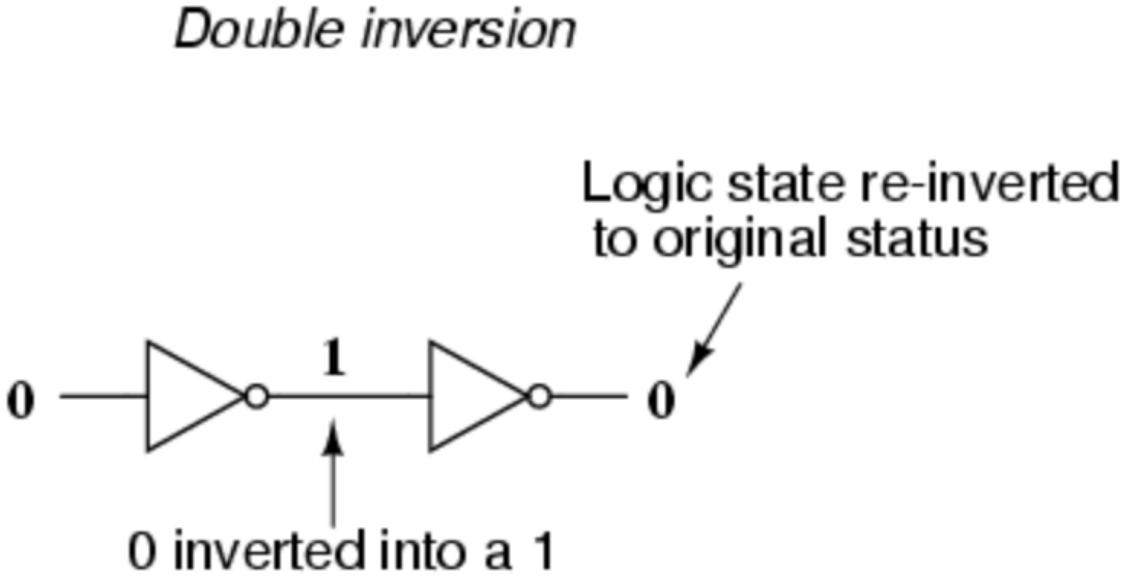
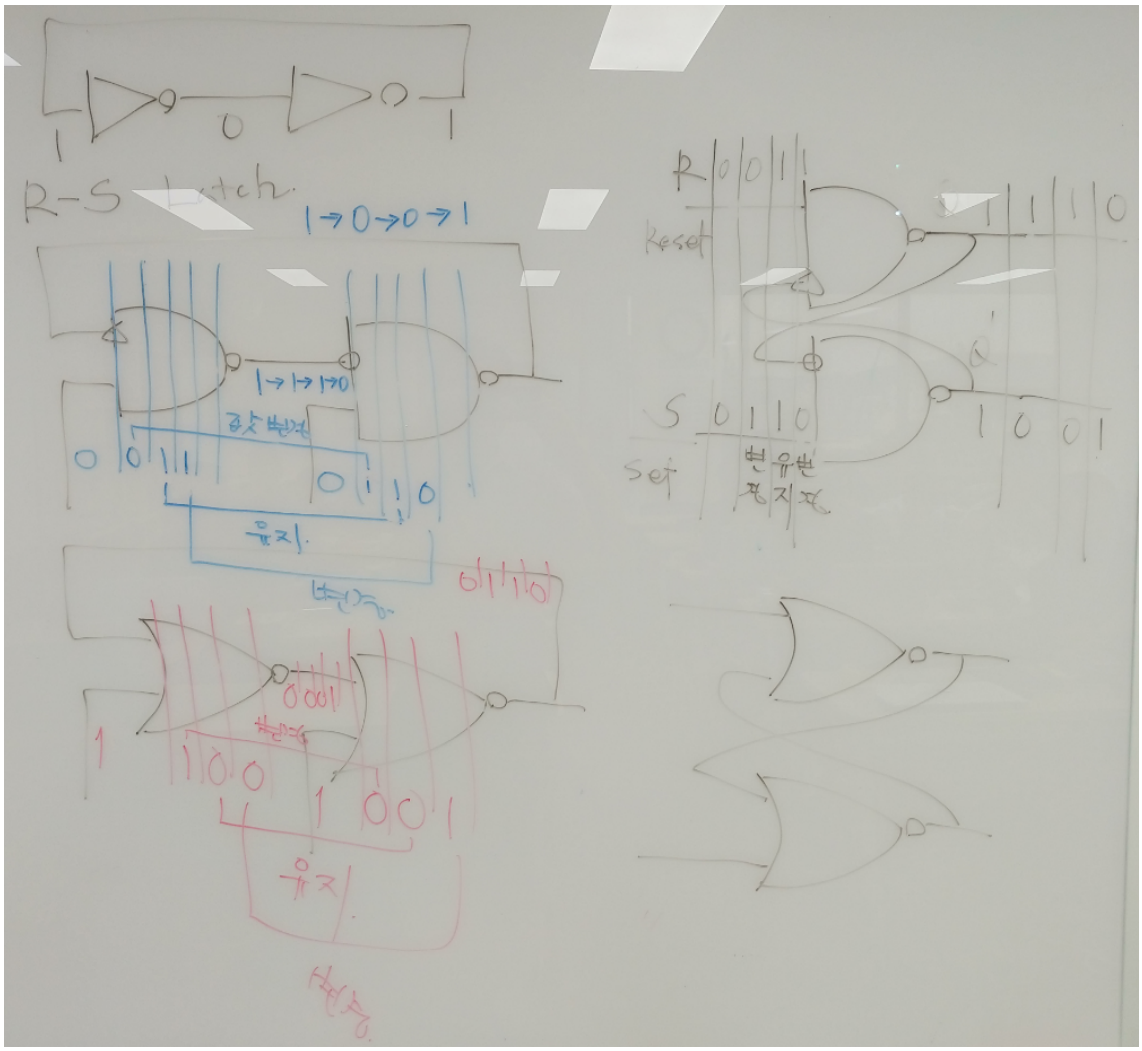
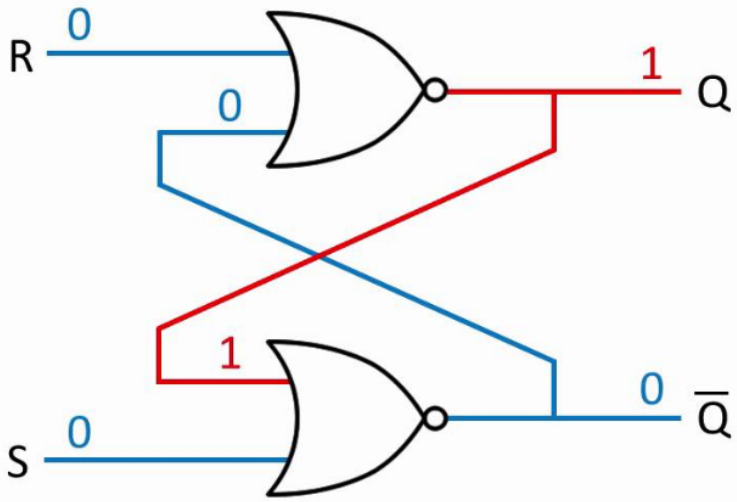
buffer gate

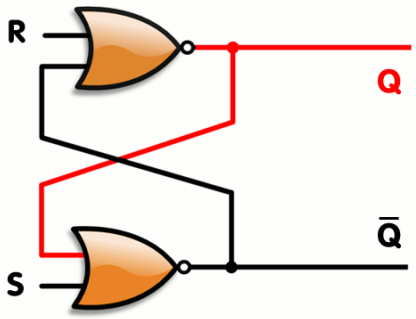

Double inversion
Logic state re-inverted
to original status
0
1
0
0 inverted into a 1

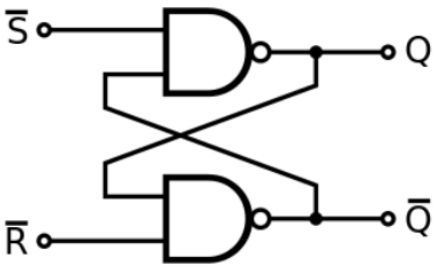

R-S latch
1→0→0→1
Reset
set

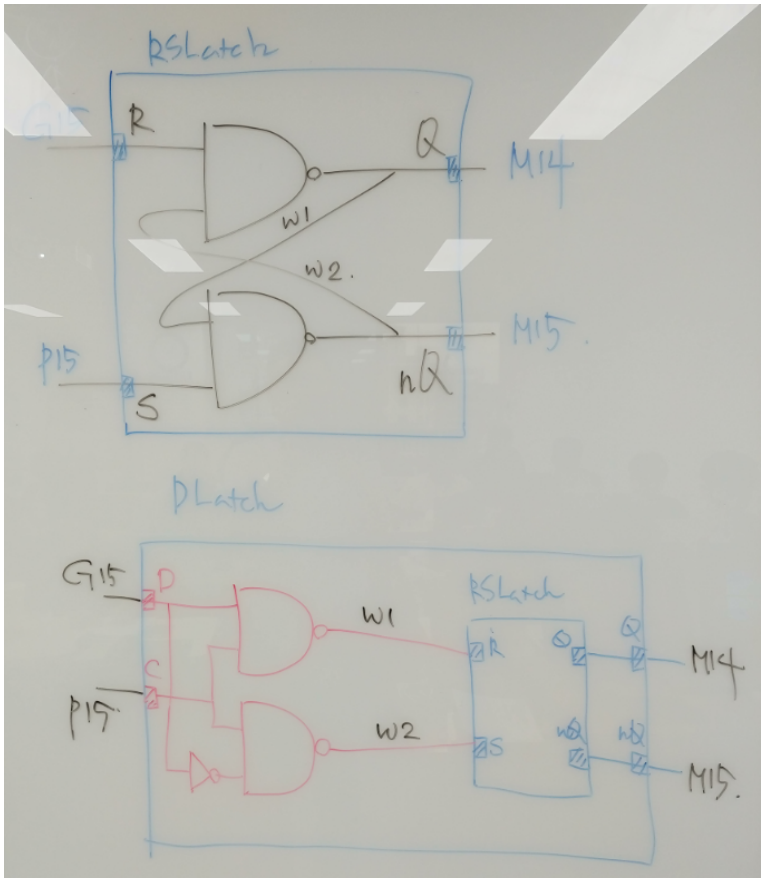
SR Latch

S	R	Q	$\overline{Q}$
0	0	1 0	0 1
0	1	0	1
1	0	1	0
1	1	0	0



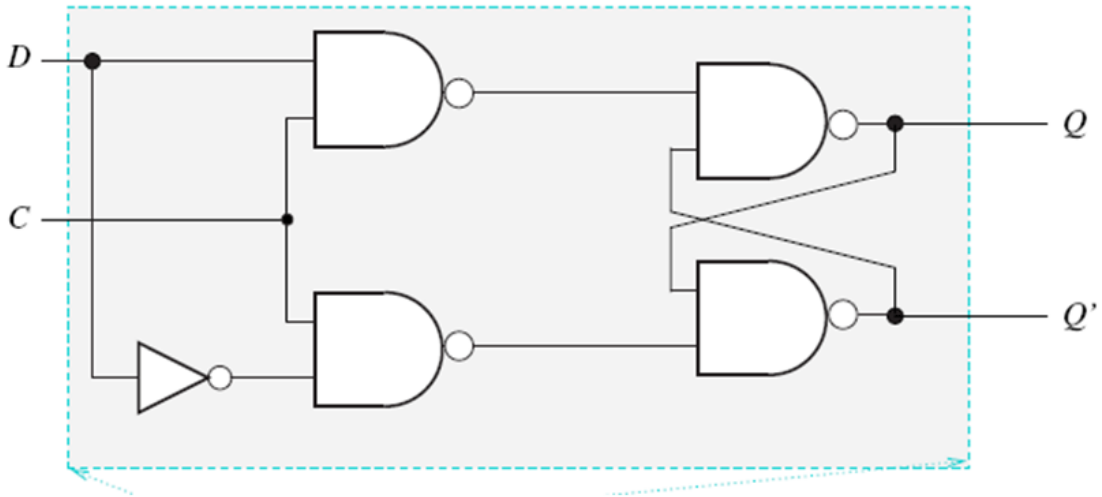




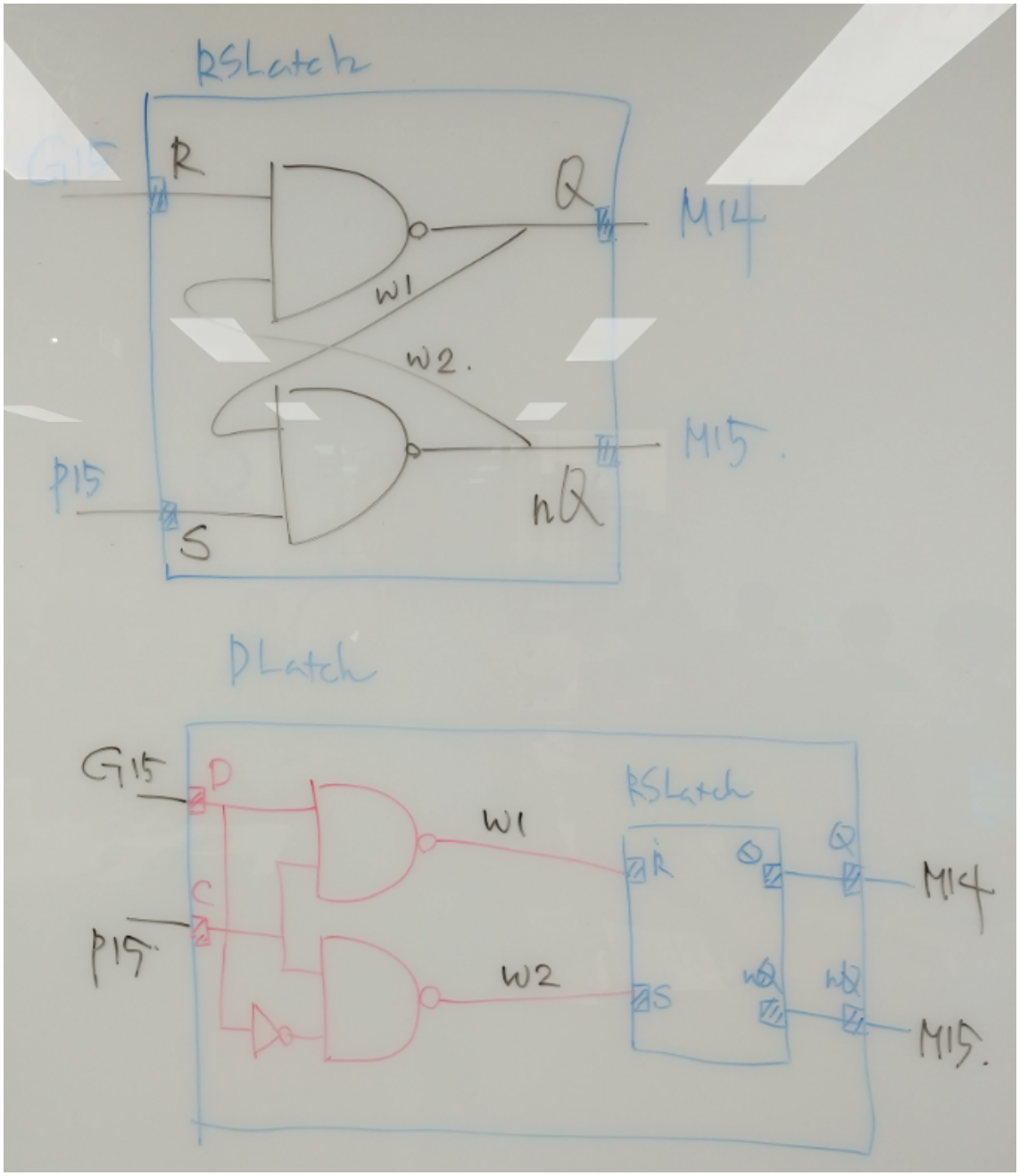

RSLatch
R
Q
M14
W1
W2.
M15.
P15
S
nQ
DLatch
G15
D
C
P15
W1
W2
RSLatch
R
S
Q
nQ
M14
M15.

```
module RSLatch(
    input R,
    input S,
    output Q,
    output nQ
    );

    (* ALLOW_COMBINATORIAL_LOOPS = "true", KEEP = "true" *)
    wire w1, w2;

    assign w1 = ~(R & w2);
    assign w2 = ~(S & w1);

    assign Q = w1;
    assign nQ = w2;

endmodule
```

DLatch



C가 0인 경우는 유지, C가 1인 경우는 값 변경


RSLatch
R
Q
M14
W1
W2.
M15.
P15
S
nQ
DLatch
G15
D
C
P15
W1
W2
RSLatch
R
S
Q
M14
M15.

```
module DLatch(
    input D,
    input E,
    output Q,
    output nQ
    );

    wire w1, w2;

    assign w1 = ~(D & E);
    assign w2 = ~(~D & E);

    RSLatch rsLatch_0(.R(w1), .S(w2), .Q(Q), .nQ(nQ));

endmodule
```

실제 사용 코드

```
module Latch(
    input D,
    input E,
    output reg Q
    output reg nQ
    );

    always @(D, E)
    begin
        if(E==1'b1)
        begin
            Q = D;
            nQ = ~D;
        end
    end

endmodule
```

```
set_property PACKAGE_PIN M14 [get_ports Q]
set_property PACKAGE_PIN M15 [get_ports nQ]
set_property IOSTANDARD LVCMOS33 [get_ports nQ]
set_property IOSTANDARD LVCMOS33 [get_ports Q]

set_property PACKAGE_PIN G15 [get_ports D]
set_property PACKAGE_PIN P15 [get_ports C]
set_property IOSTANDARD LVCMOS33 [get_ports C]
set_property IOSTANDARD LVCMOS33 [get_ports D]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets C_IBUF]
```

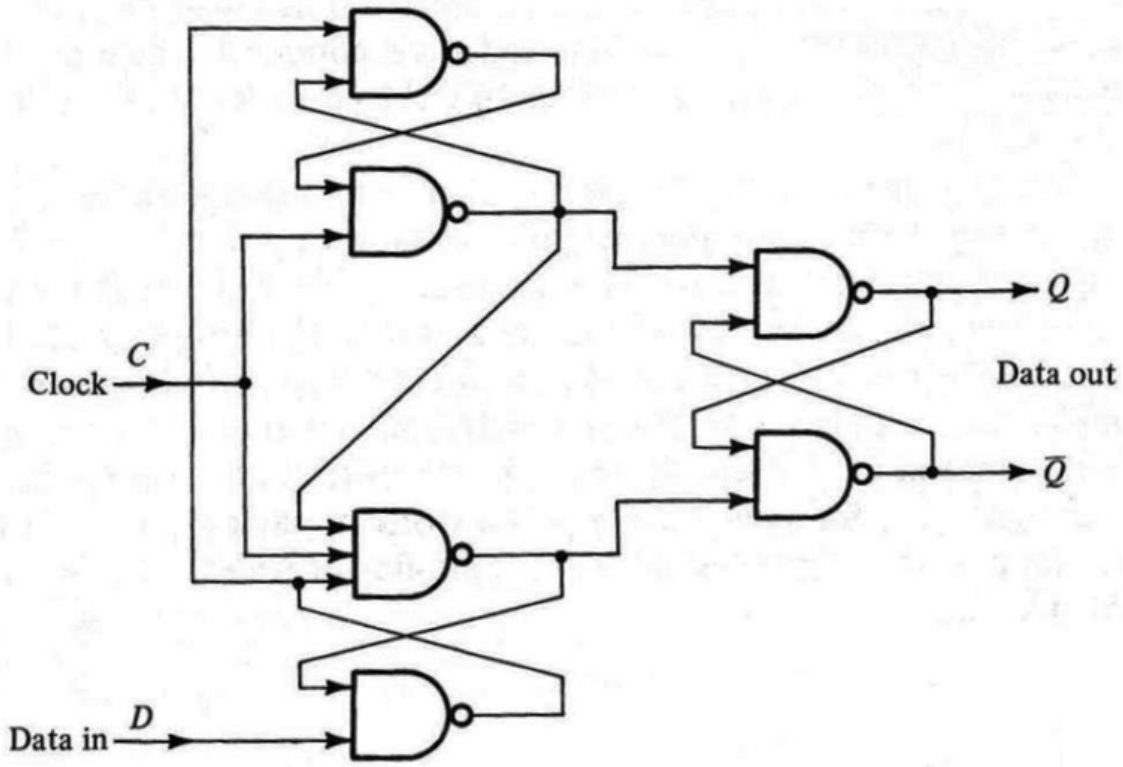
오류 메시지 확인하고 보라색 코드 복붙!

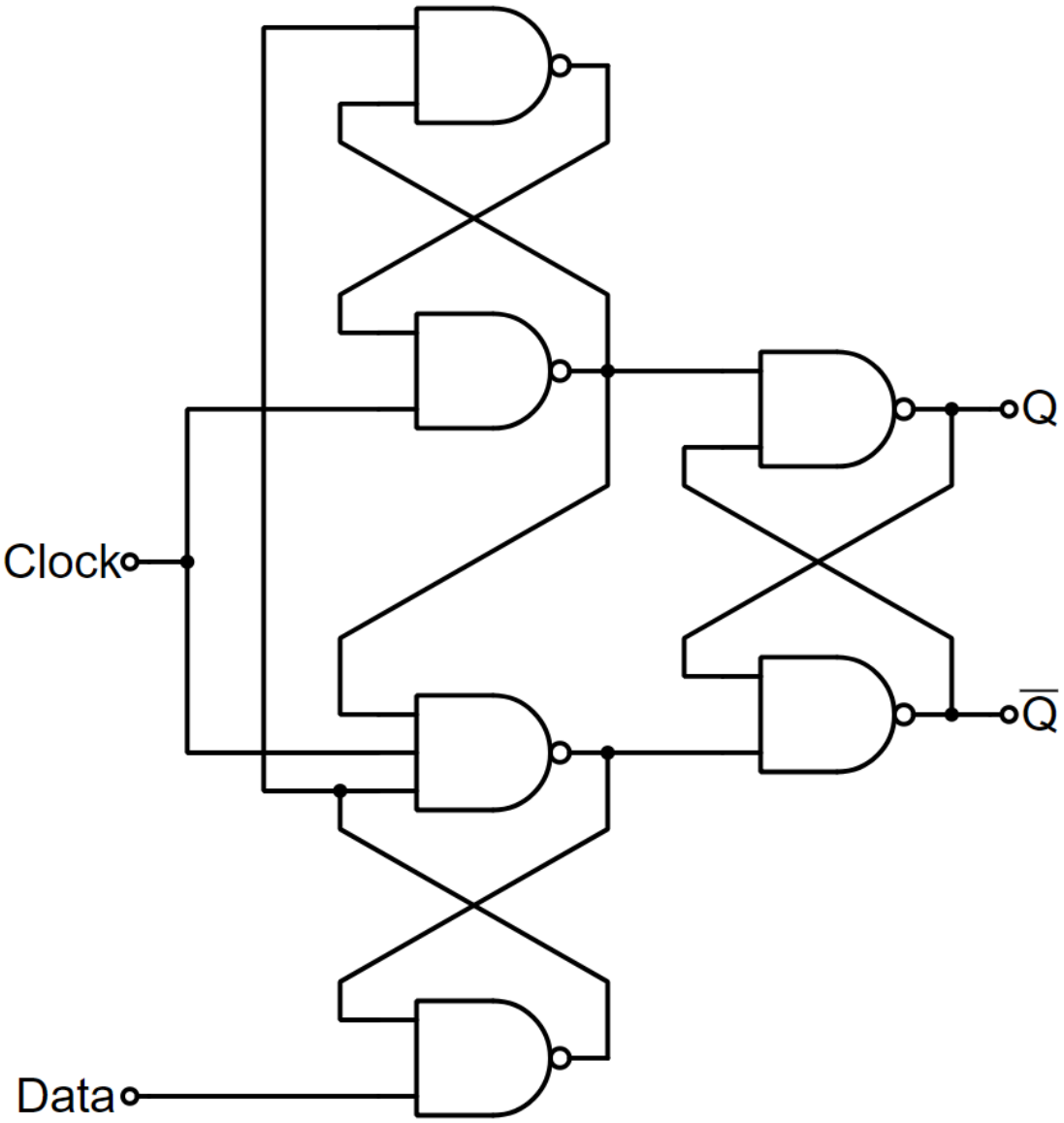
```
set_property PACKAGE_PIN G15 [get_ports R]
set_property PACKAGE_PIN P15 [get_ports S]
set_property PACKAGE_PIN M14 [get_ports Q]
set_property PACKAGE_PIN M15 [get_ports nQ]
set_property IOSTANDARD LVCMOS33 [get_ports S]
set_property IOSTANDARD LVCMOS33 [get_ports R]
set_property IOSTANDARD LVCMOS33 [get_ports Q]
set_property IOSTANDARD LVCMOS33 [get_ports nQ]
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
```

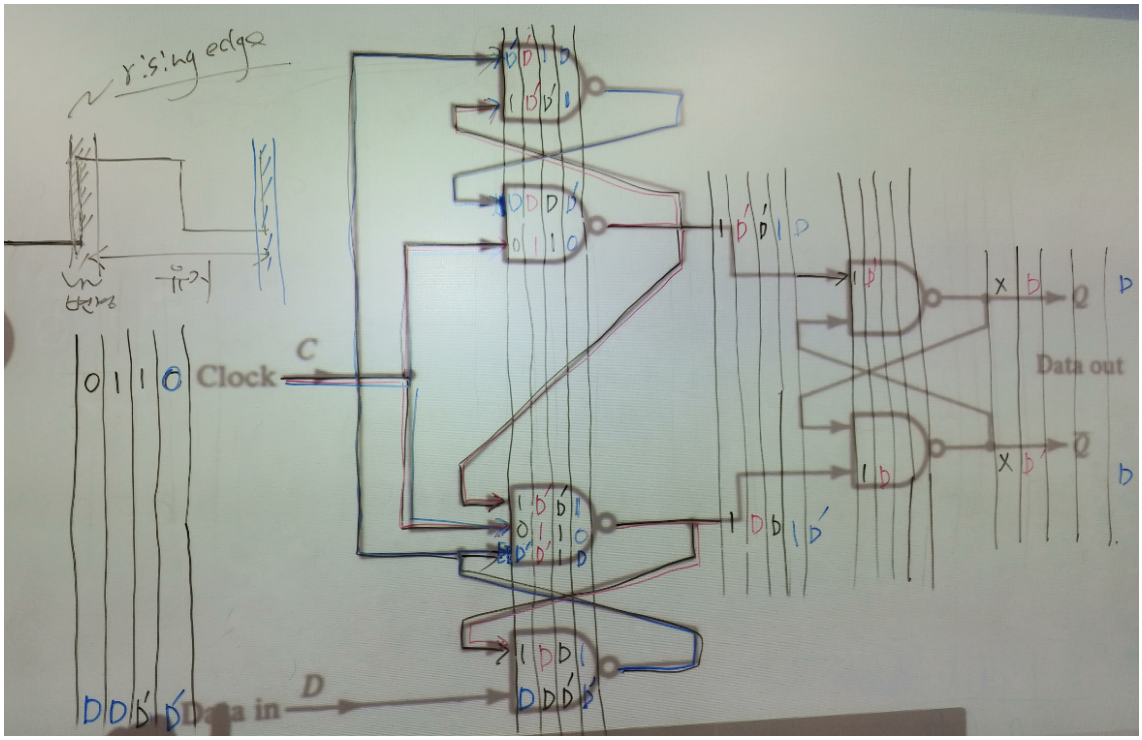
assign 문으로 Latch 구현해 보기

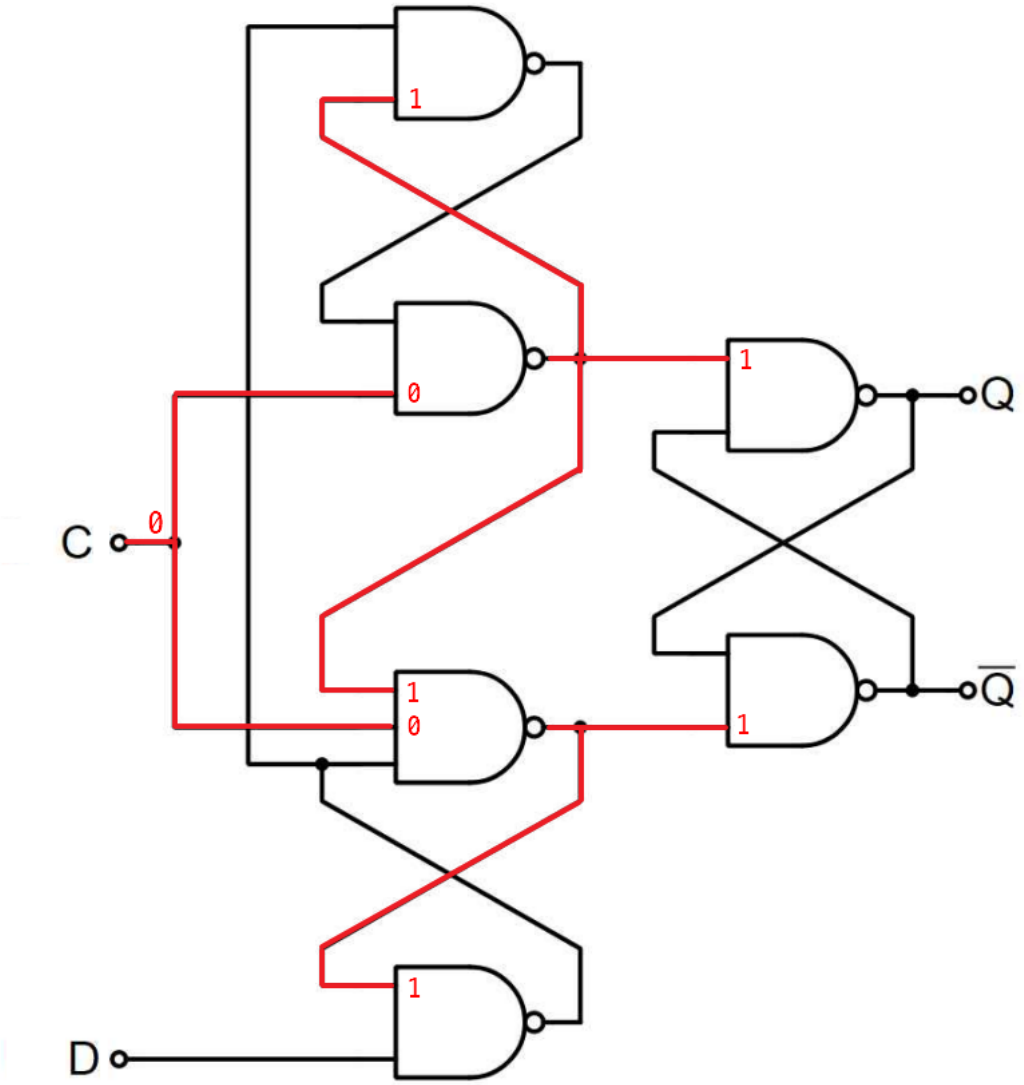
```
module MYLatch(
    input D,
    output Q,
    input En
    );

    assign Q=(En==1)? D: Q;

endmodule
```

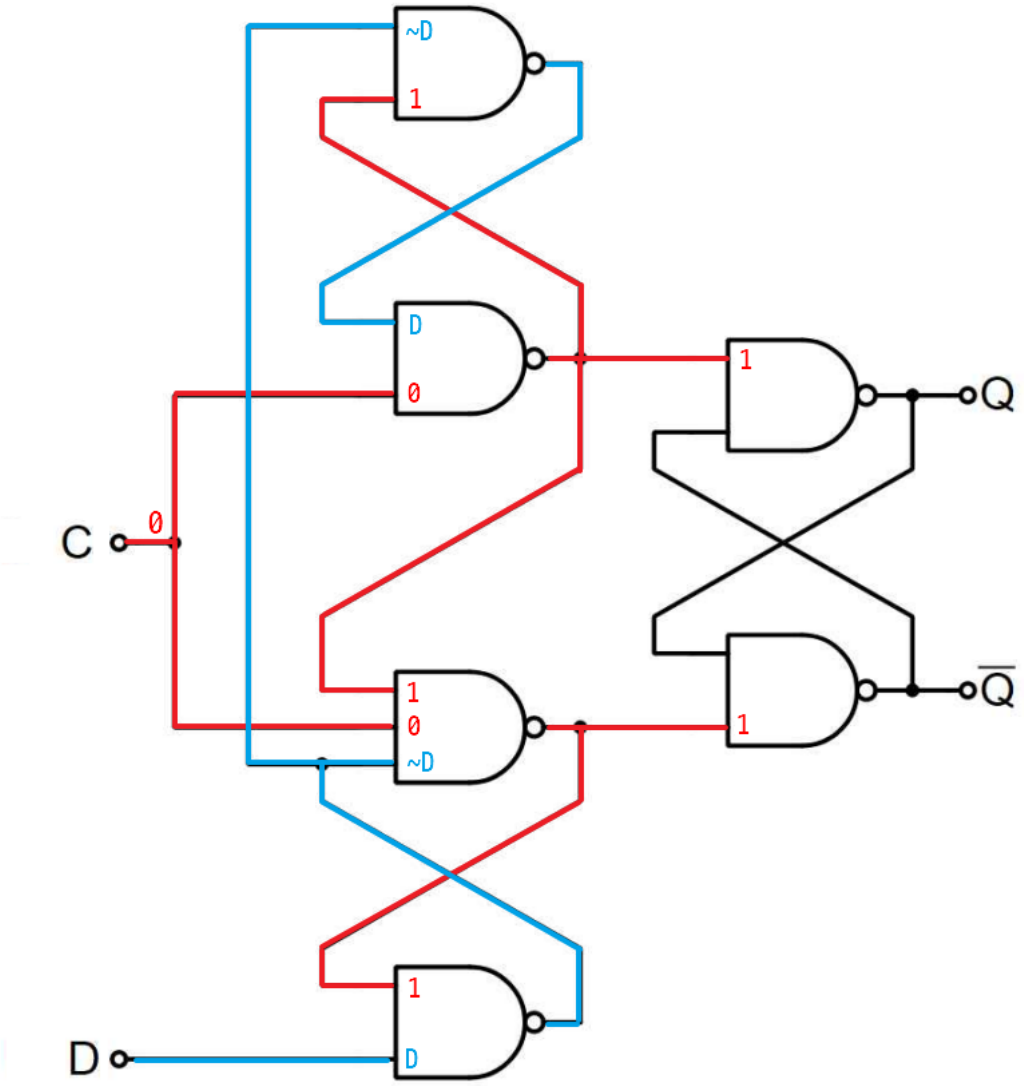
DFF

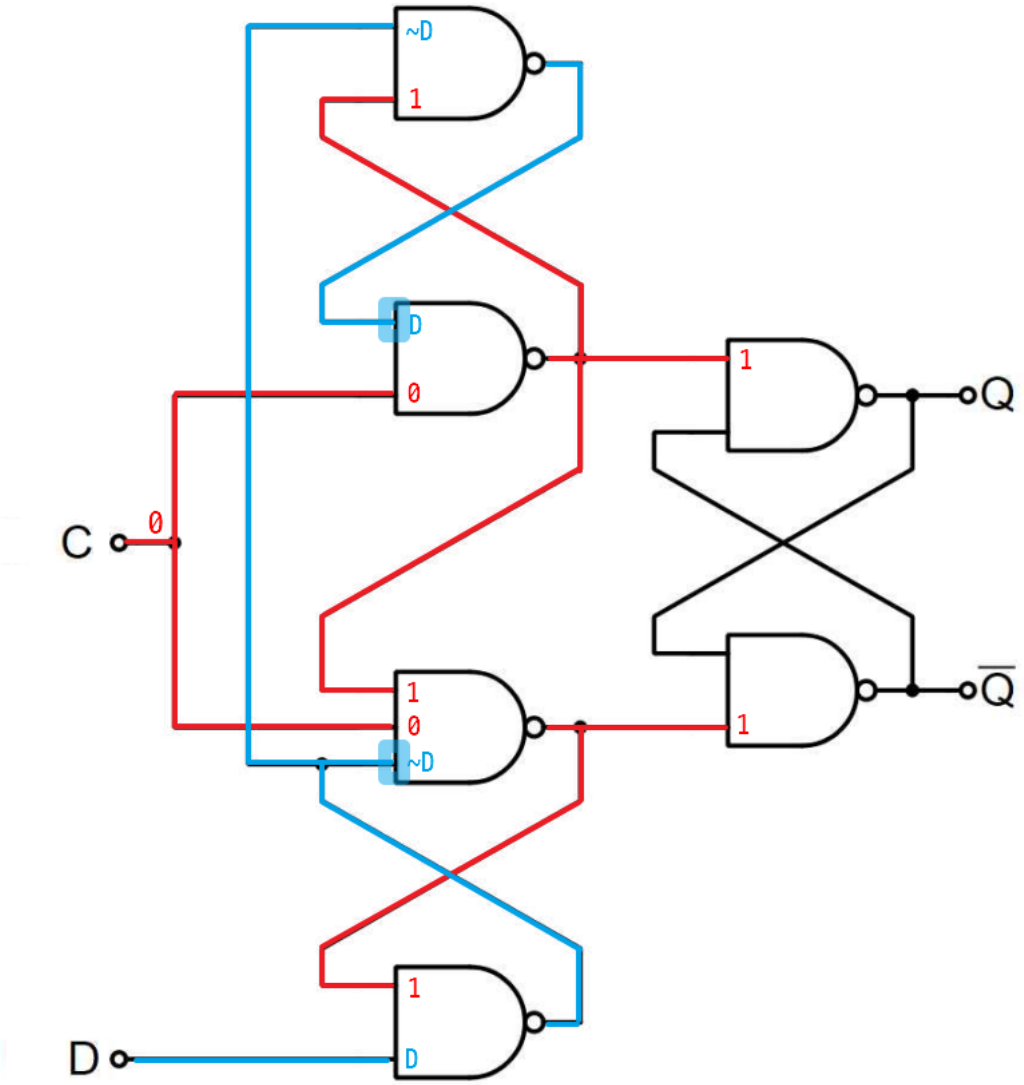


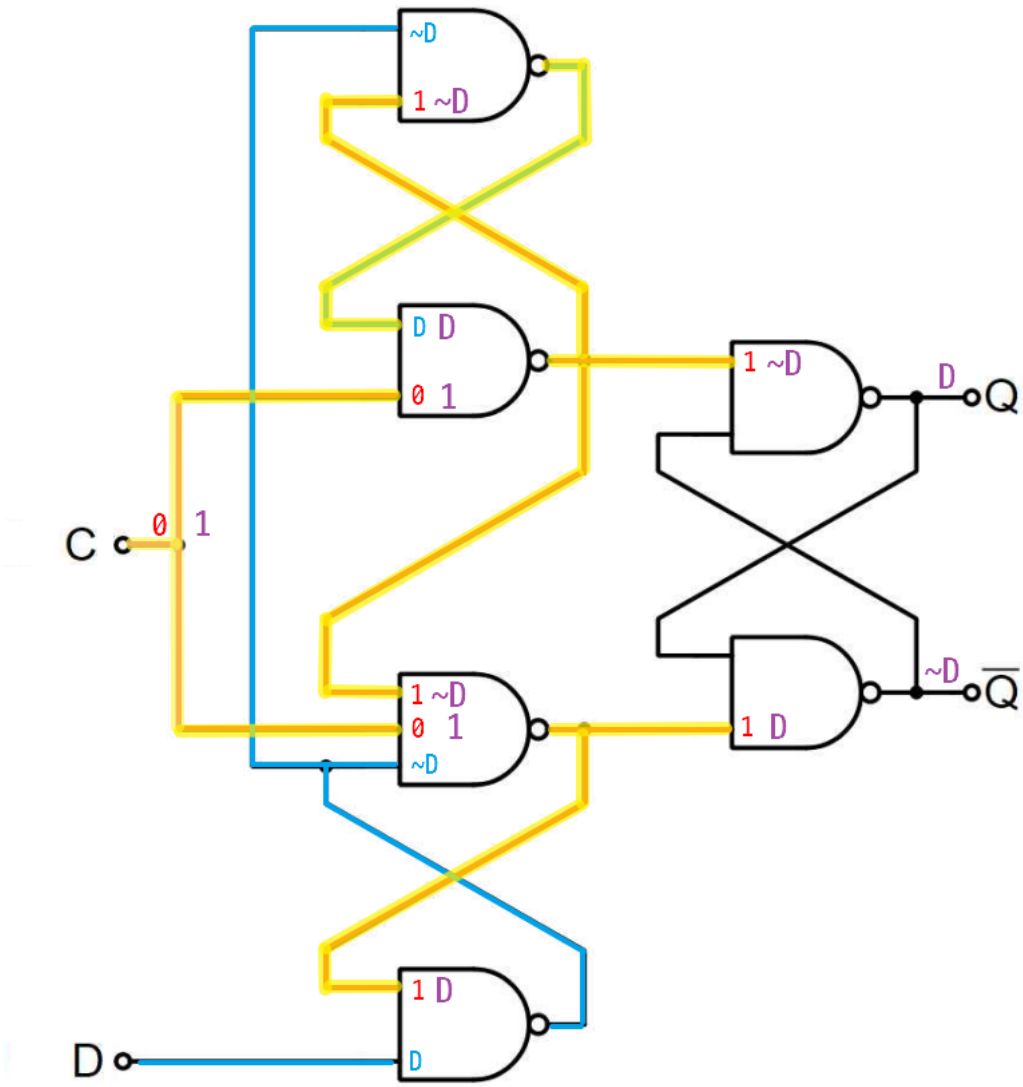

Clock
Data
Q
$\overline{Q}$

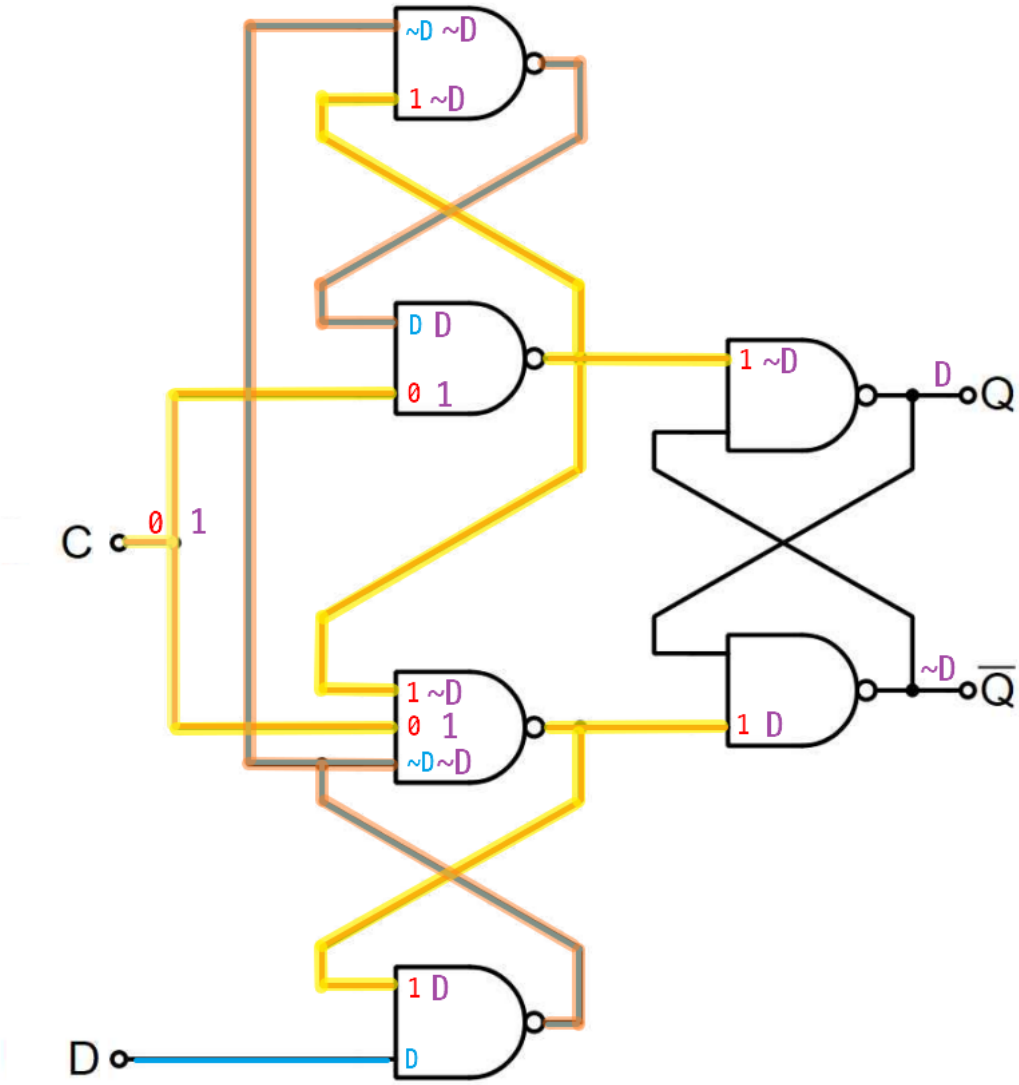

rising edge
Clock
C
Data in
D
Data out
Q

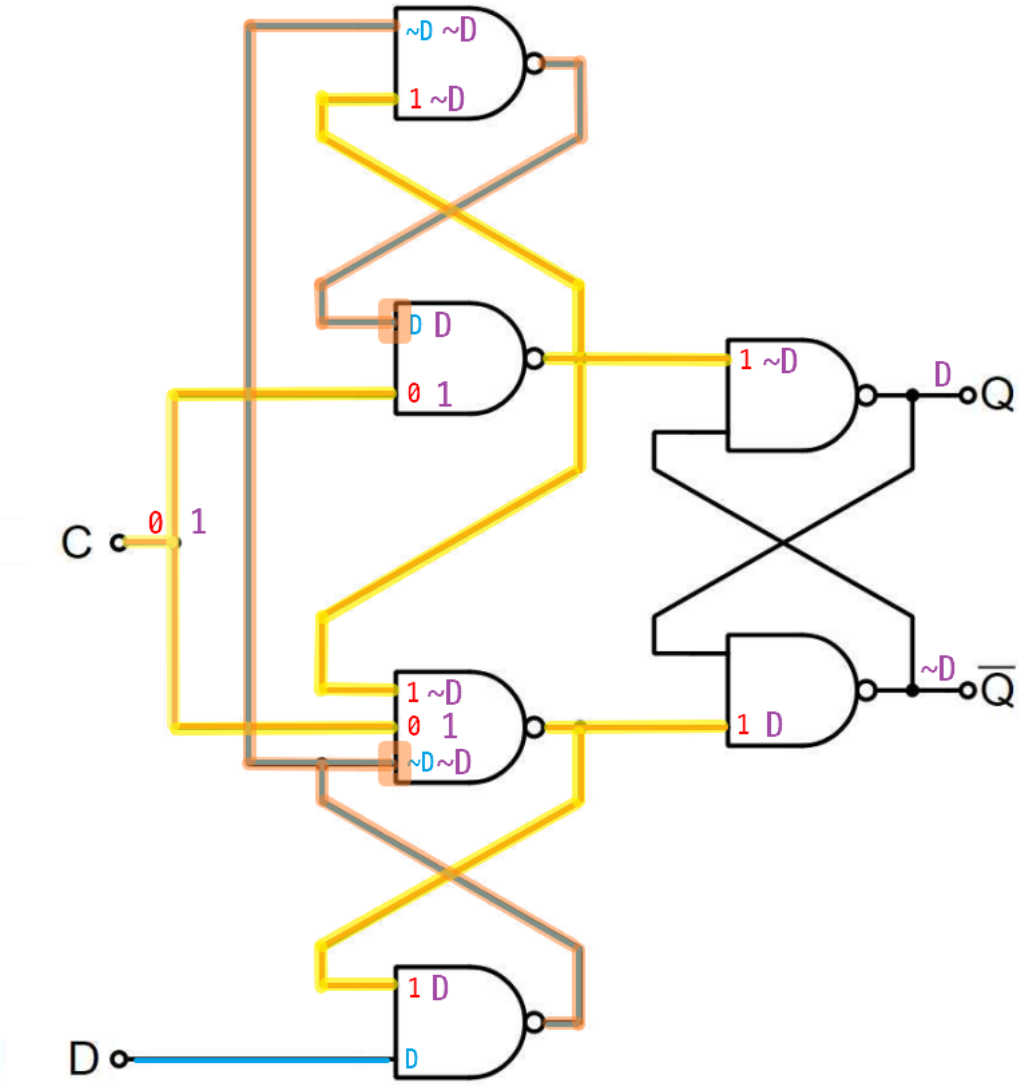

1
0
1
Q
C
0
1
0
1
$\overline{Q}$
1
D

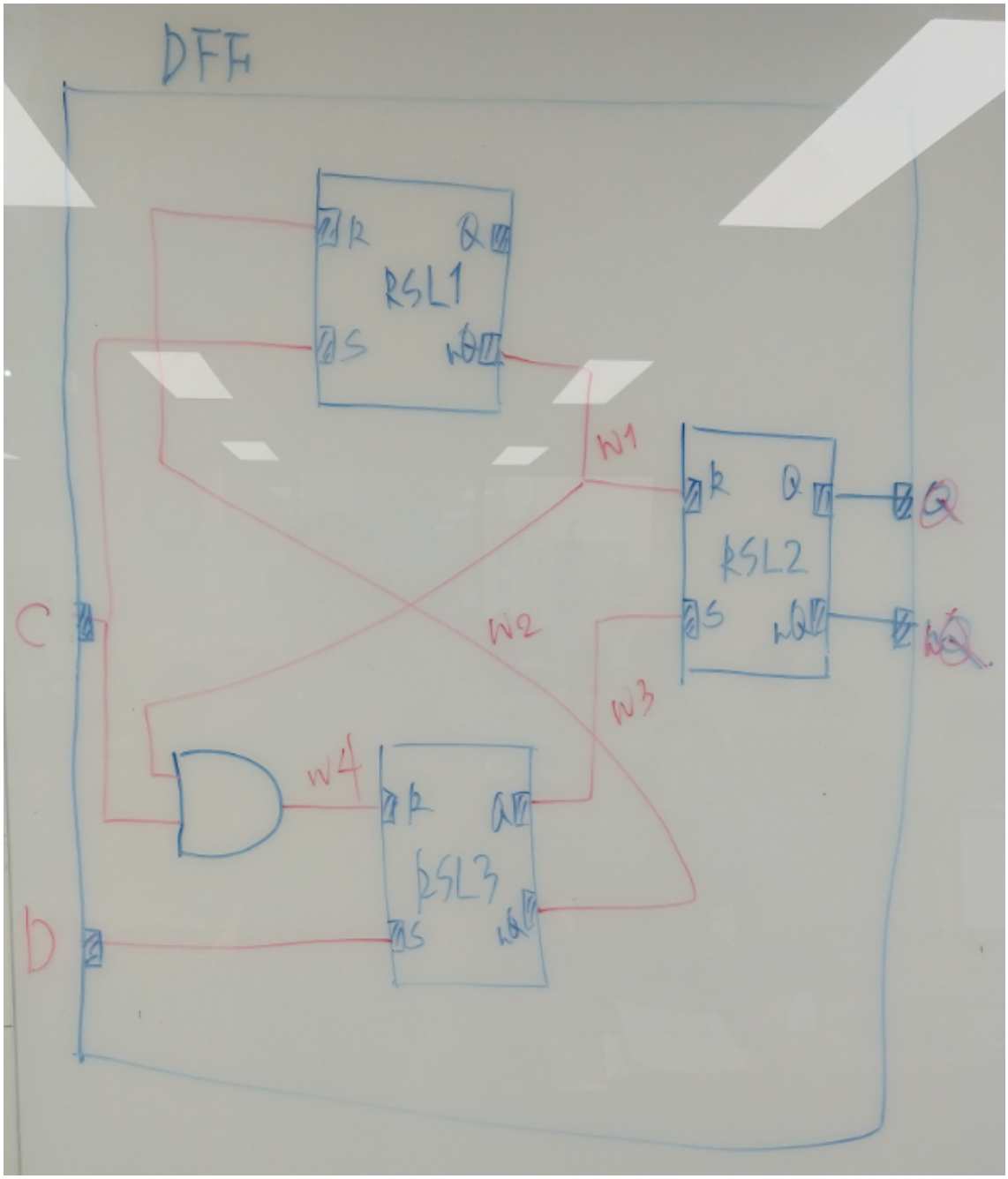

~D
1
D
0
1
Q
C
0
1
0
~D
1
$\overline{Q}$
D
1
D


~D
1
D
0
1
Q
C
0
1
0
~D
1
$\overline{Q}$
1
D


~D
1 ~D
D D
0 1
1 ~D
D
Q
C
0 1
1 ~D
0 1
~D
~D
Q
1 D
1 D
D


~D ~D
1 ~D
D D
0 1
1 ~D
D
Q
C
0 1
1 ~D
0 1
~D~D
~D
$\overline{Q}$
1 D
1 D
D
D


~D ~D
1 ~D
D D
0 1
1 ~D
D
Q
C
0 1
1 ~D
0 1
~D ~D
~D
Q
1 D
1 D
D



~(a & b & c) = ~((a & b) & b)

```
module DFF(
    input D,
    input C,
    output Q,
    output nQ
    );

    (* ALLOW_COMBINATORIAL_LOOPS = "true", KEEP = "true" *)
    wire w1, w2, w3, w4;

    assign w4 = w1 & C;
    RSLatch rsLatch_1(.R(w2), .S(C), .Q(), .nQ(w1));
    RSLatch rsLatch_2(.R(w1), .S(w3), .Q(Q), .nQ(nQ));
    RSLatch rsLatch_3(.R(w4), .S(D), .Q(w3), .nQ(w2));

endmodule
```

실제 사용 코드

```
module DFFR(
    input D,
    input C,
    output reg Q,
    output reg nQ
    );

    always @(posedge C)
    begin
        Q = D;
        nQ = ~D;
    end
endmodule
```

```
set_property PACKAGE_PIN M14 [get_ports Q]
set_property PACKAGE_PIN M15 [get_ports nQ]
set_property IOSTANDARD LVCMOS33 [get_ports nQ]
set_property IOSTANDARD LVCMOS33 [get_ports Q]

set_property PACKAGE_PIN G15 [get_ports D]
set_property PACKAGE_PIN P15 [get_ports C]
set_property IOSTANDARD LVCMOS33 [get_ports C]
set_property IOSTANDARD LVCMOS33 [get_ports D]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets C_IBUF]
```

오류 메시지 확인하고 보라색 코드 복붙!