2021 DCS 期末考

第一題 ：

Note. In the following, you don't need to write a full Verilog/SystemVerilog code module to illustrate your answers. Just a piece of code is enough.

a (2%)What is the difference between === and ==? Use a short example to show your answers.

b. (2%)What is the difference between = assignment and <= assignment? Use a short example to show your answers.

c. (4%)Write two coding style examples for combinational circuit.

d. (2%)What do the terms "wire", "reg" and "logic" in Verilog refer to?

第二題:

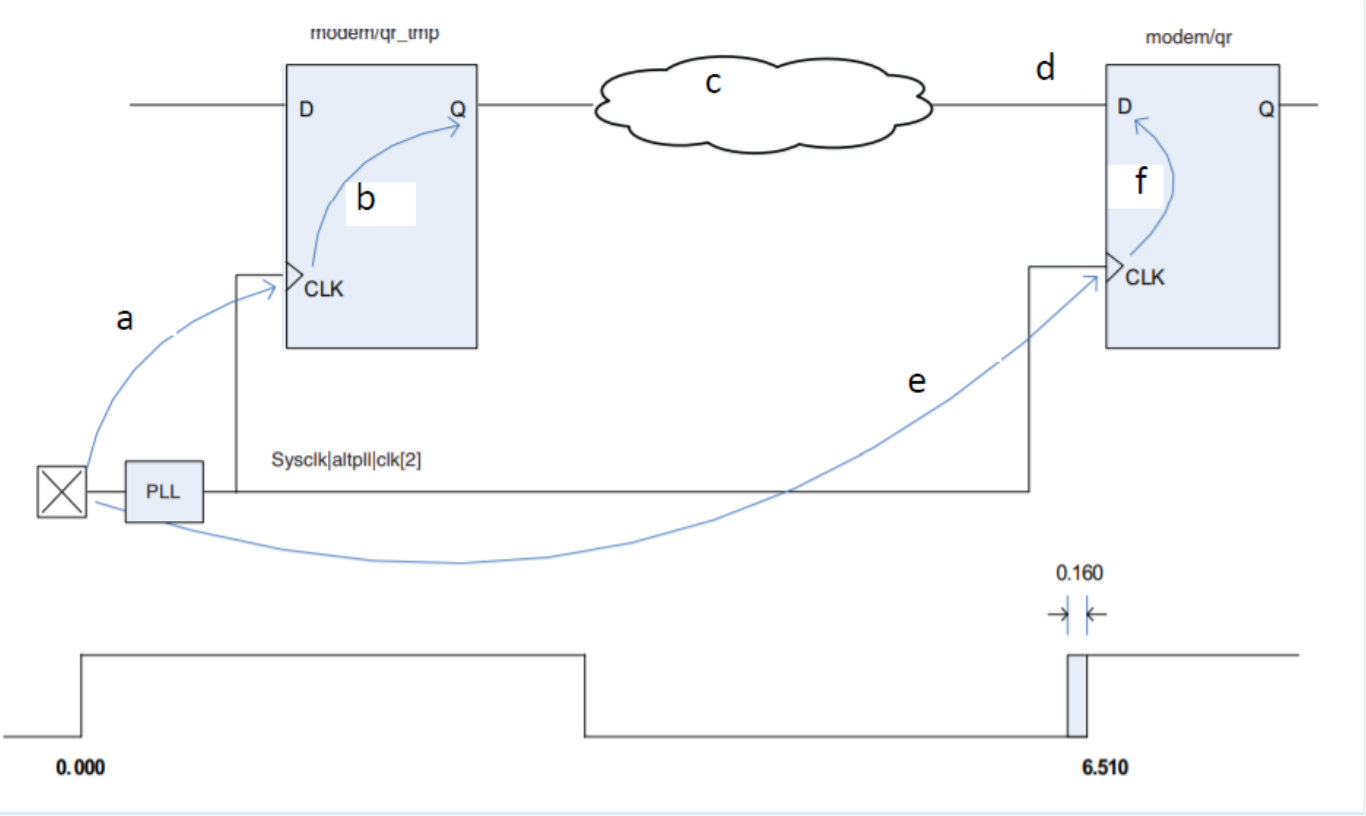
(10%) For the following timing report, please draw its register-to-register timing path labeled by a to f in the figure. (Note. You can combine all combinational logic delay into one delay.)

a. answer for path a

```
Startpoint: modem/qr_tmp
            (rising edge-triggered flip-flop clocked by Sysclk|altpll|clk[2])
 Endpoint: modem/qr
            (rising edge-triggered flip-flop clocked by Sysclk|altpll|clk[2])
 Path Group: Sysclk|altpll|clk[2]
 Path Type: max

 Point                                                    Incr       Path
 -----------------------------------------------------------------------------
 clock Sysclk|altpll|clk[2] (rise edge)                  0.000      0.000
 clock network delay (propagated)                        0.204      0.204
 modem/qr_tmp/CLK (DFF_D1_CLK1_NCLR1_CKEN1_RSCN1_SCIN1) 0.000      0.204 r
 modem/qr_tmp/Q (DFF_D1_CLK1_NCLR1_CKEN1_RSCN1_SCIN1)   0.146 &    0.350 f
 modem/qr_tmp_ASTfhInst7779/OUT (BUF_D3)                 0.073 &    0.423 f
 lcell_comb6052/OUT (BUF_D6)                             0.166 &    0.589 f
 modem/qr/D (DFF_D1_CLK1_NCLR1_RSCN1_SCIN1)              0.026 &    0.614 f
 data arrival time                                                  0.614

 clock Sysclk|altpll|clk[2] (rise edge)                  6.510      6.510
 clock network delay (propagated)                        0.321      6.831
 clock reconvergence pessimism                           0.003      6.835
 inter-clock uncertainty                                -0.160      6.675
 modem/qr/CLK (DFF_D1_CLK1_NCLR1_RSCN1_SCIN1)                       6.675 r
 library setup time                                     -0.340      6.334
 data required time                                                 6.334
 -----------------------------------------------------------------------------
 data required time                                                 6.334
 data arrival time                                                 -0.614
 -----------------------------------------------------------------------------
 slack (MET)                                                        5.720
```



第三題:

follow question 2

answer for path b

第四題:

follow question 2

answer for path c

第五題:

follow question 2

answer for path d

第六題:

follow question 2

answer for path e

第七題:

follow question 2

answer for path f

第八題:

Note. In the following, you don't need to write a full Verilog/SystemVerilog code module to illustrate your answers. Just a piece of code is enough.

Re-write the module in behavioral form. The delays can be assumed to be gate delays

```
module expl_str(x,y,a,b,c);
   input a, b, c;
   output x, y;
   wire   a, b, c, x, y;
   wire   na, nb, nc, t3, t5, t6;

   not n1(na,a);
   not n2(nb,b);
   not n3(nc,c);
   and #1 a1(t3,na,b,c);
   and a2(t5,a,nb,c);
   and a3(t6,a,b,nc);
   or o1(x,t3,t6);
   or #3 o2(y,a,t5);

endmodule
```

第九題:

Draw a schematic showing the approximate RTL-level description generated by a synthesis program you can draw it on a paper, take a photo, and upload

```
module whatsyn2(output [6:0] sum, input [15:0] nibbles, a, b, c);
   logic [15:0] n2;
   logic        last_c;

   always @( posedge a )
     if ( !b ) begin
        sum = 0;
     end else begin

        if ( c != last_c ) begin
           n2 = nibbles;
           for ( int i=0; i < 4; i++ ) begin
              sum = sum + n2[3:0];
              n2 = n2 >> 4;
           end
        end
        last_c = c;

     end
endmodule
```

第十題:

For the following code,

(a) (2%) what is the intended function of the code

(b) (2%) correct any bad things in the above code

(c) (3%) Sketch a possible synthesized circuit for the code

(d) (3%) rewrite this with assign statement

```
module block (q, I, s)
input [3:0] I;
input [1:0] s;
output q;
always @(s)
    case(s)
        00: q = i[0];
        01: q = i[1];
        10: q = i[2];
        11: q= i[3];
    endcase
endmodule
```

第十一題:

Correct any bad coding in the followings

```
module signed_adder_with_carry_in
(input  logic signed [3:0] a, b,
 input  logic              ci,
 output logic signed [3:0] sum,
 output logic              co);

  assign {co,sum} = a + b + ci;
endmodule
```

第十二題:

show the true or false result of each if condition

```
logic a;         // 1-bit 4-state variable
logic [1:0] b;   // 2-bit 4-state variable

initial begin
  a = 1;
  b = 1;

  if (!a) ... // evaluates as
  if (~a) ... // evaluates as

  if (!b) ... // evaluates as
  if (~b) ... // evaluates as
end
```

第十三題:

Correct the following code if you intend to show “a” is true if (a==1)

```
module bar;
   reg a;
   initial begin
     a = 1;
     if (a);
       $display("'a' is true");
     else

       $display("'a' is false");
   end
endmodule
```

第十四題:

Correct the following code if you intend to connect two 1-bit adder together.

```
module adder (input  logic a, b, ci,
              output logic sum, co);
  ...
endmodule

module top;
  wire a, b, ci, s1, s2, c1;
  adder i1 (.a(a), .b(b), .ci(c1), .sum(s1), .co(c1) );
  adder i2 (.a(a), .b(b), .ci(c),  .sum(s2), .co(co) );
endmodule
```

第十五題 ：

(10%) suppose you have a system where every problem must pass through four pipeline stages with delays of 30ns, 60ns, 15ns, 20ns.

(a) (2%) Which stage is the bottleneck

(b) (2%) What is the utilization rate (time spend doing useful work divided by total time) of each stage?

(c) (3%) If each stage cannot be further pipelined, but can be replicated. How many of each module do you need to have full utilization of your system? (that is, no module should be idle after a warm up period) Draw a diagram to show your answer

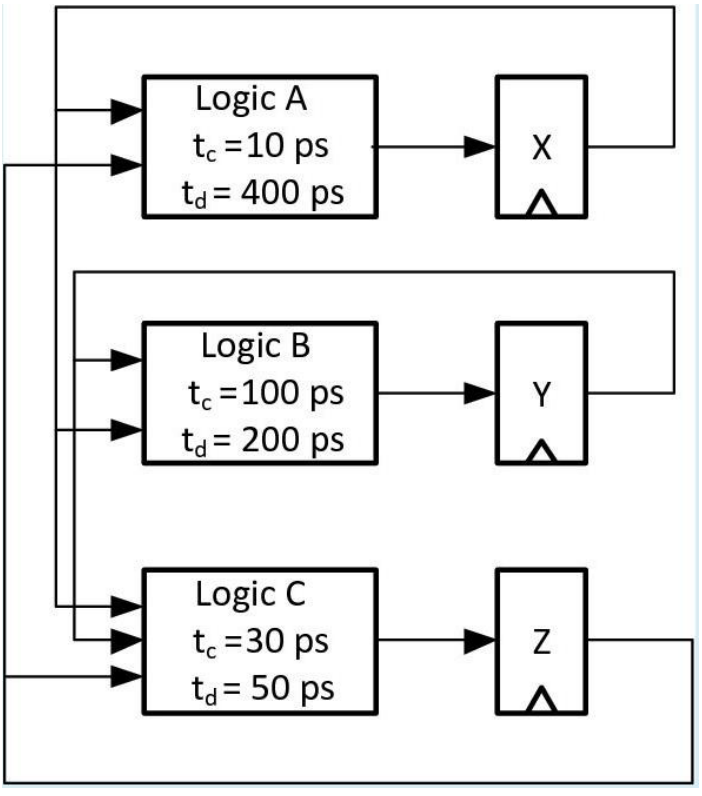
(d) (3%) If each stage cannot be replicated, but can be pipelined. What is the minimum number of stages from beginning to end in a pipeline that has no load imbalance? Draw a diagram to show your answer.

第十六題 ：

(setup time and hold time violation check)
you can write answer on the e3 text frame or upload your photo

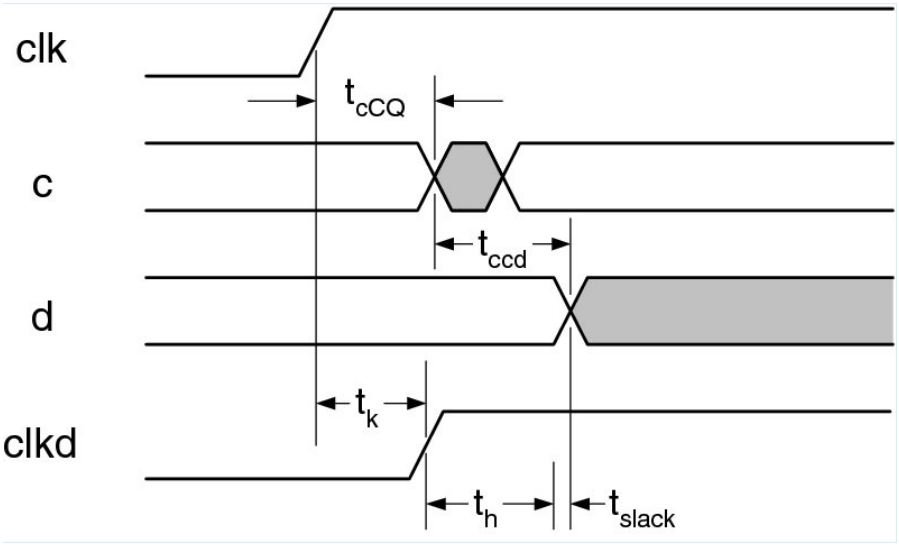
For Flip-Flop A in Table 1 (used for DFF X, Y, Z) , and a 2GHz clock, check the following figure for timing violations. If there is a hold time violation, indicate

(a) (5%) where delay must be added, and the necessary delay value

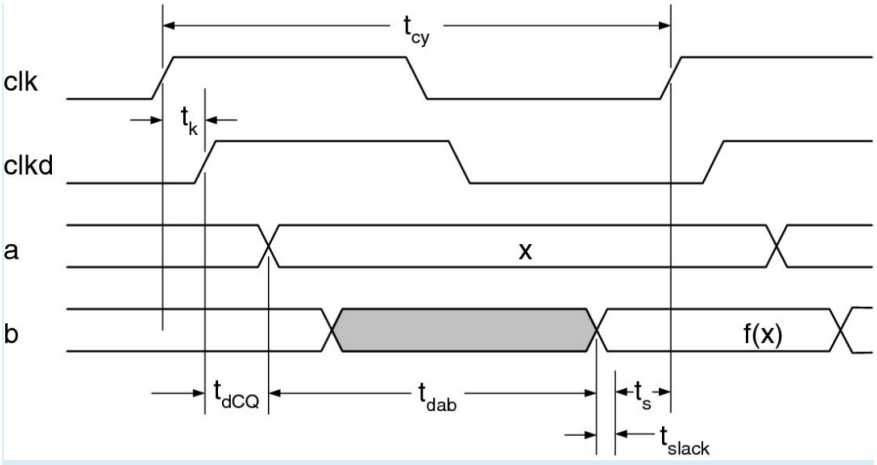


Parameter (ps)	DFF A	DFF B
Setup time: t_s	20	100
Hold time: t_h	10	-20
Contamination delay, clock to q, T_{ccq}	10	2
Propagation delay, clock to q, t_{dcq}	20	30

hold time check



setup time check



第十七題:

follow question 16

(10%) recheck for setup time violation. Is it possible to run faster than 2GHz clock? Derive the cycle time for this maximum clock frequency?

第十八題 ：

follow question 16

(15%) For DFF B in above Table (used for DFF X, Y, Z) and a 2GHz clock, check the figure again to see if there a setup time or hold time violation, and derive the minimum cycle time for this design.