$$

$$T_{cq} = T_1 + T_2 = 0.2 + 0.3 = 0.5 \text{ ns}, \quad T_{logic} = 2 + 1 = 3 \text{ ns}$$

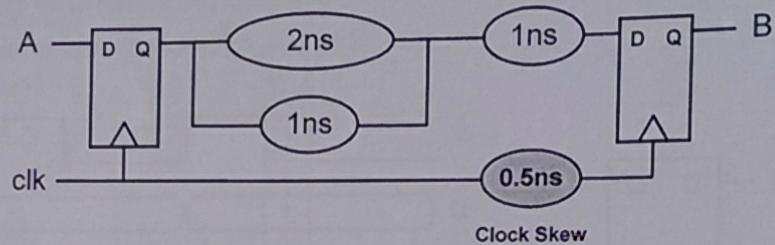
$$\therefore T_{setup} < 5 - (0.5 + 3) = 1.5 \text{ ns}.$$

② For hold time.

$$T_{hold} < T_{cq,cd} + T_{logic,cd}, \quad T_{logic,cd} = 1 + 1 = 2 \text{ ns}, \quad T_{cq,cd} = T_1 = 0.2 \text{ ns}$$

$$\therefore T_{hold} < 2 + 0.2 = 2.2 \text{ ns}$$

(b) (5 points) If there is clock skew in this circuit, as shown in bellow. Please write the timing inequality for setup time and hold time at the clock frequency of **200MHz** without any timing violation.



$$T_{\text{setup}} < 1.5 \text{ ns}$$

~~$$T_{\text{hold}} < 2.2 \text{ ns}$$~~

~~-2~~ clock skew

$$\rightarrow T_{\text{setup}} < 1.5 - T_{\text{skew}} = 1 \text{ ns}$$

$$T_{\text{hold}} < 2.2 - T_{\text{skew}} = 1.7 \text{ ns}.$$

3. <Verilog HDL -Code Debugging> (10%)

(10pts) Identify syntax errors, correct them and explain: 1pt for each. Identify inappropriate code (or semantics errors), correct them and explain: 1pt for each. The number of total errors may be more than 10.

① 命名不能用數字開頭
module 3inputMACProcessor (clk, reset, in1, in2, in3, out);

② 缺少引號

```
input reset; // Active High Asynchronous Reset  
input [3:0] in1;  
input [3:0] in2;  
input [7:0] in3;  
output [8:0] out;
```

input clk.

```
reg [3:0] in1;  
reg [3:0] in2;  
reg [7:0] in3;
```

```
reg [7:0] product_temp;
```

③ 連續賦值語句 continuous assign, 宣告成 wire,

```
assign product_temp <= in1_r * in2_r;
```

⑨ <=> → ④ 需在上面宣告 reg [8:0], out;

```
always@(*) begin
```

```
    out <= product_temp + in3_r;
```

```
end
```

/* Sequential Logic */ ⑤ comment /* */

```
always@(posedge clk and posedge reset)
```

```
begin
```

① reset, ② active high
if (!reset) ③ begin ④ or
 in1_r <= 3'd0; ⑤ asynchronous reset,
 in2_r <= 3'd0;
 in3_r <= 7'd0;

else

in1_r <= in1;
 in2_r <= in2;
 in3_r <= in3;

```
end
```

endmodule

⑩ alt endmodule

⑥ 不需分號 ;

⑦ 上面需宣告 reg [3:0] in1_r

可以把宣告的 in1, in2, in3

改為 in1_r, in2_r, in3_r

input 與 wire 即可。

(沒特別宣告, input 即是 wire)

4. <Verilog HDL –FSM and Simulation> (10%)

(10pts) Given below is a Finite-State-Machine (FSM). We have put this module in our testbench as a Design-Under-Test (DUT). After the simulation, the command window has printed response from monitor. Please Finish the output results below based on this FSM and given information.

```
module FSM  (clk, rst, in, out_r);

parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

input  clk, rst, in;
output [1:0] out_r;

reg [1:0] out_r, out;
reg [1:0] current_state, next_state;

// Next State Logic
always@(*) begin
    case(current_state)
        S0: next_state = (in == 1'b0)? S0 : S2;
        S1: next_state = (in == 1'b0)? S0 : S2;
        S2: next_state = (in == 1'b0)? S1 : S3;
        S3: next_state = (in == 1'b0)? S1 : S3;
        default: next_state = 2'b00;
    endcase
end

// Current State Memory & Output Register
always@(posedge clk or posedge rst)
begin
    if(rst) begin
        current_state <= 0; out_r <= 0;
    end
    else begin
        current_state <= next_state;  out_r <= out;
    end
end
```

```

// Output Logic
always@(*) begin
    out[1] = in ^ current_state[0];
    out[0] = in ^ current_state[0] ^ current_state[1];
end

endmodule

```

'timescale 1ns/1ns

module testbench;

reg clk, rst;

reg in;

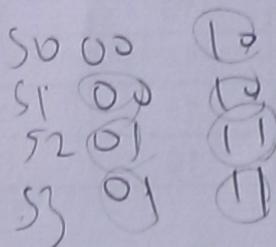
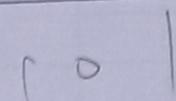
wire [1:0] out;

FSM DUT(.clk(clk), .rst(rst), .in(in), .out_r(out));

// APPLY STIMULUS

\$monitor("%t %b %b %b %b", \$time, clk, rst, in, out);

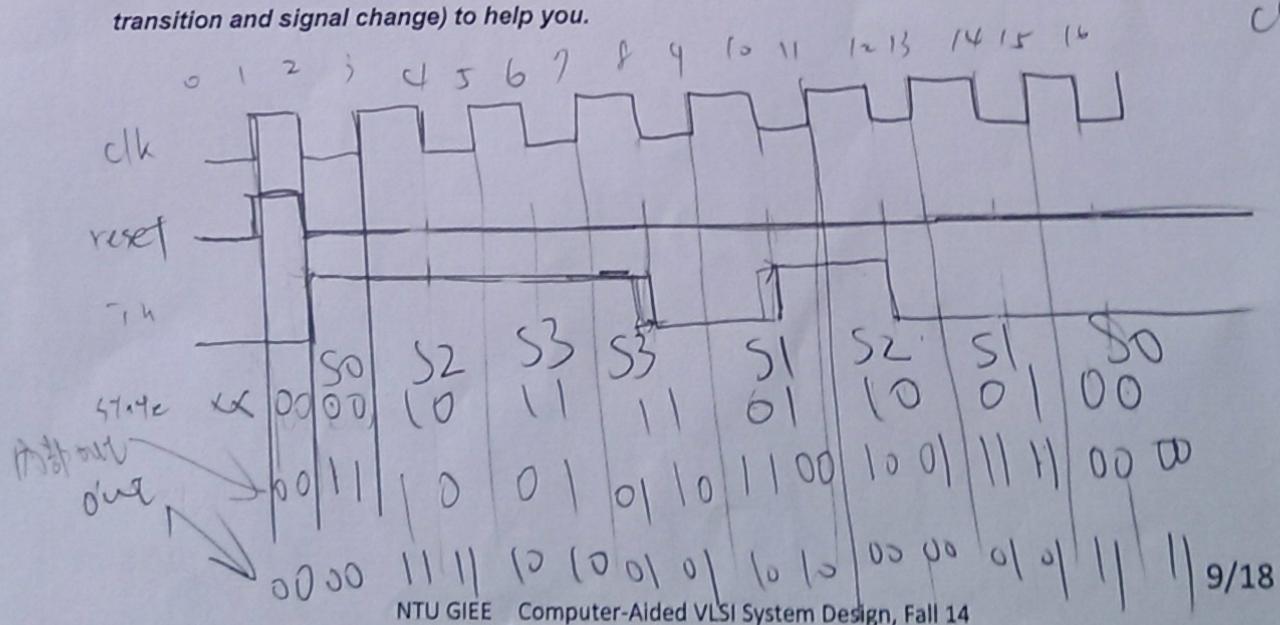
endmodule



$$\text{out}[1] = \text{in} \oplus \text{current_state}[0]$$

$$\text{out}[0] = \text{in} \oplus \text{current_state}[0] \oplus \text{current_state}[1]$$

Hint: Draw a timing diagram and state transition graph for capturing critical events (state transition and signal change) to help you.



Monitor Output Response:

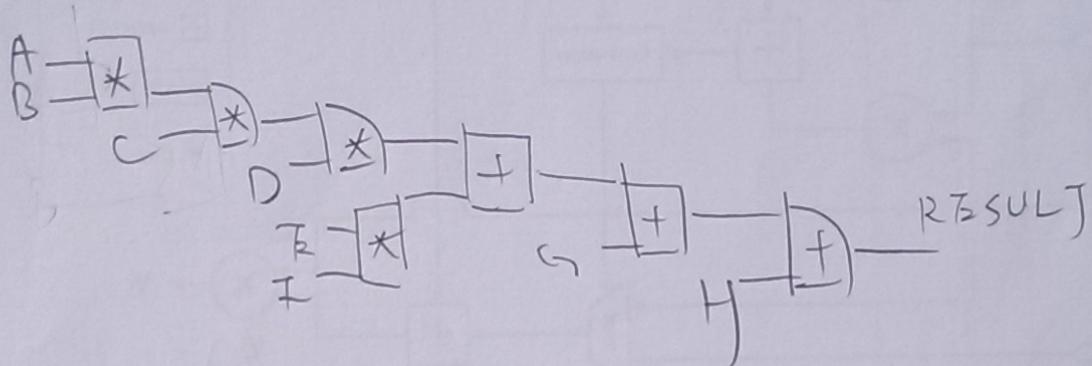
Time	clk	reset	in	out
0	0	0	0	xx
1	1	1	0	00
2	0	0	1	00
3	1	0	1	<u>10</u>
4	0	0	1	<u>11</u>
5	1	0	1	<u>10</u>
6	0	0	1	<u>10</u>
7	1	0	1	<u>01</u>
8	0	0	0	<u>01</u>
9	1	0	0	<u>10</u>
10	0	0	1	<u>10</u>
11	1	0	1	<u>00</u>
12	0	0	0	<u>00</u>
13	1	0	0	<u>01</u>
14	0	0	0	<u>01</u>
15	1	0	0	<u>11</u>
16	0	0	0	<u>11</u>

5. <Verilog HDL -Coding for Synthesis Optimization> (20%)

A. (10pts) Suppose the delay time of a two-input adder is $2T$, and the delay time of a two-input multiplier is $3T$. When we use only the two-input adders and two-input multipliers to obtain the following result:

$$\text{RESULT} = A * B * C * D + E * F + G + H$$

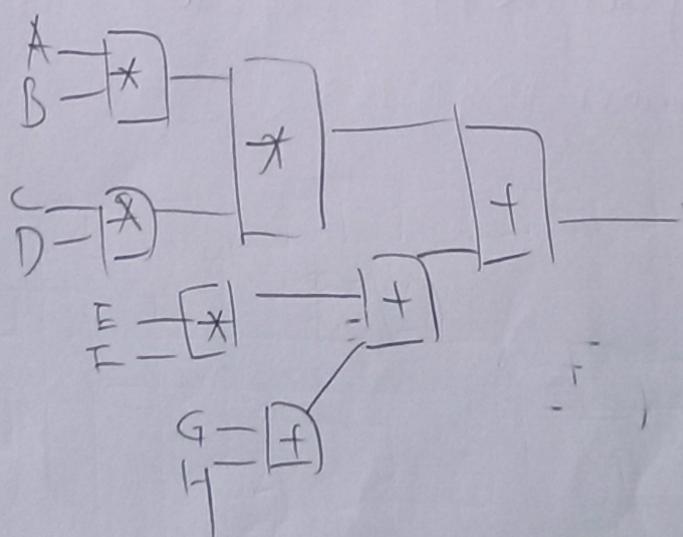
(a) (5 points) What is the critical path delay of the direct implementation?



$$\therefore \text{critical path delay} = (3+3+3+2+2+2)T = 15T$$

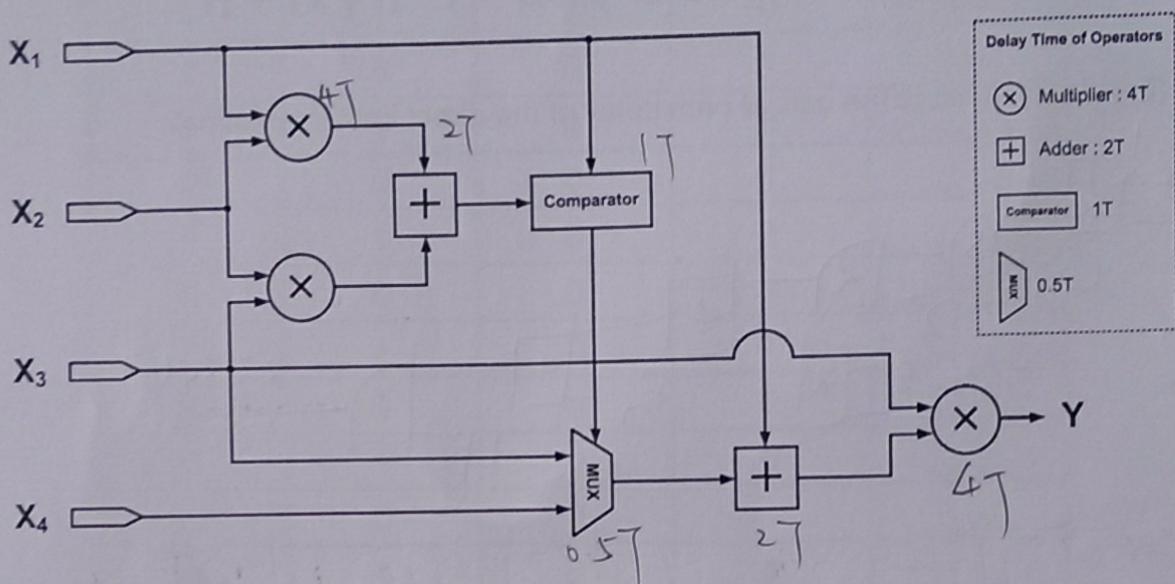
(b) (5 points) Is it possible to reduce the path delay to $8T$ by modifying the coding style? If it's possible, please write down the modified function.

$$\text{RESULT} = \left[(A * B) * (C * D) \right] + \left[(E * F) + (G + H) \right]$$



$$\text{critical path delay} = 8T$$

B. (10pts) An arithmetic combinational circuit with 4 input ports and 1 output ports is shown below. Suppose the delay time of multipliers, adders, comparators and multiplexers are $4T$, $2T$, $1T$ and $0.5T$, respectively.



(a) (2 points) Please analyze this circuit. What is the critical path delay?

$$\begin{aligned} \text{critical path delay} &= 4T + 2T + 1T + 0.5T + 2T + 4T \\ &= 13.5T \end{aligned}$$

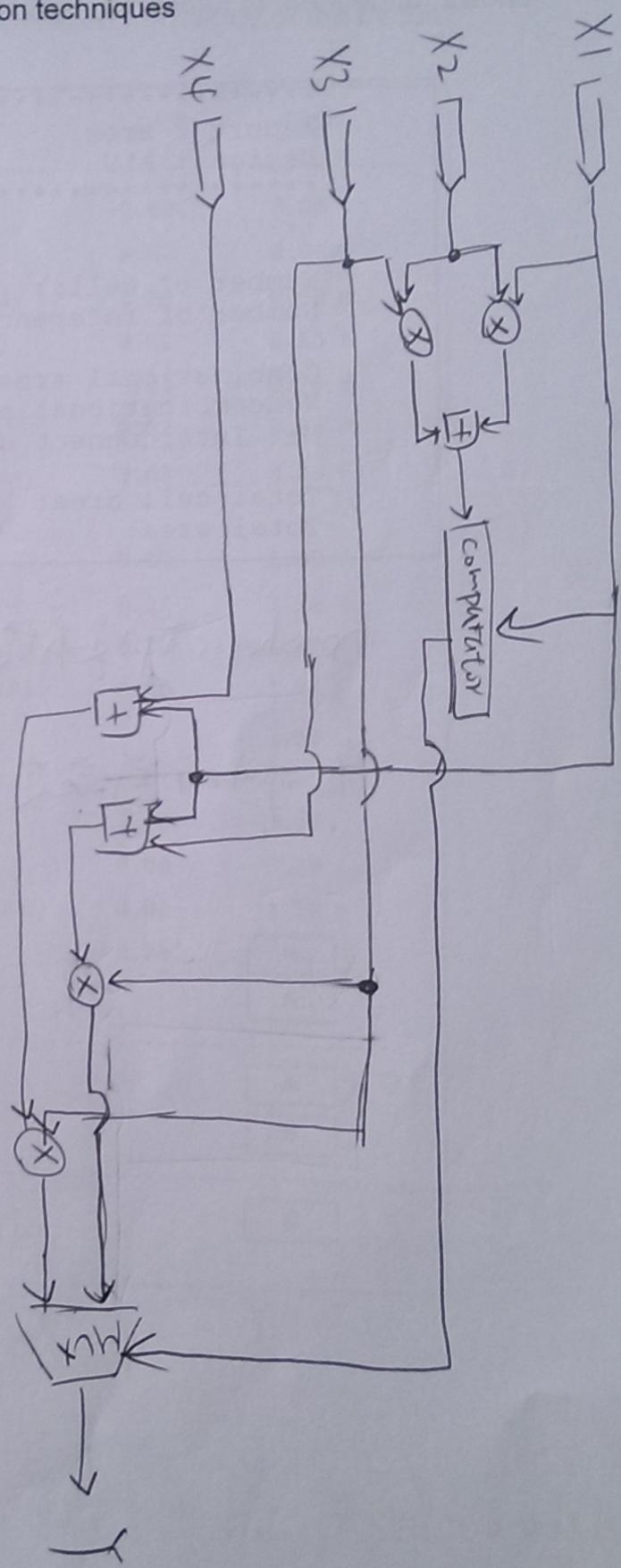
(MUX 需要等 0.5T, sel 为 1 时 \otimes , sel 为 0 时 $+$ ，但因，
(1 由 comparator.)

(b) (8 points) We can try to use the "data-path duplication" to reduce the critical path with sacrificing some area overhead. Please draw the corresponding circuits and show the critical path delay after data-path duplication techniques

$$\therefore \text{critical path delay} = [T + 0.5T] = 1.5T$$

(for 4 of Mux)

✓



6. <Synthesis Report> (10%)

A. (5pts) The following figure is the area report after synthesis. It sometimes shows "undefined" in total area. Please explain it and describe how to fix it.

```
*****
Report : area
Design : ALU
*****
...
Number of cells: 90
Number of references: 10
Combinational area: 1939.291626
Noncombinational area: 2049.062256
Net Interconnect area: undefined
Total cell area: 3988.353760
Total area: undefined
```

synthesis 前沒有設定好 wire model, ✓

在 synthesis 前設定 wire model 即可解決

B. (5pts) In the timing report, the most important information are the data required time, data arrival time and slack value. Please fill in these values according to this timing report. And what is the maximum operation frequency of the circuit?

Point	Incr	Path
clock CLK (rise edge)	0.00	0.00
clock network delay (ideal)	0.00	0.00
memory_map_reg[1][0]/CK (SDFFRX4)	0.00	0.00 r
memory_map_reg[1][0]/Q (SDFFRX4)	0.55	0.55 f
U1924/Y (OR2X8)	0.21	0.76 f
U1807/Y (CLKINVX12)	0.07	0.83 r
U1493/Y (INVX16)	0.05	0.89 f
U3316/Y (NAND2X8)	0.19	1.08 f
U3315/Y (OR2X8)	0.22	1.30 f
U2504/Y (NAND2X8)	0.20	1.50 r
current_state_reg[0]/D (SDFFRHQX8)	0.00	1.50 r
data arrival time		1.50
clock CLK (rise edge)	3.20	3.20
clock network delay (ideal)	0.00	3.20
current_state_reg[0]/CK (SDFFRHQX8)	0.00	3.20 r
library setup time	-0.20	A 3.0
data required time		A
data required time		A 3.0
data arrival time		B 1.5
slack <input type="checkbox"/> D	Met or Violated? Met.	C 1.5

$$A = 3.0 \quad B = 1.5, \quad C = 1.5$$

$$\text{Maximum frequency} = \frac{1}{1.5 \text{ ns}} = \frac{10^9}{1.5} = \frac{10}{15} \times 10^9 = 6.67 \times 10^8$$

↓
(以單位 ns 計算)

$$= 667 \text{ MHz}$$

單位都以 ns 計

A) Data required time?

3.0 ns



B) Data arrival time?

1.5 ns



C) Slack?

1.5 ns



D) Met or violated?

Met



Maximum operation frequency of the circuit?

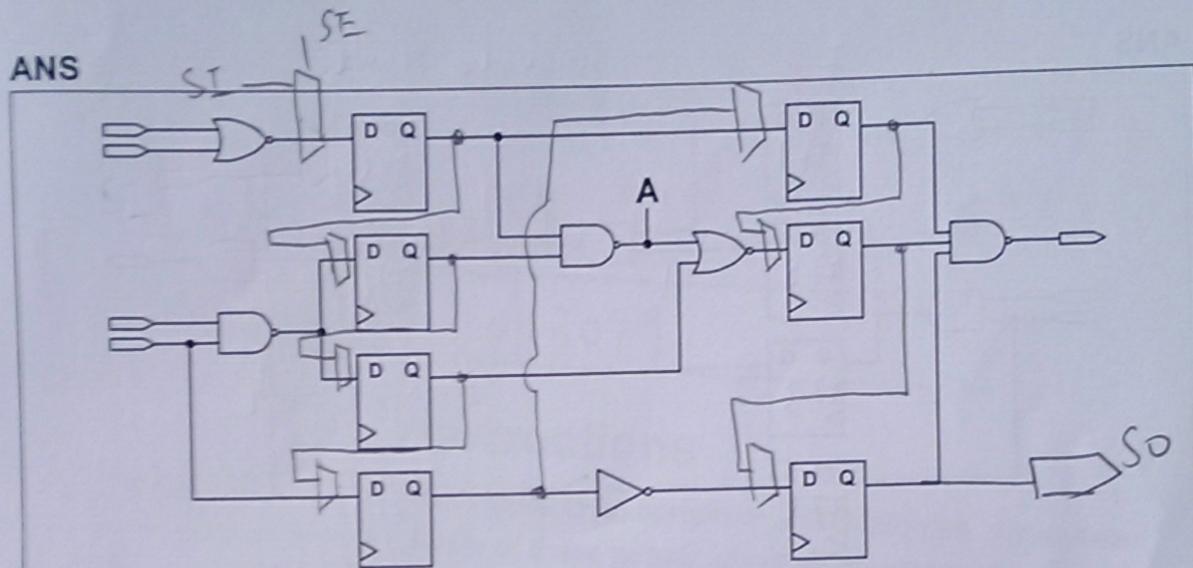
$$\frac{1}{1.5 \text{ ns}} = 667 \text{ MHz}$$



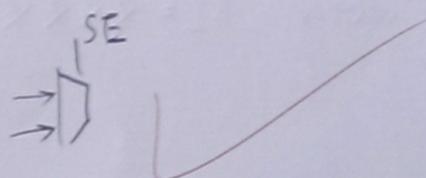
(-1)

7. <Design for Testability> (20%)

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



每個 Mux 的 sel 都是 SE



The added input and output pins are:

input pins : SI, SE

output pin : SO

DD 1 ~ 20.

B. (10pts) 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.

