1. Code Debugging and Simulation (10pts)

-3 **A. (6pts)** Identify syntax and semantic errors. Correct them and put annotations. Miss one error or mistake one error will minus 1 point until 0.

```
module 2value_multiplier (
    clk,    // Clock
    rst_n,  // Asynchronous reset active low
    valueA,
    valueB,
    valueSum,
;

input clk;
input rst_n;
input [3:0] valueA;
input [3:0] valueB;
output [7:0] valueSum;


reg [3:0] valueA_r, valueB_r;
reg [7:0] valueSum_r;
reg [7:0] valueA_sgnExt, valueB_sgnExt;
reg [7:0] multiply_result;

assign valueA_sgnExt ={{4{valueA_r[3]}}, valueA_r};
assign valueB_sgnExt ={{4{valueB_r[3]}}, valueB_r};
assign multiply_result = valueA_sgnExt * valueB_sgnExt;
assign valueSum = valueSum_r;
```

special character 開頭, 把 -① 拿掉改成 value_multiplier.

多一個 "," 逗號, -②

不用加

少一個 ; -③

應該改成 wire [7:0] -④

signed extension 少 {} -⑤

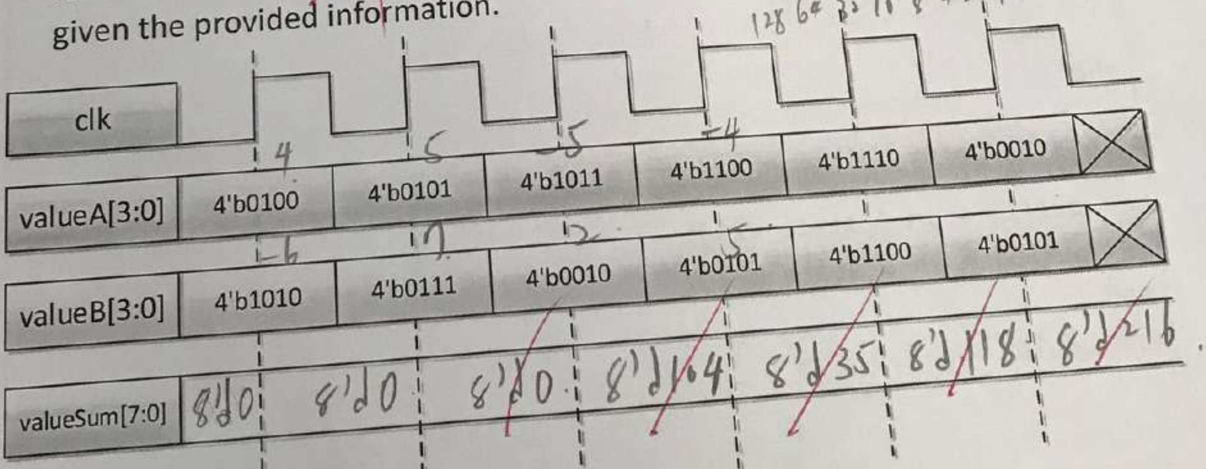
```
always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        valueA_r <= 0;
        valueB_r <= 0;
        valueSum_r <= 0;
    end else begin
        valueA_r <= valueA;
        valueB_r <= valueB;
        valueSum_r <= multiply_result;
    end
end
endmodule
```

~rst_n (!rst_n). negedge reset 要用 !rst_n. 或是 ~rst_n. -6

1 1 1 0 1 0 0 0
0 0 0 1 0 1 0 0
64 32 16 8 4 2 1 0
0 0 1 1 0 0 0 0
1 1 0 0 1 1 1 1
1 1 0 1 0 0 0 0
0 0 1 0 1 0 0 0
1 1 0 1 0 1 1 1

B. (4pts) Please show the waveform for the circuit in part A. Suppose that the signal **valueSum** is reset to zero in the beginning and all the syntax error and functionality are correct. Use "xx" to indicate values that cannot be determined given the provided information. 4

1 1 0 1 1 0 0 0
128 64 32 16 8 4 2 1



4 5 -5 -4

valueA 0

valueB 0 -6 7 2 5

mul'sub 0. -24. 35. -10. -20.

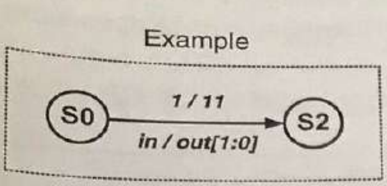
2. Finite State Machine and Simulation (8pts)

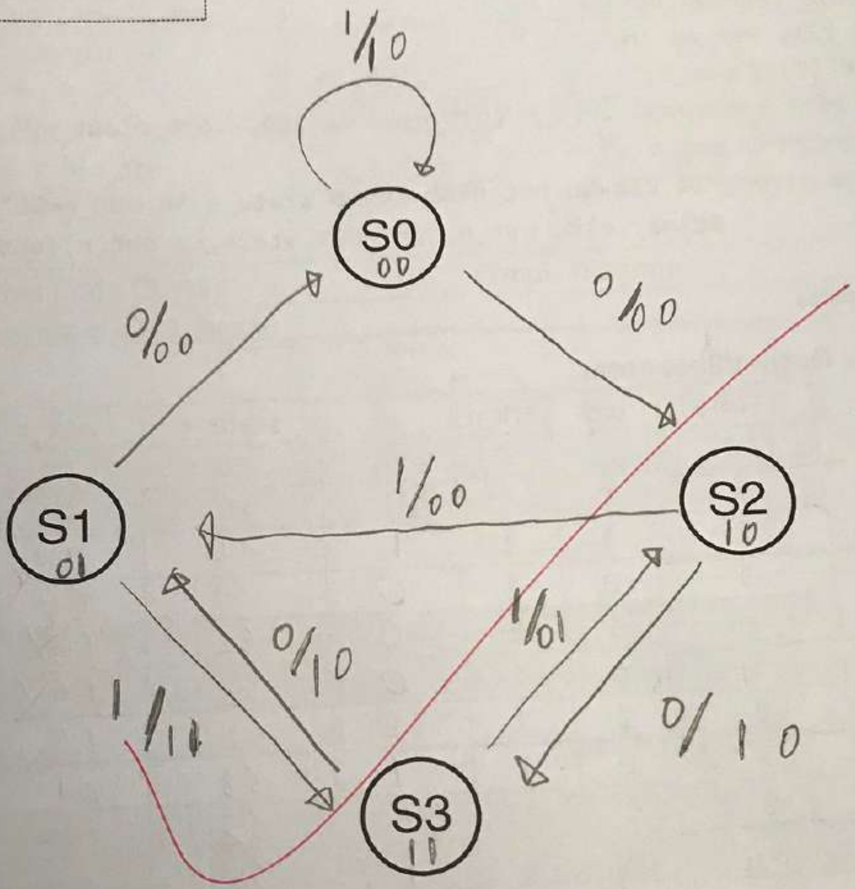
Given a Finite-State-Machine (FSM) as below.

```
module FSM (clk, rst_n, in, out_r);
parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

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

reg [1:0] out_r, out;
reg [1:0] state_c, state_n;

always @(*) begin
    case(state_c)
        S0: state_n = (in == 1'b0) ? S2 : S0;
        S1: state_n = (in == 1'b0) ? S0 : S3;
        S2: state_n = (in == 1'b1) ? S1 : S3;
        S3: state_n = (in == 1'b1) ? S2 : S1;
    endcase
end

always @(posedge clk or negedge rst_n) begin
    if(~rst_n) begin
        state_c <= S0;
        out_r   <= 2'b00;
    end
    else begin
        state_c <= state_n;
        out_r   <= out;
    end
end

always @(*) begin
    out[1] = in ^ state_c[1];
    out[0] = in & state_c[0];
end
endmodule
```

(a) **(3pts)** Please draw a Mealy state transition graph below for this FSM.

Example





(b) **(5pts)** The FSM is included as a design under test (DUT) in the testbench. After simulation, the terminal shows the outputs. Please design a DUT and input pattern to test **all the possible paths (include self-loops)** based on operations of the FSM **within the given period**.
(You should only change the value of `in` on **negedge** `clk`)

-X

```
module testbench;
reg clk, rst_n, in;
wire [1:0] out_r;
FSM DUT(.clk(clk), .rst_n (rst_n), .in(in), .out_r(out_r));
always @(*) begin
    $monitor("%t clk=%b rst_n=%b in=%b state_c=%d out_r=%b",
            $time, clk, rst_n, in, DUT.state_c, out_r);end
// ...
endmodule
```

Monitor Output Response:

negedge.

Time	clk	rst_n	in	state_c	out_r	
0	1	0	1	S0	00	
1	0	1	1	S0	00	
2	1	1	1	S0	10 ✗	S0 (00)
3	0	1	0	S0	00	
4	1	1	0	S2.	10 ✗	S2 (10)
5	0	1	0	S2	10 ✗	
6	1	1	0	S3	10 ✗	S3 (11)
7	0	1	1	S3	01	
8	1	1	1	S2.	00	S2 (10)
9	0	1	1	S2.	00	
10	1	1	1	S1	11	S1 (01)
11	0	1	1	S1	11	
12	1	1	1	S3	01 ✗	S3 (11)
13	0	1	0	S3	10	
14	1	1	0	S1	00	S1 (01)
15	0	1	0	S1	00	
16	1	1	0	S0	~~00~~	S0 (00)

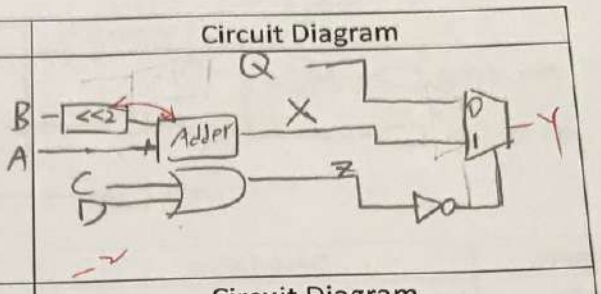
00

3. Logic Synthesis + Blocking & Non-Blocking (12 pts)

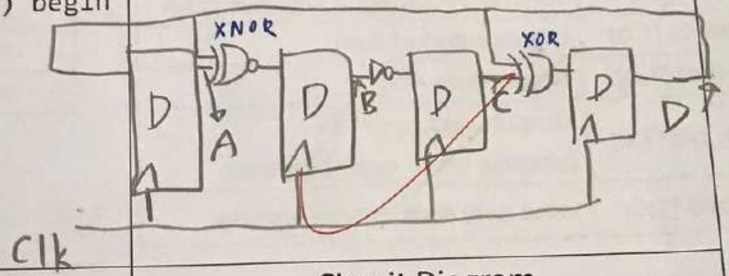
Please draw the corresponding circuits (in the right column) according to the Verilog codes (in the left column). You can use AND, OR, NAND, NOR, XOR, XNOR, NOT, MUX, D Flip-Flop, Latch, Shifter, Adder, Multiplier in the circuit diagram.

-2

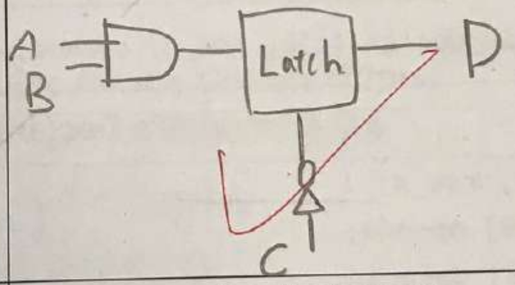
(a) Verilog Code **(3 pts)**	Circuit Diagram

`always @(*) begin`
`  X = A + B << 2;`
`  Z = C | D;`
`  Y = (~Z) ? X : Q;`
`end`
	 Q
	B
	<<2
	Adder
	A
	X
	0
	1
	Y
	C
	D
	Z
	-2

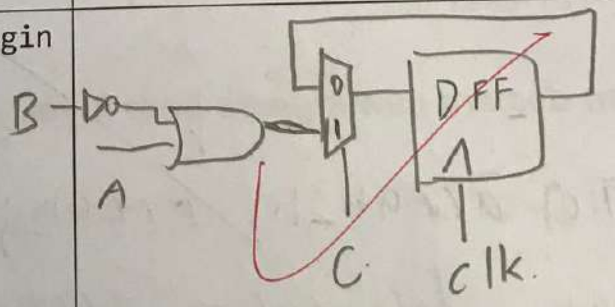
(b) Verilog Code **(3 pts)**	Circuit Diagram

`always @(posedge clk) begin`
`  A <= D;`
`  B <= A ~^ D;`
`  C <= ~B;`
`  D <= C ^ D;`
`end`
	

(c) Verilog Code **(3pts)**	Circuit Diagram

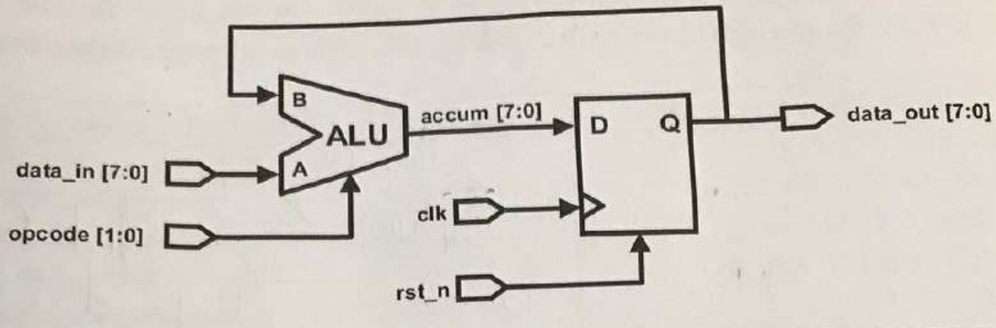
`always @(A or B or C) begin`
`  if (~C)`
`    D = A & B;`
`end`
	

(d) Verilog Code **(3pts)**	Circuit Diagram

`always @(posedge clk) begin`
`  if (C)`
`    D <= A | ~B;`
`end`
	

4. Verilog Design (6pts) +5.5

A deign with an ALU unit and its description of functions are shown below.



Signal	Description
clk	Input clock
rst_n	Input asynchronous negative reset
data_in [7:0]	Input unsigned data
accum [7:0]	Result from ALU
data_out [7:0]	Output unsigned data (positive clock edge triggered)
opcode [1:0]	Input operation control signals

opcode	ALU operation
2'b00	A + B
2'b01	A - B
2'b10	A XOR B
2'b11	A NOR B

(6pts) Please complete the Verilog code of this counter (you **don't** have to consider overflow in this design).

```
module alu_design (clk, rst_n, opcode, data_in, data_out);
```

I/O & Reg/Wire Declaration (1pt)

```
input clk, rst_n;
input [1:0] opcode;
input [7:0] data_in;
output [7:0] data_out;
```

// You can declare new signals here if you need

```
reg [7:0] accum_n, accum;
assign data_out = accum;
```

Combinational Logic for the Counter (3pts)

```
always @ (*) begin
    case ( opcode)
    2'b00 : accum_n=data_in + data_out;
    2'b01 : accum_n=data_in - data_out;
    2'b10 : accum_n=data_in ^ data_out;
    2'b11 : accum_n=data_in~^ data_out;
    endcase .
```

-0.5

```
end
```

Sequential Logic for the Counter (2pts)

```
always @ (posedge clk or negedge rst_n) begin
    if (~rst_n)  begin
        accum<=8'd0;
    end else begin
        accum <= accum_n;
end
```

```
endmodule
```

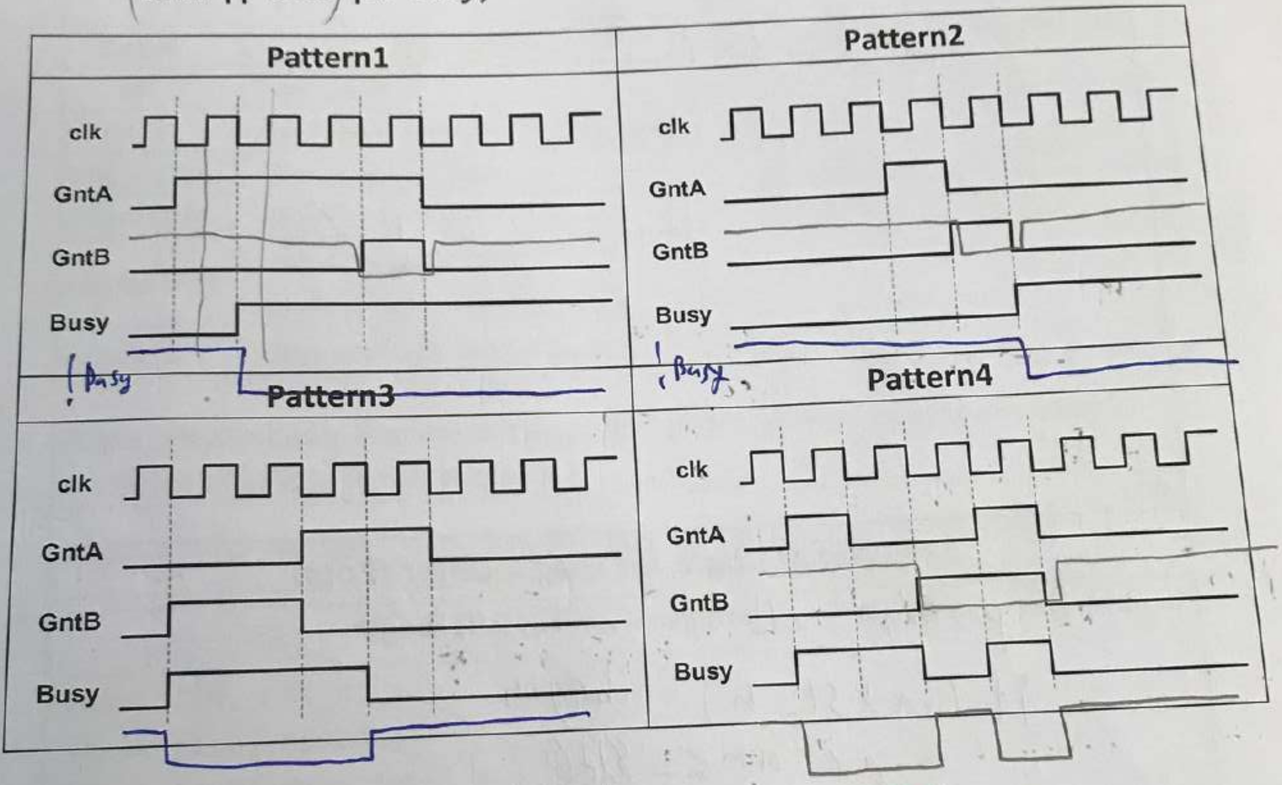
5. Formal Verification (9pts)

Consider 3 assertions and 4 patterns below. Please identify the patterns which are violated for each assertion.

(a)
```
assert_0: assert property (@(posedge clk)
  !(GntB && !Busy))
```

(b)
```
assert_1: assert property (@(posedge clk)
  GntA && !GntB |-> ##[0:2] Busy)
```

(c)
```
assert_2: assert property (@(posedge clk)
  (GntA || GntB) |=> Busy)
```



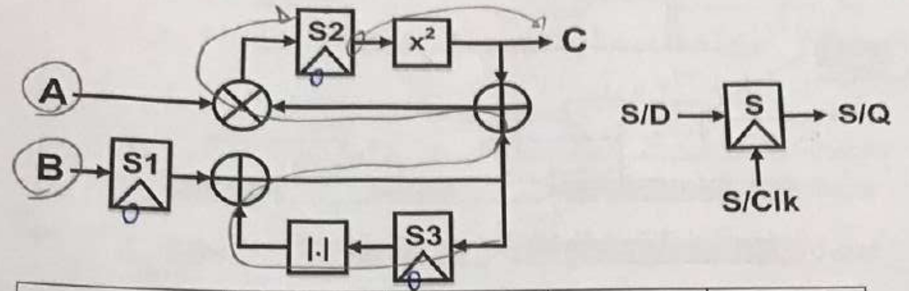
-2

(9 pts) Fill in **Pass/Fail** in each block.

	Pattern1	Pattern2	Pattern3	Pattern4
(a)	Pass	Fail	Pass.	Fail
(b)	~~Fail~~	Pass.	Fail	~~Fail~~
(c)	Pass	Fail	Fail	Fail

6. Timing path & Setup/hold time (22pts)

Consider the circuit and the delay information below. Answer the following questions.



Operation	⊕	⊗	|·|	x²
Min Delay (ns)	0.5	3.0	0.2	1.5
Max Delay (ns)	1.0	4.0	0.5	2.0

A. (5pts) List all timing paths. How many timing paths in total? (e.g. A -> S1/D)

共 3 case

1 A → S2/D : tpd = 4, tcd = 3

2, B → S1/D tpd = 0, Tcd = 0

2⊕, 1⊗ 3, S1/clk → S2/D tpd = 2+4 = 6, tcd = 2·0.5+3 = 4

1 ⊕ 4, S1/clk → S3/D tpd = 1, tcd = 0,5

x² 5 S2/clk → C. tpd = 2, tcd = 1,5

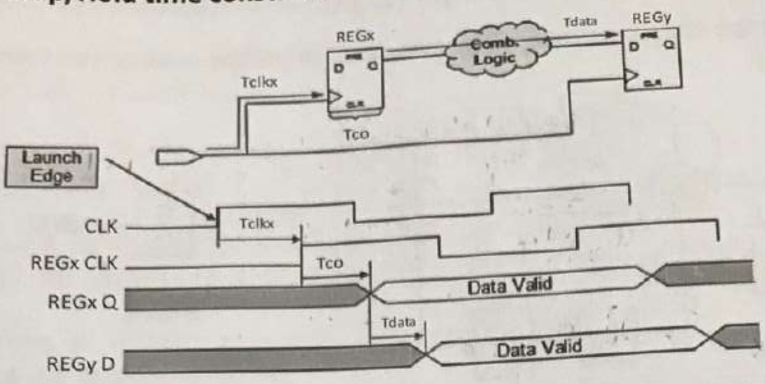
x², ⊕, ⊗ 1, 1, 1 6 S2/clk → S2/D. tpd = 2+4+1 = 7. tcd = 0,5+1,5+3 = 5

|·|, 2⊕, 1⊗ 1 7, S3/clk → S2/D. tpd = 0,5 + 2×1 + 4 = 6,5 tcd = 0,2 + 2×0,5 + 3 = 4,2

8, S3/clk → S3/D. tpd = 1,5 tcd = 0,7

$\boxed{8 \text{ in total}}$ #

B. Setup/Hold time constraint



Following the circuit in **6.A**. The registers' timing diagrams are shown above. T_{co} = **0.5ns**, and T_{clk} = **0.3ns** are the same for every register. Assuming that **T_{setup} (Setup Time) = 0.4ns, T_{hold} (Hold Time) = 0.1ns**, and **$D_A = D_B = D_C$ = 1.5ns**. The circuit operates at the clock frequency of **125MHz.**

B1. (5pts) Check that whether there are setup time violations or not by showing if the related inequalities for all timing paths are satisfied. If setup time violation(s) occur, identify the violated path(s). → $T_{data} = T_{pd}$.

125MHz → $T_{period} = \frac{1}{1,25\times10^8} = 8$ ns.

① Sx/clk → Sx/D $T_{clkx} + T_{co} + T_{data} + T_{su} \le T_{period} + T_{clky}$.

$T_{su} \le T_{period} - T_{clkx} - T_{co} - T_{data} + T_{clky}$

② A, B → Sx/D $D_A + T_{data} + \boxed{T_{su}} \le T_{period} + T_{clk} \Rightarrow T_{su} \le T_p + T_{clk} - D_y - T_{data}$

③ Sx/D → C. $D_c + T_{su} \le T_{period} \Rightarrow T_{su} \le T_{period} - D_c$.

1, $T_{su} \le 8 + 0,3 - 1,5 - 4 = 3,8$ (V) ✗

$8 + 0,3 - 1,5 = 6,8$ (V) ✓

2, $T_{su} \le 8 - 0,5 - 6 = 1,5$ (V) ✓

3, $T_{su} \le 8 - 0,5 - 1 = 6,5$ (V) ✓

4, $T_{su} \le 8 - 0,5 - 2 = 5,5$ (V) ✗

5 $T_{su} \le$ (V) ✓

6, $T_{su} \le 8 - 0,5 - 7 = 0,5$

$T_{su} \le 8 - 0,5 - 6,5 = 1$ (V) ✓

7, 8, $T_{su} \le 8 - 0,5 - 1,5 = 6$ (V) ✓

−2

⇒ all path pass

B2. (5pts) Check that whether there are hold time violations or not by showing if the related inequalities for all timing paths are satisfied. If hold time violation(s) occur, identify the violated path(s). ⇒ $\boxed{T_{cd} = T_{data}}$ from A,1

~~cases~~ 共 3 case

① (Sx/clk → Sy/D) $\cancel{T_{clk}} + T_{co} + T_{data} \geq T_{hold} + \cancel{T}_{clkx}$

→ $T_{hold} \leq T_{co} + T_{data} + T_{clkx} - T_{clky}$

② A, B → Sx/D $\boxed{T_{data} + D_{d}} \geq T_{hold} + T_{clk}$

→ $T_{hold} \leq D + T_{data} - T_{clkx}$

③ Sx/D → C $D_{c} + 0 \geq D_{c}$ $\geq T_{hold} + 0$ (V) ✗

-1

1, $T_{hold} \leq 1,5 + 3 - 0,3 = 4,2$. (V)

2, $T_{hold} \leq 1,5 + 0 - 0,3 = 1,2$. (V)

3, $T_{hold} \leq 0,5 + 4 = 4,5$ (V)

4, $T_{hold} \leq 0,5 + 0,5 = 1$ (V).

5, $T_{hold} \leq 1,5$

6, $T_{hold} \leq 0,5 + 5 = 5,5$ (V)

7, $T_{hold} \leq 0,5 + 4,2 = 4,7$ (V)

8, $T_{hold} \leq 0,5 + 0,7 = 1,2$ (V)

⇒ all path pass

B3. (7pts) Assuming that there is **time skew** between T_{clk3} and other T_{clk}, the relationship between those parameters is: $T_{clk3} = T_{clki} + 0.1$***Clk_cycle, where i = 1, 2. T_{clk1}, T_{clk2}, and other conditions are the same as A2**. Explain if there are timing violations in this circuit. If yes, identify the violated path(s). ⇒ clock period.

$T_{clk3} = T_{clki} + 0,1 \cdot 8 = 1,1$

① set-up: 找 clock skew 為負之情形

→ 看 path ⑦

⑦ $T_{su} \leq 8 - 0,5 - 6,5 - 1,1 + 0,3 = 0,2$.

$0,4 = T_{su} \leq 0,2$ ✗ ⇒ path 7 setup timing violation

(S3/clk → S3/D).

Q.E.D.

② hold time: 找 clock skew 為正之 case

⇒ 看 path ④

④ $T_{hold} \leq 0,5 + 0,5 + 0,3 - 1,1 = 0,2$.

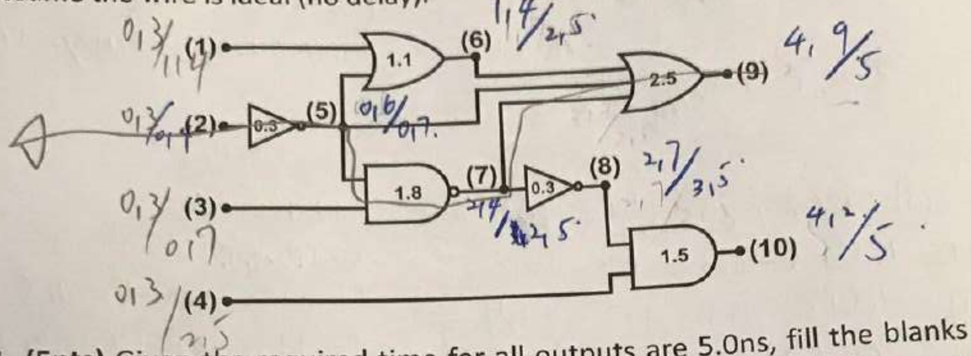
$0,1 = T_{hold} \leq 0,2$ (V). path 4 pass

~~⑦~~ hold?

④ setup?

7. Slack graph (8pts)

The figure below shows a combinational circuit with primary inputs. The delay of each input is 0.3ns and the delay of every gate is given on the gate (unit: ns). Assume the wire is ideal (no delay).



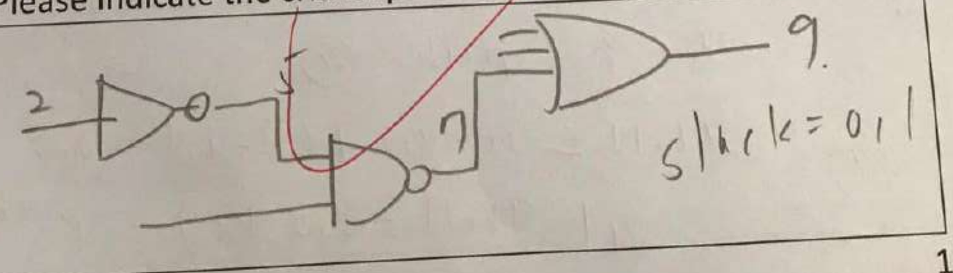
-2

C1. (5pts) Given the required time for all outputs are 5.0ns, fill the blanks on the form with corresponding arrival time, required time, and slack of every node.

AT = Arrival Time, RT = Required Time.

Node	(1)	(2)	(3)	(4)	(5)
AT (ns)	0, 3	0, 3	0, 3	0, 3.	0, 6.
RT (ns)	1, 4	0, 4.	0, 7	2, 5.	0, 7
Slack (ns)	1, 1	0, 1	0, 4	2, 2	0, 1
Node	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**
AT (ns)	1, 7	2, 4	2, 7	4, 9.	4, 2,
RT (ns)	2, 5.	2, 5.	3, 5.	5	5
Slack (ns)	0, 8.	0, 1	0, 8	0, 1	0, 8

C2. (3pts) Please indicate the critical path and show its slack values.

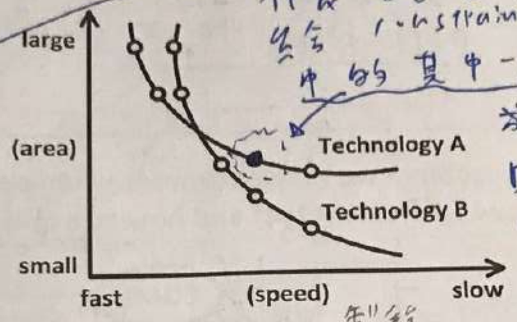


slack = 0, 1

8. Synthesis (33pts)

1. Please explain why synthesis is **constraint-driven** and what is the meaning of the **tradeoff between area and speed** according to the following image. **(2pts)**

"constraint-driven" 代表 DC 在合成時要下 tcl 給 constraint, 給定 design space 中的其中一點", 此時 DC 才會合成。若沒下 constraint, DC 是無法合成出 designer 想要的 design。

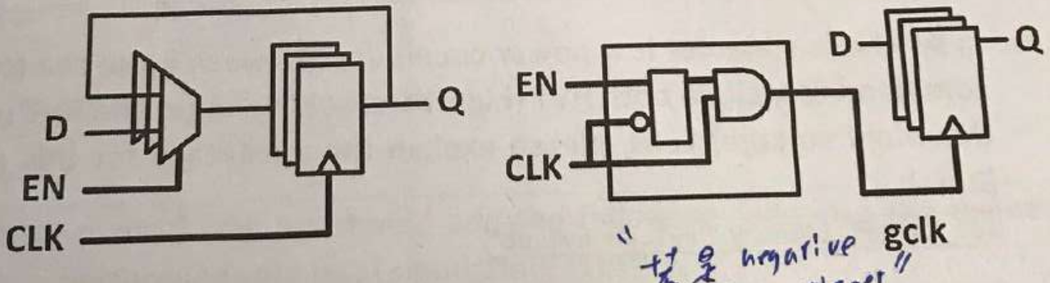


-0.5

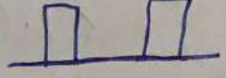
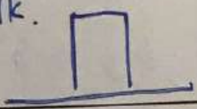
從上圖可以看出不論製程 A, B, speed 和 area 都成 "反比"。 負相關
因此, 當我想要 area 小之 design, spped 自然會慢
反之亦然。

2. Why we preferred the architecture on the right hand side instead of left hand side design in clock gating? **(2pt)**

0



"若是 negative edge trigger"

左邊的 ckt 在 clock gating 時會有 glitch, 如下圖 gclk. , 因此要用右邊的 Latch 避免 glitch 生成。 gclk. 

?

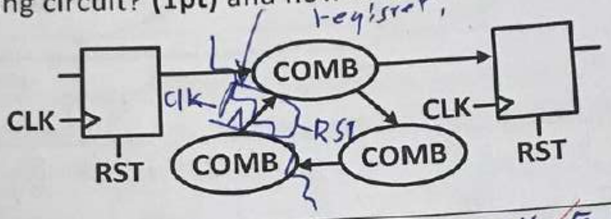
3. Please explain why should we consider ***PVT condition*** when synthesis **(1pt)** and what does ***PVT*** stand for? **(1pt)**

~~a,~~ 在 technology library (.db) 中有 PVT condition 資訊，要指定ㄓㄜ一條件 constraint driven 之合成 tool，才能合成。

−1

b, process, voltage, temperature.

4. What is the problem we may encounter when performing logic synthesis for the following circuit? **(1pt)** and how to avoid this situation? **(1pt)**



①, STA 分析，timing 不[illegible]收斂.

② 打斷 comb loop，在中間插一級 register，(如圖)

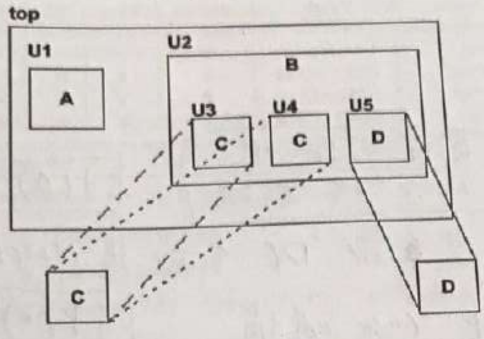
5. In synthesis stage for low power circuit design, we will use the following command to include both HVT (high-threshold voltage) and RVT (regular-threshold voltage) cells, please explain the advantage for this process. **(2pts)**

```
set_target_library "rvt.db hvt.db"
```

在 non-critical path 上用 "hvt.db"，使得 threshold voltage 上升，減少電容充放電之漏電流
在 critical path 上用 "rvt.db"，使 design 有更好的 timing。

6. In a hierarchical design, submodules are sometimes referenced by more than one cell instance. What method is required to solve this situation? **(1pt)** And what is the effect after applying this method? **(1pt)**

~~unqui quity~~
ununily



① 使用 don touch method，先合成 submodule，再合成整個 design。但 submodule 要在 top 下 don't_touch_command。

② compile 速度加快，memory 用量變少，維持 design Hirarachy.

7. Please explain the difference between ***RTL Level Netlist*** and ***Gate Level Netlist***. **(1pt)**

RTL 沒有 timing 資訊

但 Gate level 有。

—1

↳ 看不到

8. Please explain why we should add the following code into the testbench when performing gate level simulation. **(1pt)**

```
$sdf_annotate ("SDF_FILE_NAME", top_module_instance_name);
```

SDF 裡面有 timing 資訊，gate level sim 結果

若沒有加 SDF file，則

會錯誤，他會去 tsmc13.v 找 default 之 timing.

造成 sim 結果發生 timing violation,

9. Why the synthesis report shows register with <u>latch</u> type when loading design? **(1pt)** and please also explain how to avoid this situation. **(1pt)**

```
===============================================================================================
|  Register Name   |    Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================================
|    reg_B_reg     |   Latch    |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
|   reg_ins_reg    | Flip-flop  |   4   |  Y  | N  | Y  | N  | N  | N  | N  |
|   alu_out_reg    | Flip-flop  |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
|    reg_A_reg     | Flip-flop  |   8   |  Y  | N  | Y  | N  | N  | N  | N  |
===============================================================================================
```

RTL code 中

①, mux case 沒寫滿 → 造成 DC 合成出 latch.

if(a) out= in.

② 把 case 補滿

if(a) out = in;
else out = 0;

10. Synopsys Design Constraints File (.sdc) specifies the timing constraints for synthesis. Please write the ***commands*** might be included in the ***.sdc*** file to meet the following specification. **(10pts, 1pt for each)** $\frac{1}{2\times10^8}$

A. This design is an ALU with clock port **i_clk** with **200 MHz** operating frequency (1%)
B. All **flip-flops should meet the hold-time constraints** (1%)
C. Clock tree synthesis will be built in the place and route stage, so the **clock network cannot be re-buffered** (1%)
D. Clock tree synthesis will be built in the place and route stage, so the **clock network is considered with no delay** (1%)
E. The input delay is considered as **0.1*clock_cycle** (1%)
F. The output delay is considered as **0.1*clock_cycle** (1%)
G. The operating condition for synthesis should include both ***slow*** and ***fast*** library for ***max*** and ***min*** condition (1%)
H. The clock latency is considered as **0.05*clock_cycle** (1%)
I. The clock transition is considered as **0.01*clock_cycle** (1%)
J. The clock skew and jitter are considered as **0.1 ns** and **0.2ns, respectively** (1%)

These are ***<u>only example commands</u>*** you may use, please ***<u>choose the correct ones and modify them</u>*** to meet the abovementioned specification. **<u>(you do not have to consider the commands for including library)</u>**

```
1.  set period 10.0
2.  set_operating_conditions -max_library slow -max slow
3.  create_clock -name clk -period 10.0
```

```
4.  set_dont_touch_network [get_clocks clk]
5.  set_ideal_network [get_clocks clk]
6.  set_fix_hold [get_clocks clk]
7.  set_clock_latency 0.5 [get_clocks clk]
8.  set_clock_transition 0.1 [get_clocks clk]
9.  set_max_transition 0.1
10. set_input_delay 0.5 -clock clk [all_inputs]
11. set_output_delay 0.5 -clock clk [all_outputs]
12. set_clock_uncertainty 0.1 [get_ports clk]         -0.5
13. set_load   1 [all_outputs]                        -4
14. set_drive  1 [all_inputs]                         -1
15. set_false_path -from {A} -through {C} -to {OUT}
16. set_max_delay 1 from [all_inputs] -to [all_outputs]
```

Clock specification

A, Set period 5, create-clock - name ~~T-clk~~ - period 5.0 T-clk

B, set_fix_hold [get clocks clk]

D set_ideal_network [get clocks clk]

C. set_dont_touch_network[get clocks clk]

E set-input_delay 0,5 - clock clk [all inputs]

F set-output_delay 0,5 -clock clk [all_outputs]

G set_operating_conditions - max_library slow -max slow

set-operating-conditions -min_library fast -min fast.

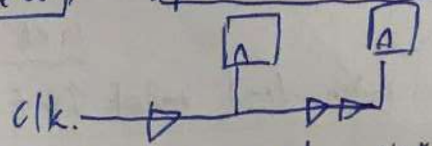
H set_clock_latency. 0,25 [get-clocks clk]

I. set_clock_transition. 0,05 [get-clocks clk]

J. set_clock_uncertainty ~~0,2~~ 0.3 [get ports clk].

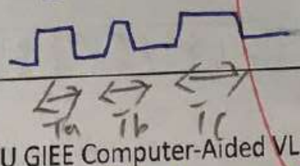
11. Two situations are considered for clock uncertainty: clock skew and clock jitter. Please explain the definition of ***clock skew*** **(1pt)** and ***clock jitter*** **(1pt)**.

skew: spatial variation of clock arrival time.



⇒ clock 到每個 Flip flop 時間不一致.

jitter: temporal variation of clock period



• Ta≠Tb≠Tc.

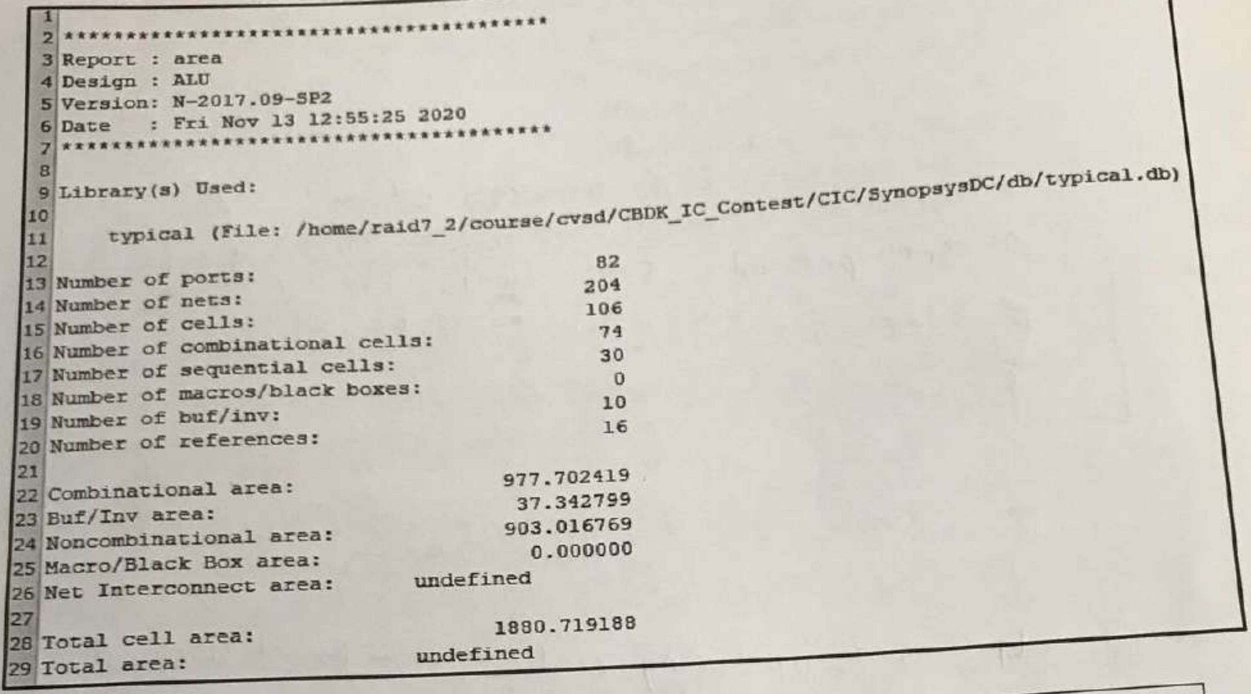
每個時刻, clock period 大小不一樣.

12.Please read the area synthesis report below and answer the questions:

a. How to fix the ***undefined interconnect area***? **(1pt)**

b. If Macro/Black Box area is not zero, what might be indicated? **(1pt)**

c. In cell-based IC design flow, we will mainly focus on total cell area instead of total area, please explain ***why***. **(1pt)**



```
****************************************
Report : area
Design : ALU
Version: N-2017.09-SP2
Date   : Fri Nov 13 12:55:25 2020
****************************************

Library(s) Used:

    typical (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/typical.db)

Number of ports:                           82
Number of nets:                           204
Number of cells:                          106
Number of combinational cells:             74
Number of sequential cells:                30
Number of macros/black boxes:               0
Number of buf/inv:                         10
Number of references:                      16

Combinational area:                977.702419
Buf/Inv area:                       37.342799
Noncombinational area:             903.016769
Macro/Black Box area:                0.000000
Net Interconnect area:          undefined

Total cell area:                  1880.719188
Total area:                     undefined
```

a, set wire load model

b, 使用非 standard cell 之 IP,
例如：HW3 中的 SRAM cell

c, 因為在 DC 時，chip 實際在電路中的
位置不確定，所以用 wire load model
所估計出之 total area. 包含 cell area 和
net area
但此時的 wire load model 很不準確，
因此，只需 focus 在 cell area 即可。
Total area 之評估等到 APR 結束後.

才會是比較正確的估計。

13. When performing timing analysis with the following command, if there are ***only*** two paths in the timing report ***Design.timing*** shown as the following table, please indicate which path is the critical path **(1pt)** and explain why. **(1pt)**

```
report_timing -path full -delay max > Design.timing
```

Path1

```
Startpoint: Y[0] (input port clocked by clk)
Endpoint: FB_A[11] (output port clocked by clk)
Path Group: clk
Path Type: max

Point                                  Incr       Path
------------------------------------------------------
clock clk (rise edge)                  0.00       0.00
clock network delay (ideal)            0.50       0.50
input external delay                   5.00       5.50 f
Y[0] (in)                              0.00       5.50 f
U622/Y (CLKINVX1)                      0.04       5.54 r
U621/Y (MXI2X1)                        0.09       5.63 f
U432/Y (AND2X1)                        0.21       5.84 f
U293/CO (ADDFXL)                       0.36       6.19 f
U294/CO (ADDFXL)                       0.39       6.58 f
U430/Y (AND2X1)                        0.21       6.79 f
U428/Y (AND2X1)                        0.18       6.97 f
U427/Y (XOR2X1)                        0.15       7.12 f
U701/Y (MXI2X1)                        0.54       7.66 r
U368/Y (INVX12)                        0.74       8.41 f
FB_A[11] (out)                         0.00       8.41 f
data arrival time                                 8.41

clock clk (rise edge)                 10.00      10.00
clock network delay (ideal)            0.50      10.50
clock uncertainty                     -0.10      10.40
output external delay                 -0.50       9.90
data required time                                9.90
------------------------------------------------------
data required time                                9.90
data arrival time                                -8.41
------------------------------------------------------
slack (MET)                                       1.49
```

Path2

```
Startpoint: Y[1] (input port clocked by clk)
Endpoint: FB_A[10] (output port clocked by clk)
Path Group: clk
Path Type: max

Point                                  Incr       Path
------------------------------------------------------
clock clk (rise edge)                  0.00       0.00
clock network delay (ideal)            0.50       0.50
input external delay                   5.00       5.50 r
Y[1] (in)                              0.00       5.50 r
U620/Y (CLKINVX1)                      0.04       5.55 f
U619/Y (MXI2X1)                        0.12       5.66 r
U293/CO (ADDFXL)                       0.61       6.27 r
U294/CO (ADDFXL)                       0.33       6.60 r
U430/Y (AND2X1)                        0.19       6.79 r
U429/Y (XOR2X1)                        0.16       6.95 f
U702/Y (MXI2X1)                        0.56       7.51 r
U363/Y (INVX12)                        0.75       8.26 f
FB_A[10] (out)                         0.00       8.26 f
data arrival time                                 8.26

clock clk (rise edge)                 10.00      10.00
clock network delay (ideal)            0.50      10.50
clock uncertainty                     -0.10      10.40
output external delay                 -0.50       9.90
data required time                                9.90
------------------------------------------------------
data required time                                9.90
data arrival time                                -8.26
------------------------------------------------------
slack (MET)                                       1.64
```

a, "Path 1"

b, 因為其 slack 較小，代表 timing 比 Path 1 緊很多
(=required - arrival time)