

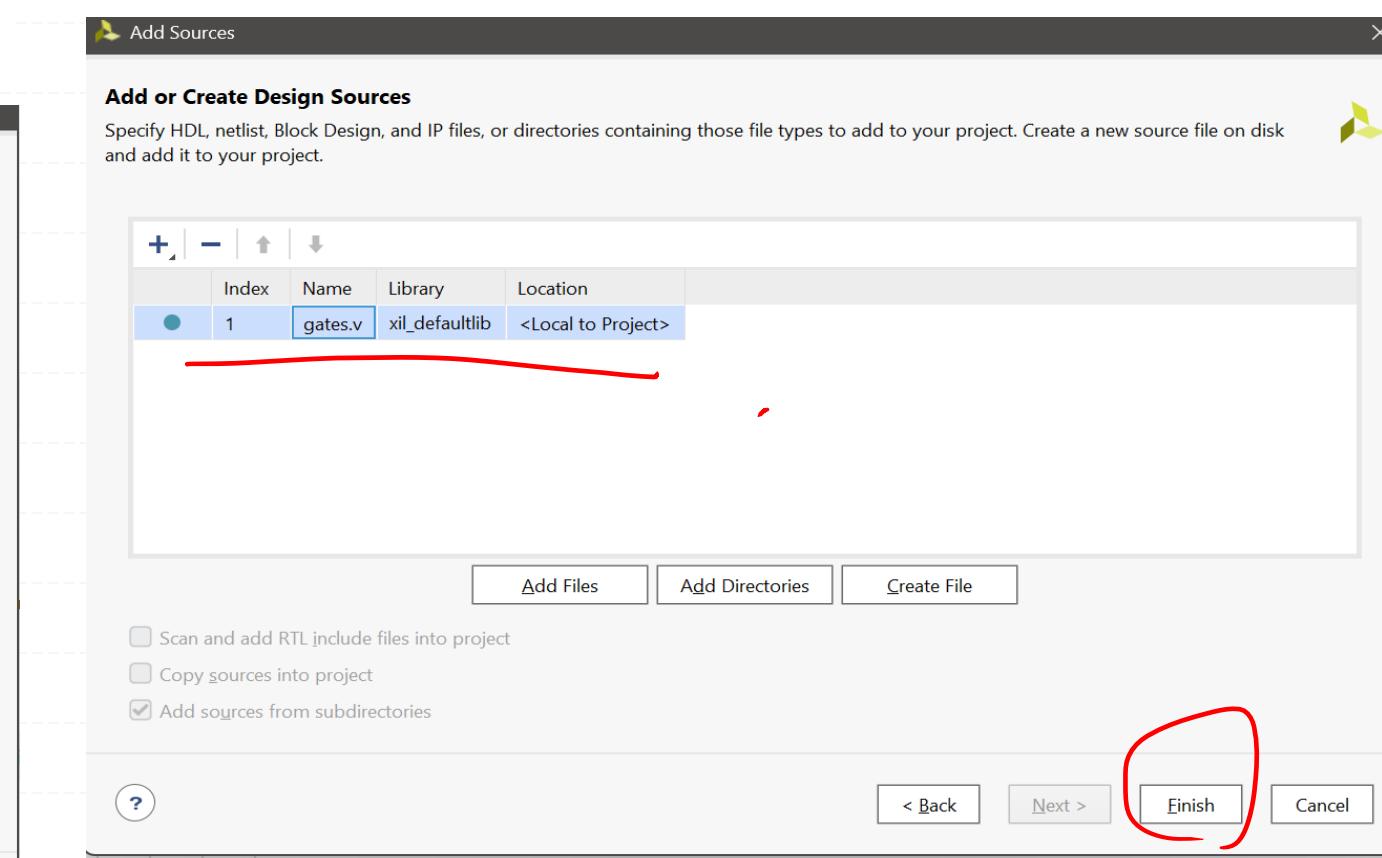
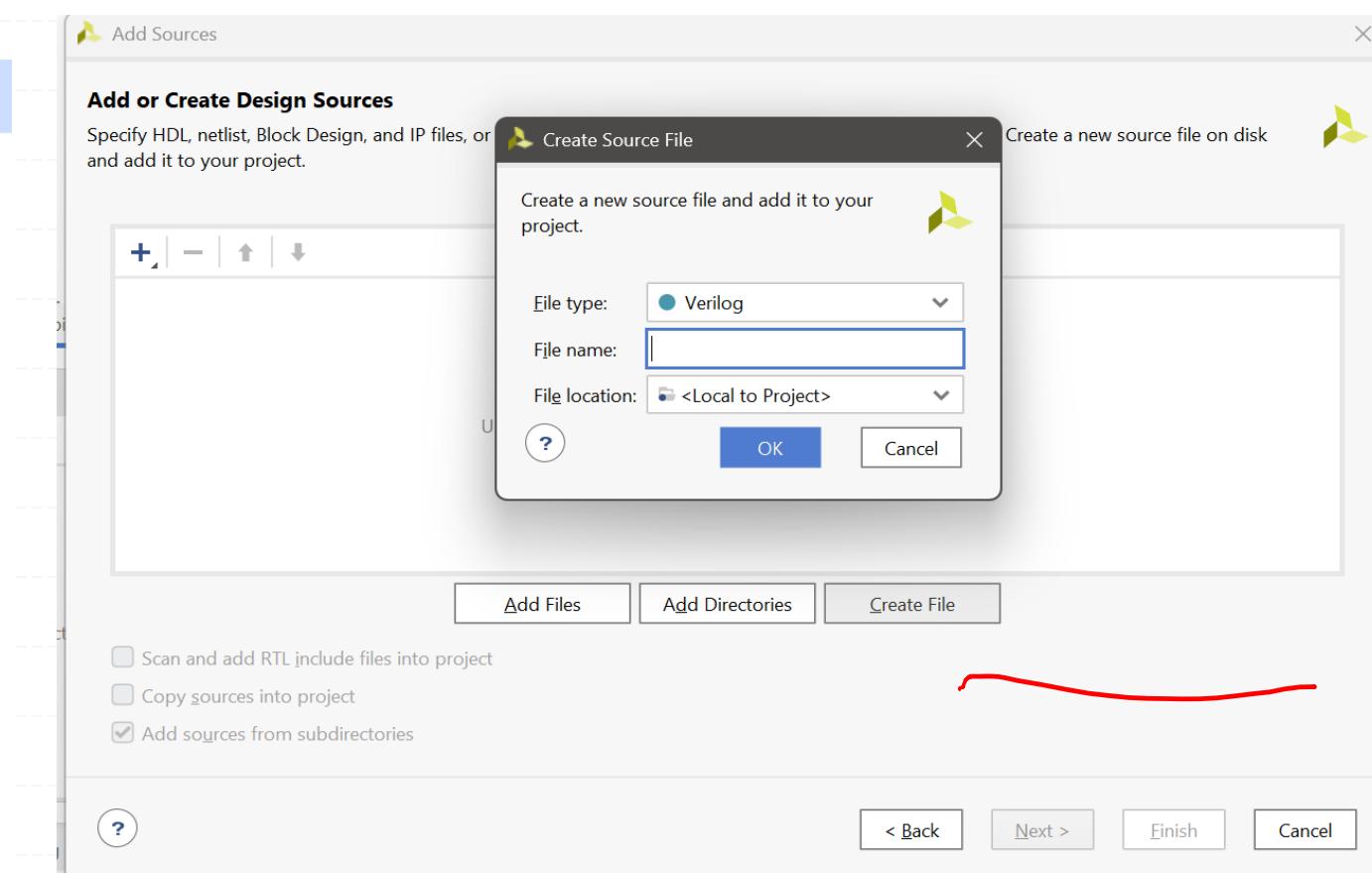
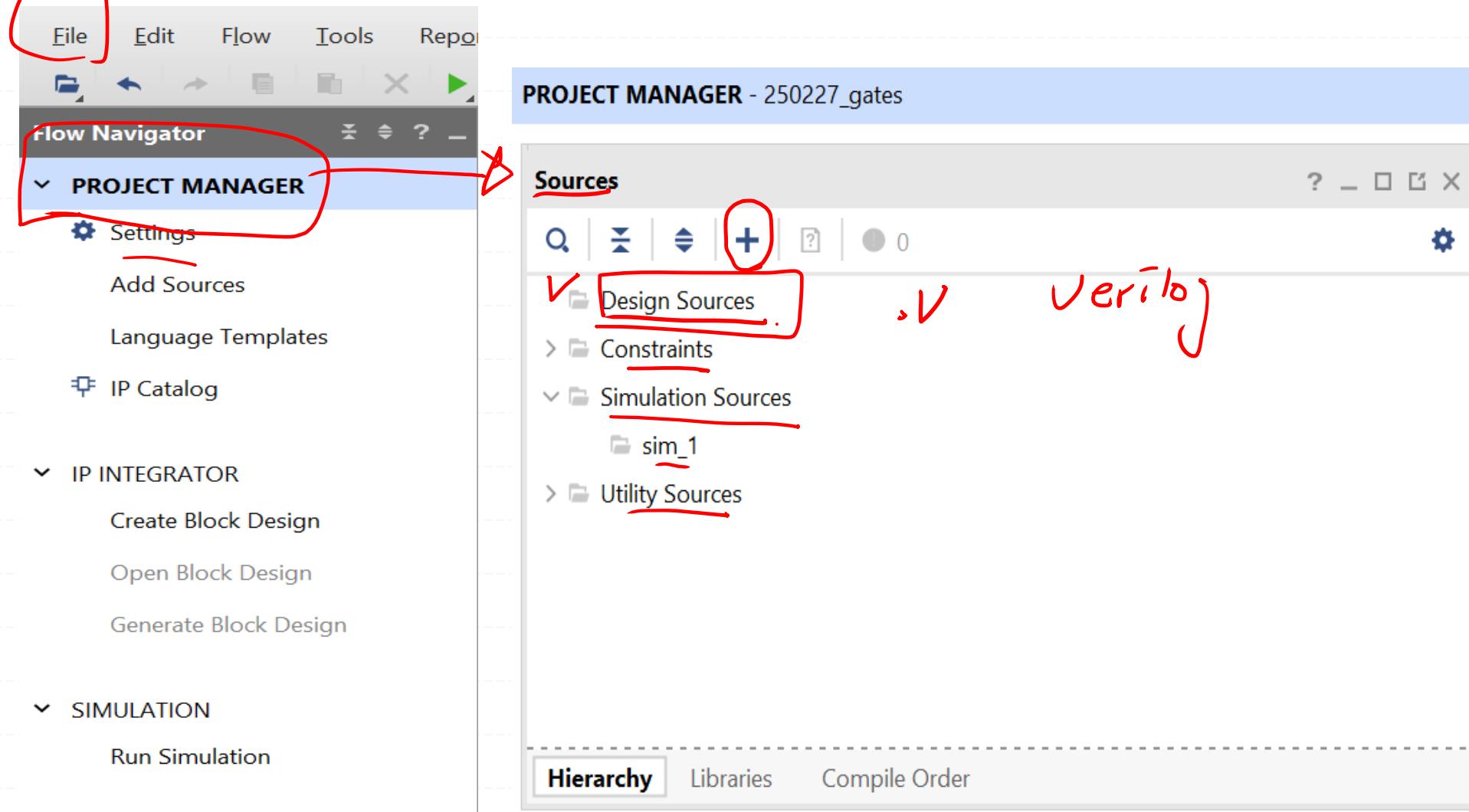
## 달력정보

달력		양음력변환			전역일계산		
		2025.03					
일	월	화	수	목	금	토	
23	24	25	26	27	28	1	삼일절
				Q는			
2	3 대체 휴일 납세자의 날	4	5 경칩	6	7	8 국제 여성... 3·8 민주의...	
9	10	11	12	13	14 음 2.15 발포(mini)	15 각주 10명	
16	17	18	19 상공의 날	20 춘분 국제 행복...	21 암예방의 날 발포(mini).	22 세계 물의...	
23 국제 강아...	24	25	26	27	28 서해수호...	29 음 3.1	
30	31 발포 b2t,,	1	2	3	4	5	

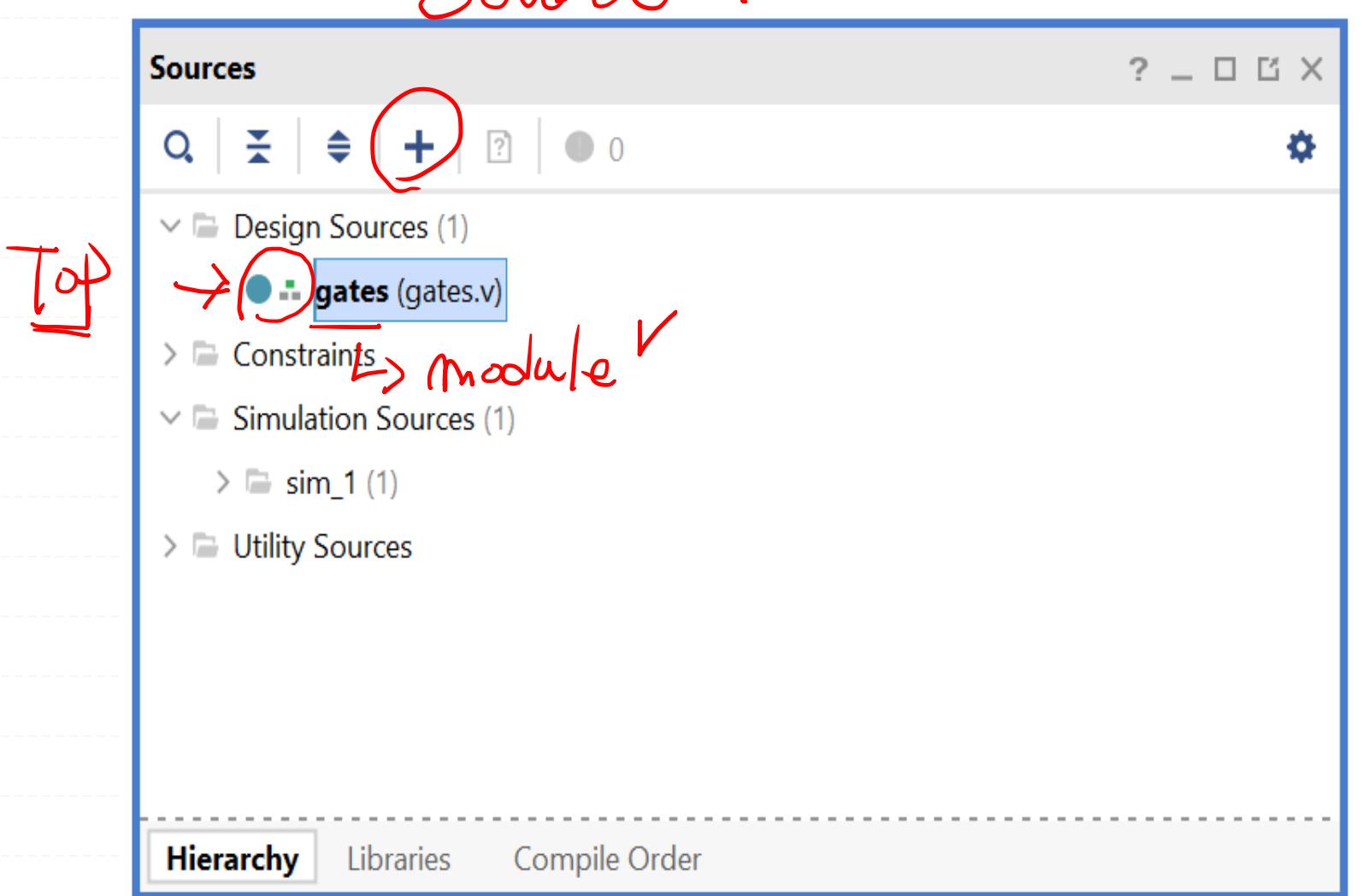
821 1시간

ALU: Adder, FND.

bus 10 Interface  
전시제어 JART - FIFO



Source 추가



Top 1st module



종작시간 단위

→ 1 `timescale 1ns / 1ps  
2  
3 module gates()  
4  
5 ;  
6 endmodule

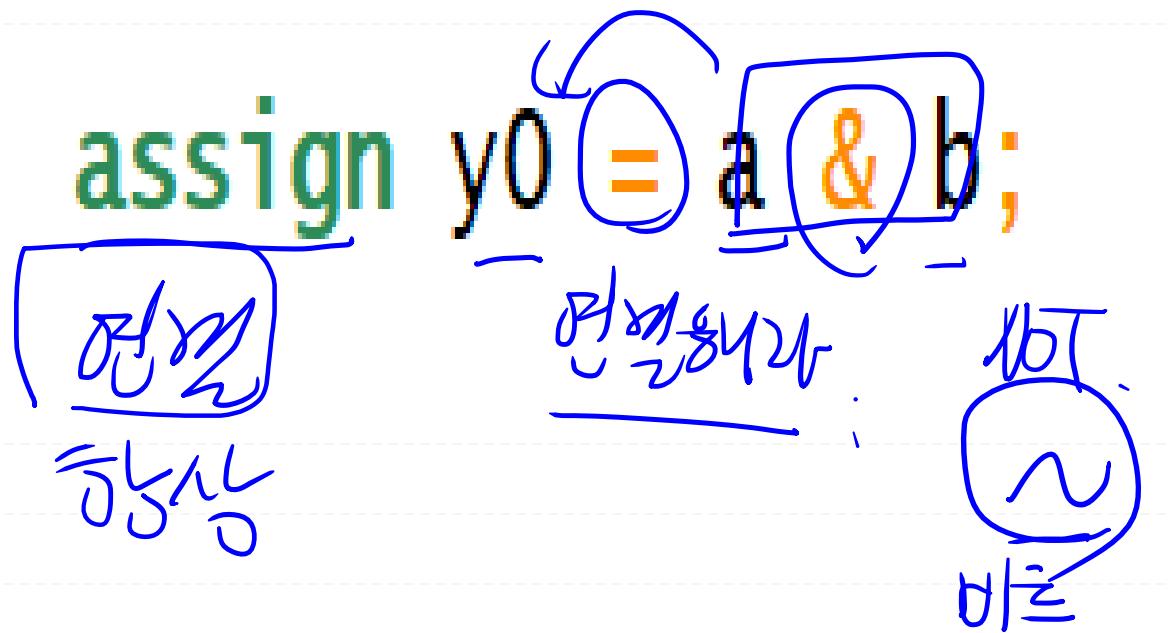
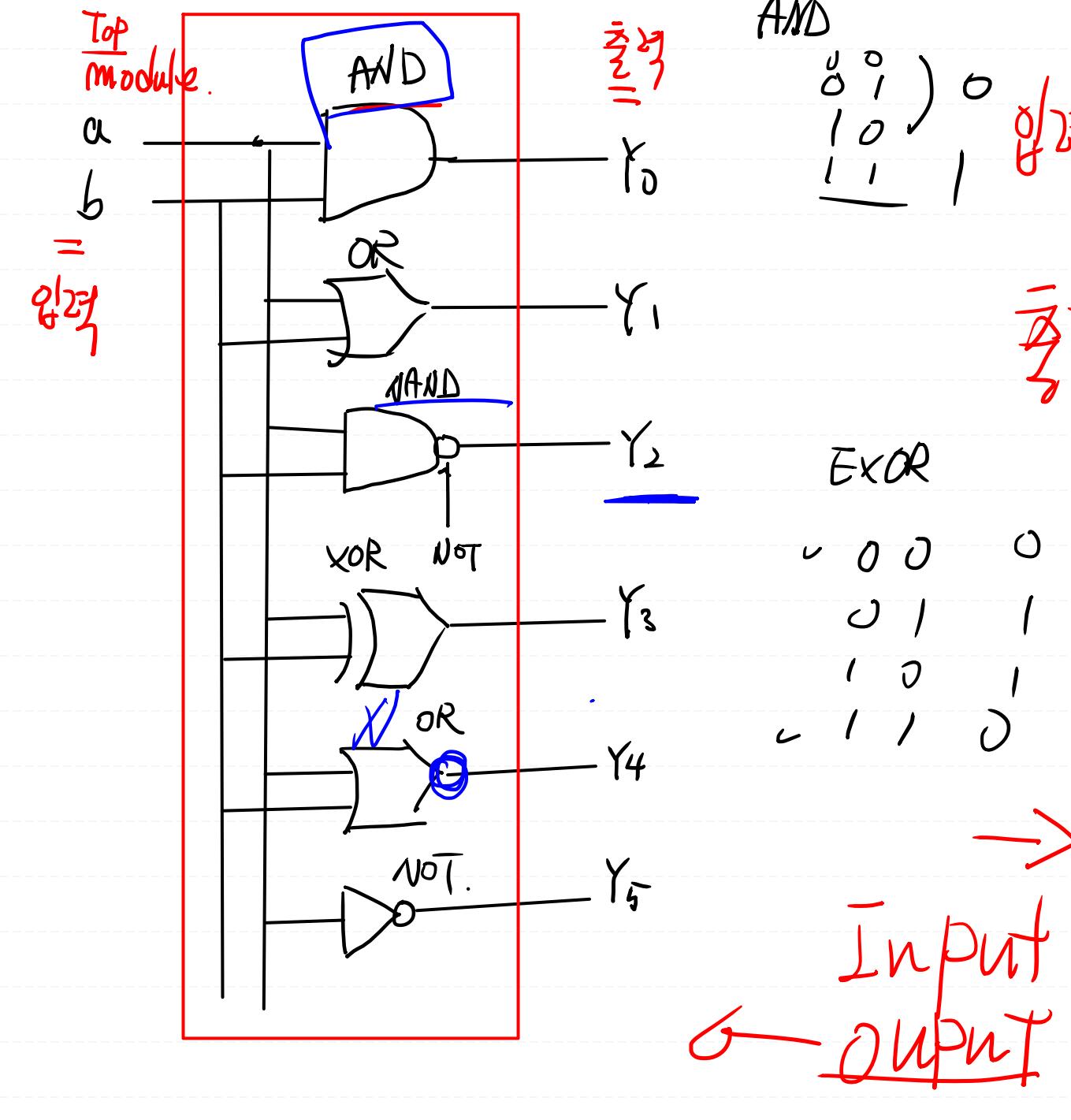
$$\mu : 10^{-6}$$
$$m : 10^{-3}$$

1ns / ps

1 ns / 10 ps

1 `timescale 1ns / 1ps

2 ✓  
3 module gates()  
4  
5 endmodule



AND  

$$\begin{array}{c} \textcircled{0} \\ \textcircled{1} \\ \textcircled{0} \\ \textcircled{1} \end{array} \rightarrow \begin{array}{c} \textcircled{0} \\ \textcircled{1} \\ \textcircled{0} \\ \textcircled{1} \end{array}$$
  
 입력  
 출력  
 list

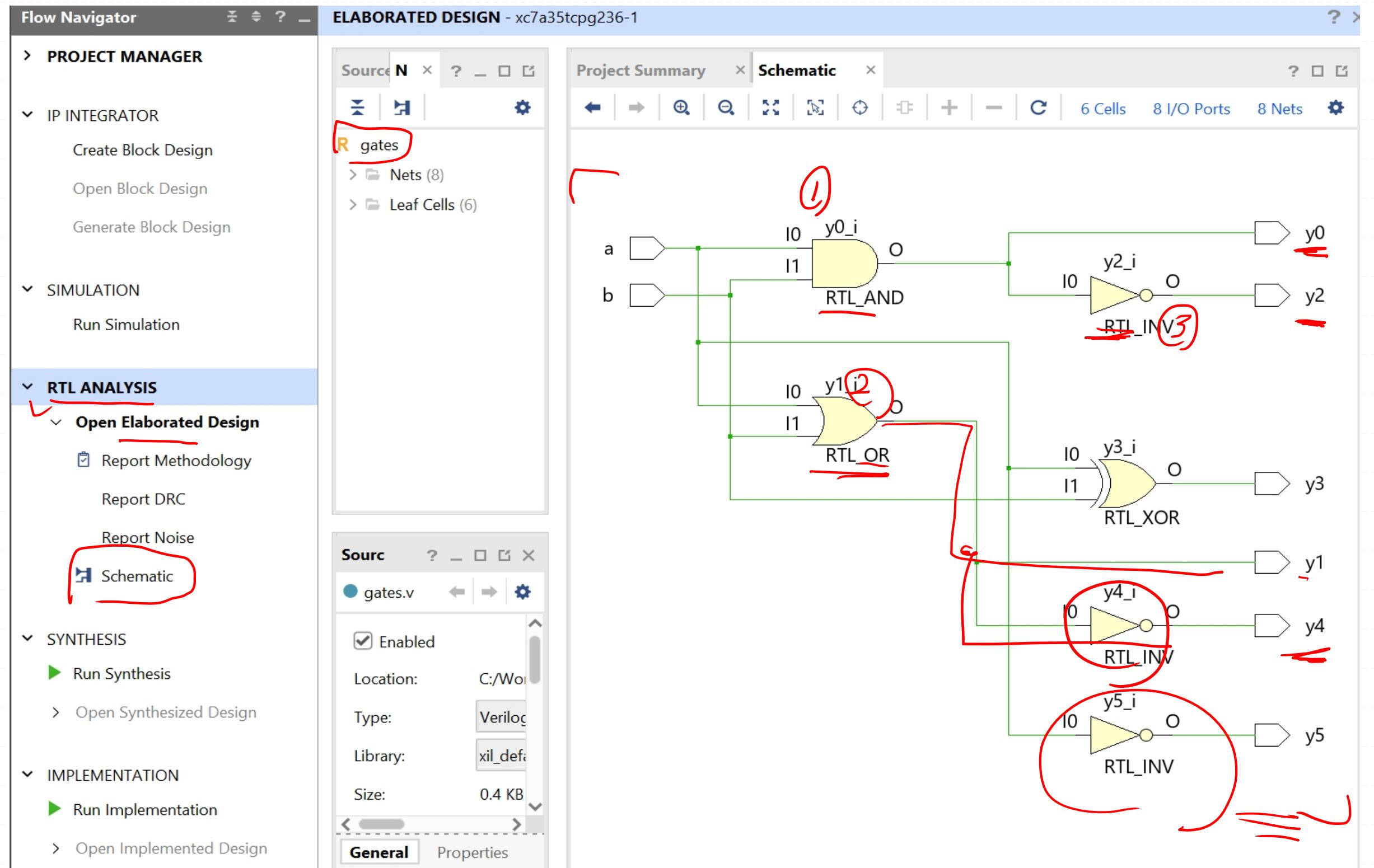
ExOR  

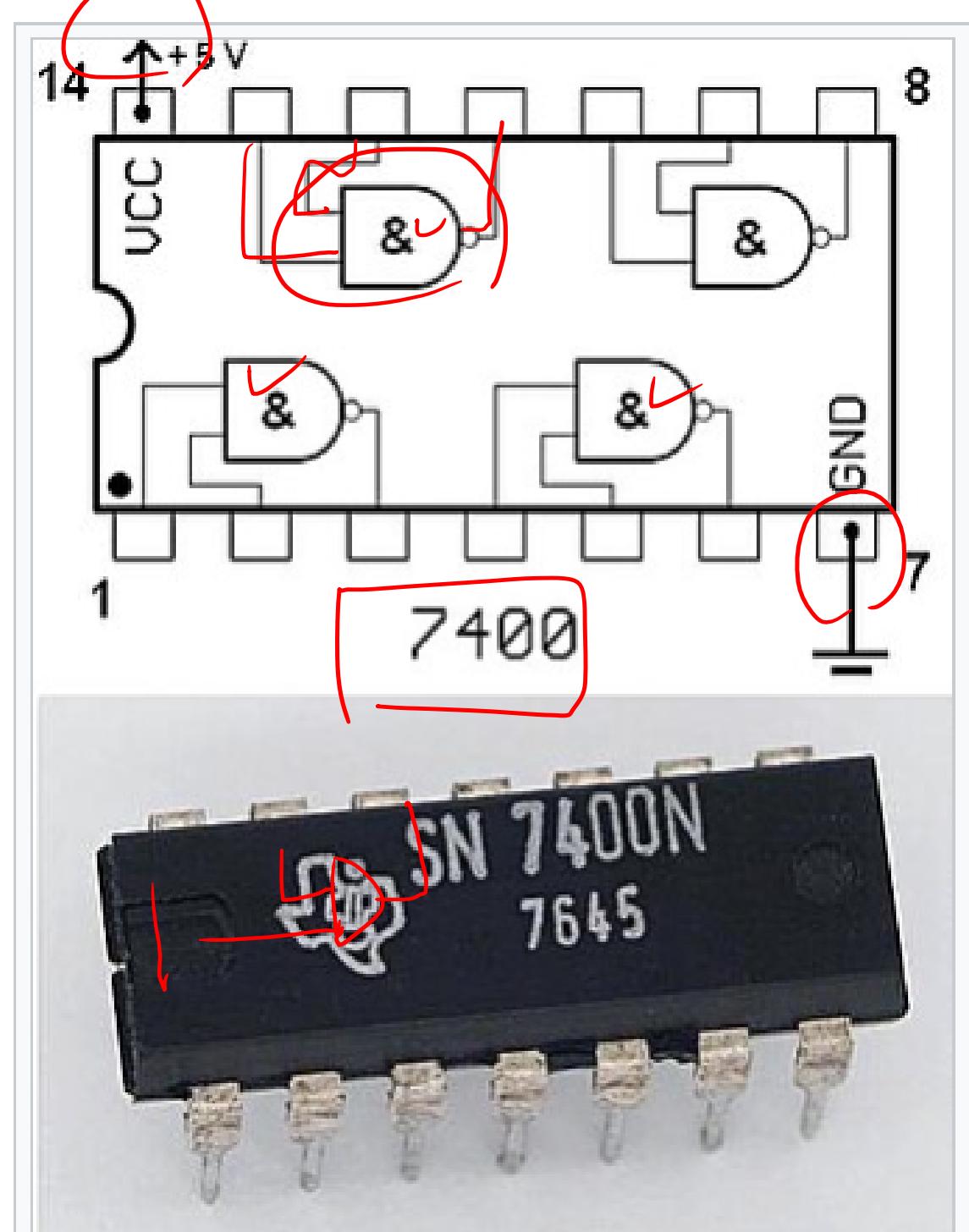
$$\begin{array}{ccccc} \textcircled{0} & \textcircled{0} & \textcircled{0} & \textcircled{1} & \textcircled{1} \\ \textcircled{0} & \textcircled{1} & \textcircled{1} & \textcircled{0} & \textcircled{0} \\ \textcircled{1} & \textcircled{0} & \textcircled{1} & \textcircled{1} & \textcircled{0} \\ \textcircled{1} & \textcircled{1} & \textcircled{0} & \textcircled{0} & \textcircled{1} \end{array}$$
  
 입력  
 출력  
 list

3. module gates( // top module  
 4. input a,  
 5. input b,  
 6. output y0,  
 7. output y1,  
 8. output y2,  
 9. output y3,  
 10. output y4,  
 11. output y5  
 12. )  
 13.  
 14. endmodule

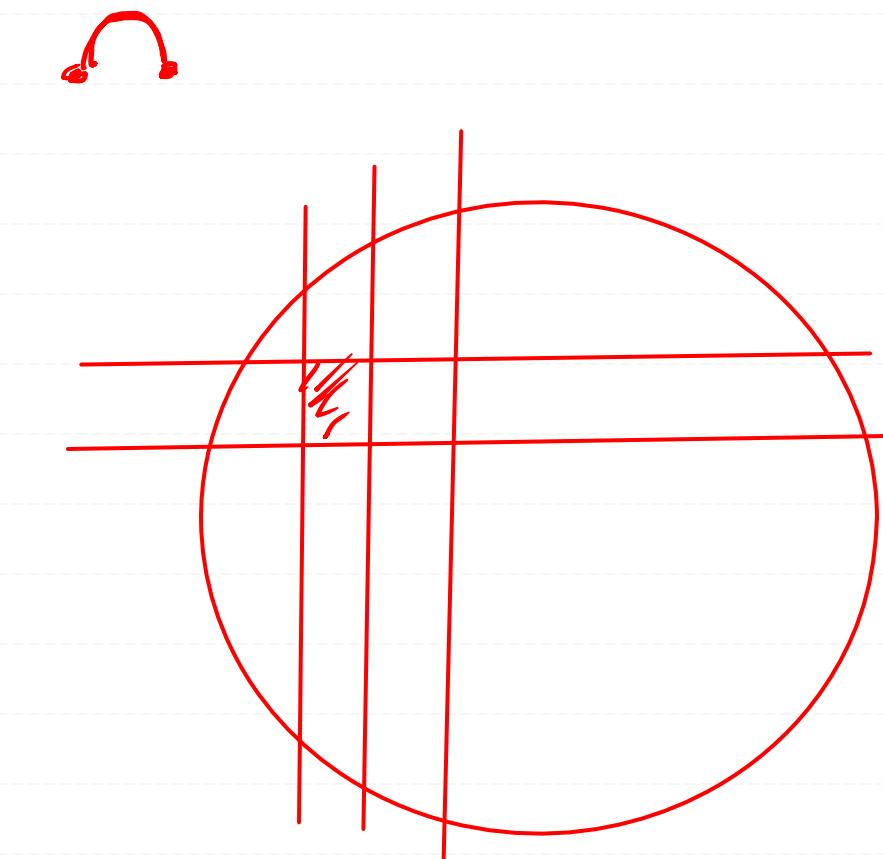
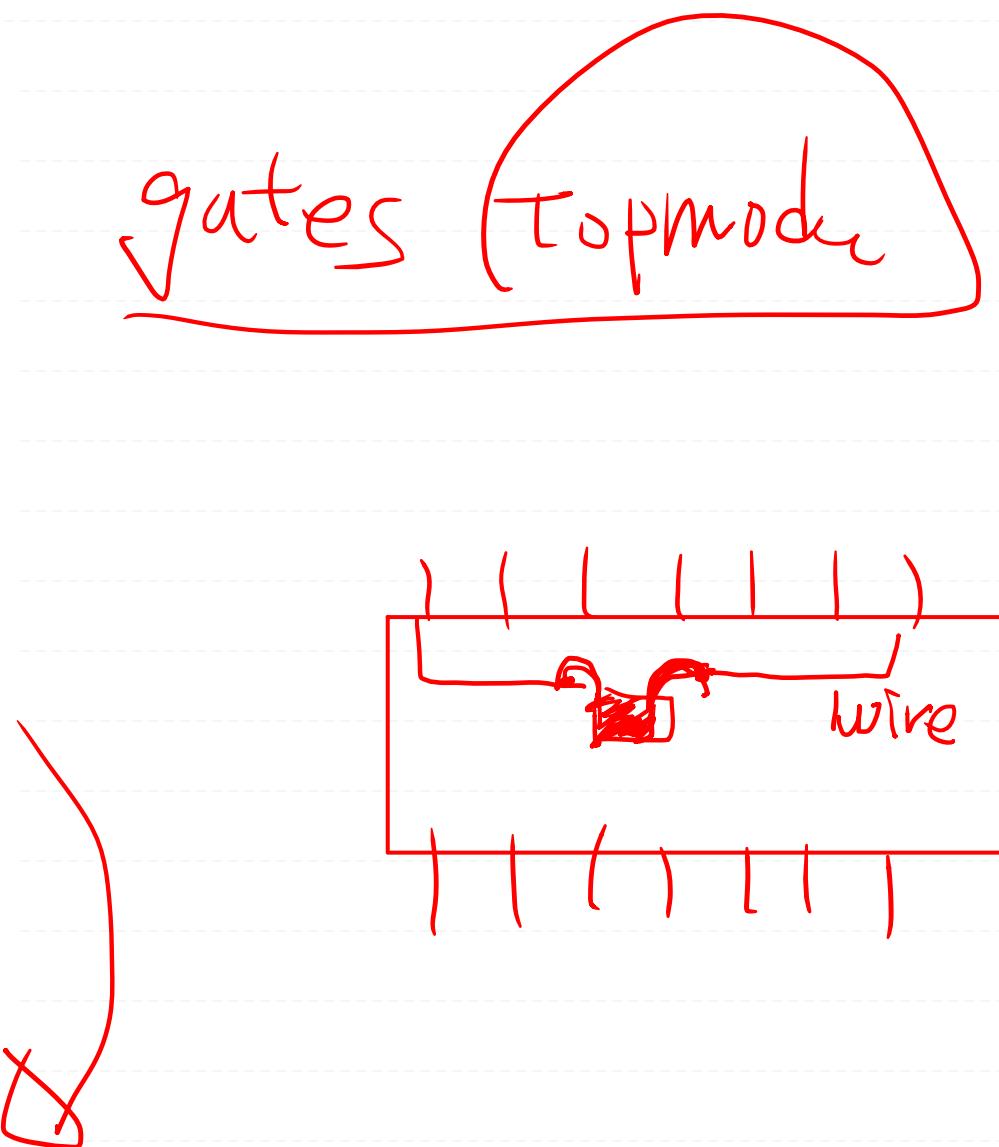
Input  $\wedge$  a<sub>j</sub>  
wire  $\Rightarrow$  default  
input wife a  
input a

Input a  
! : 2가지 것. ~~4가지~~ 3bit.





The SN7400N chip contains



Run Simulation

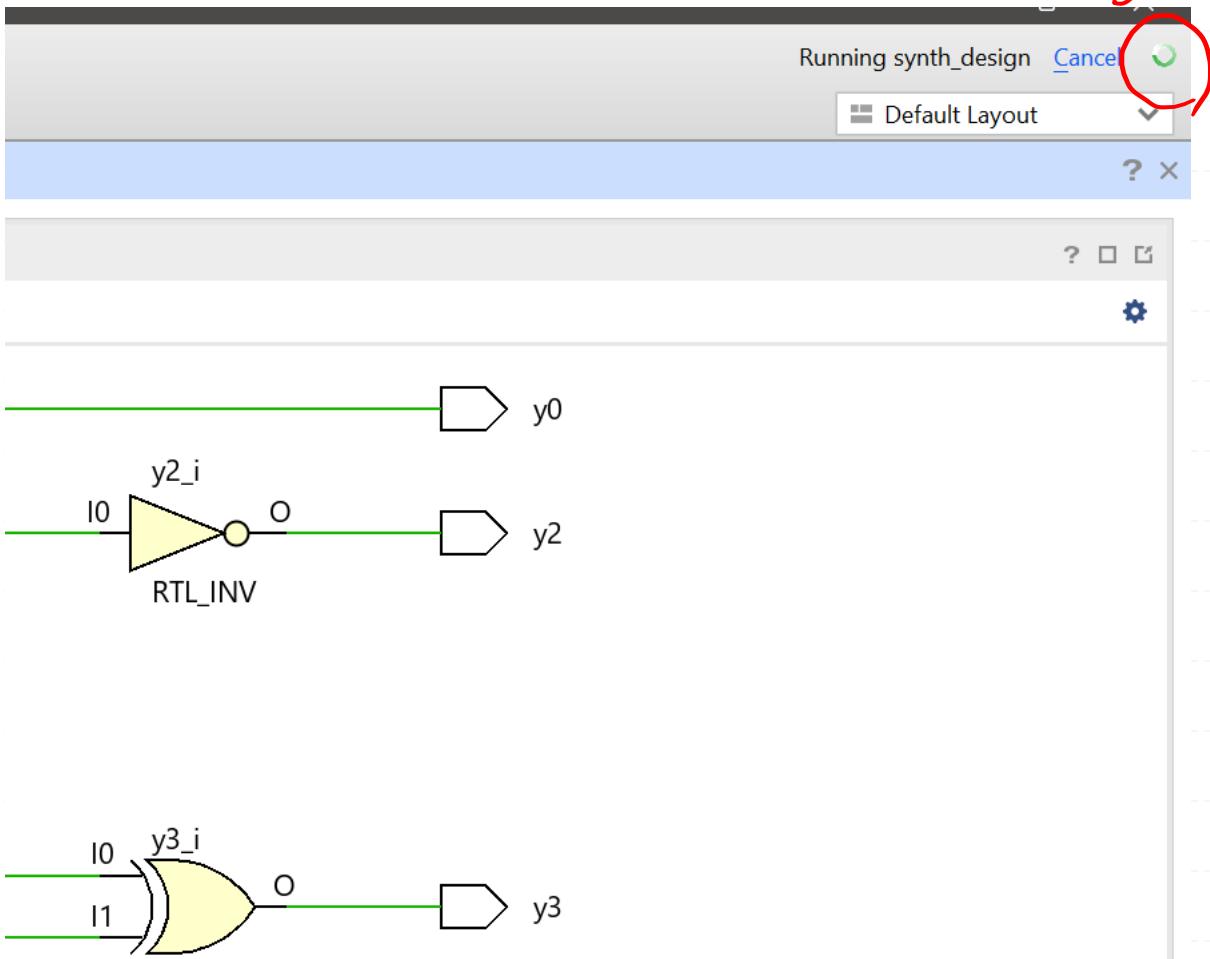
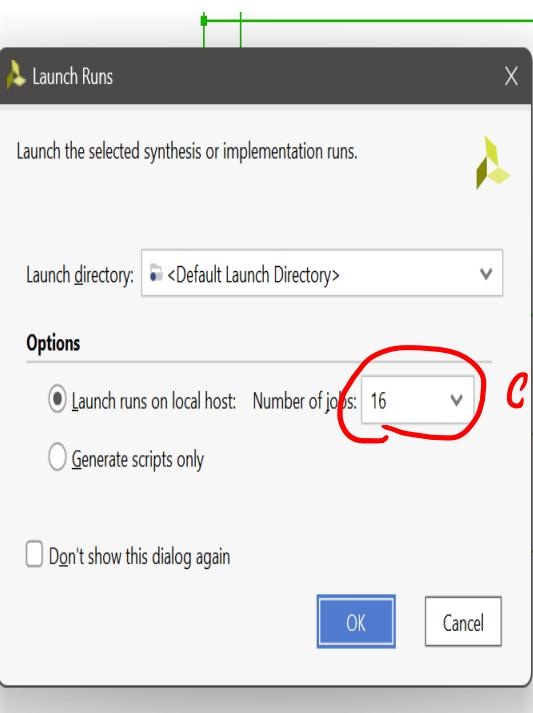
RTL ANALYSIS

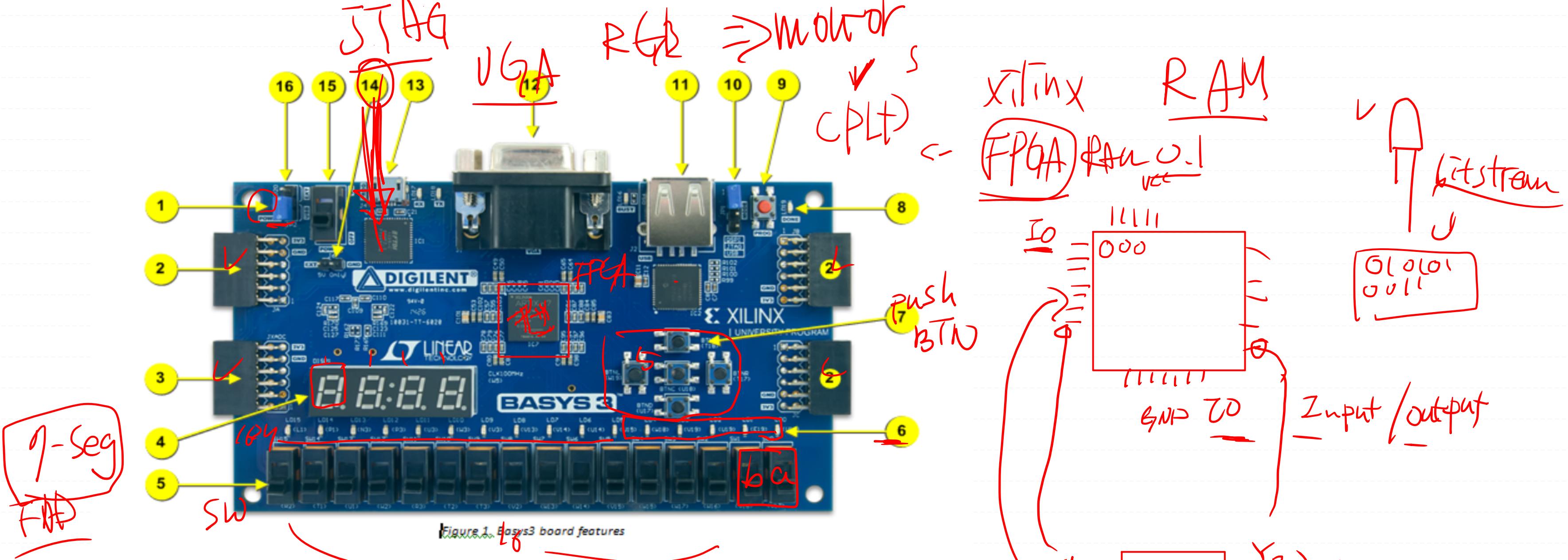
- Open Elaborated Design
- Report Methodology
- Report DRC
- Report Noise
- Schematic

SYNTHESIS

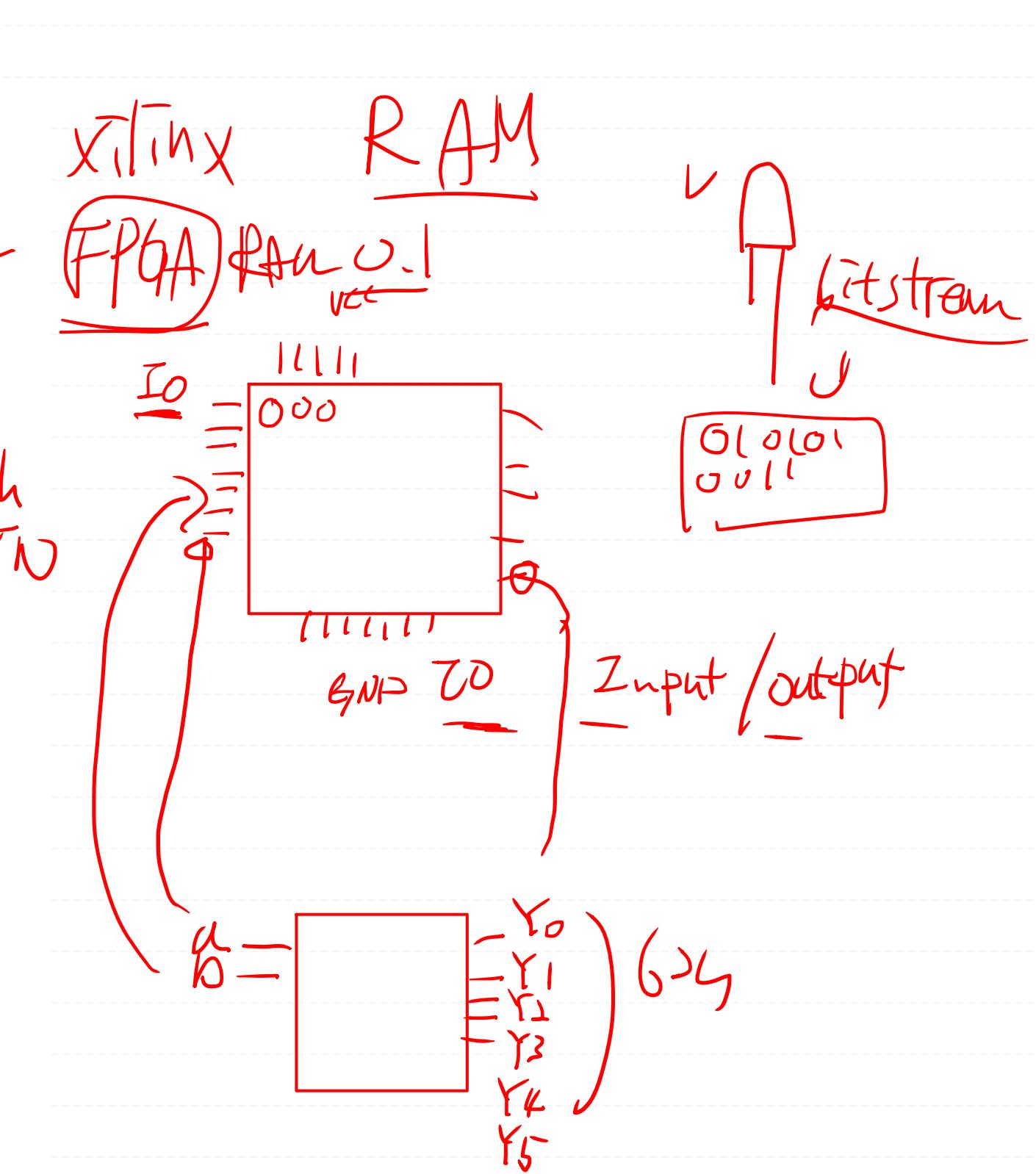
- Run Synthesis
- Open Synthesized Design

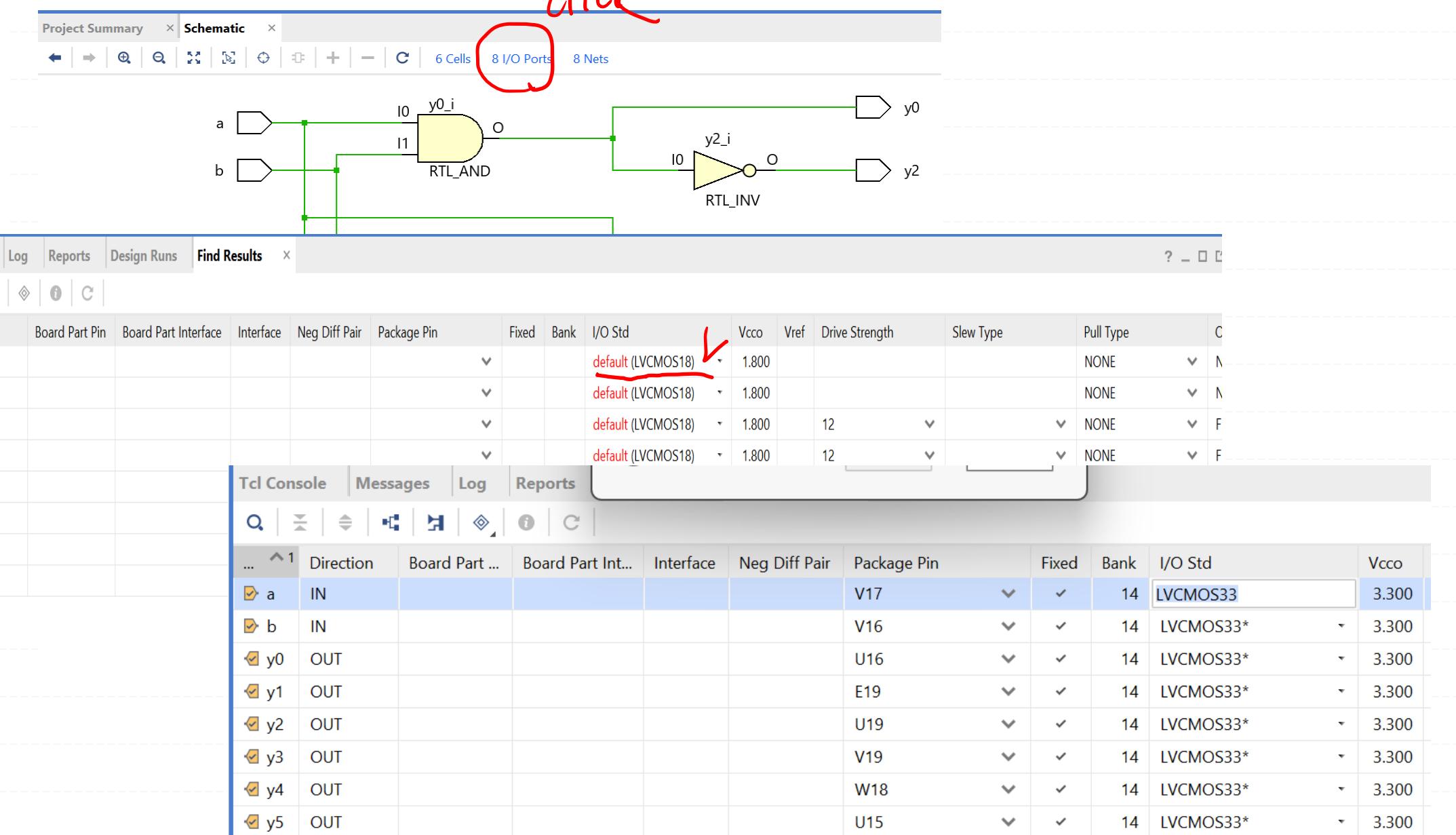
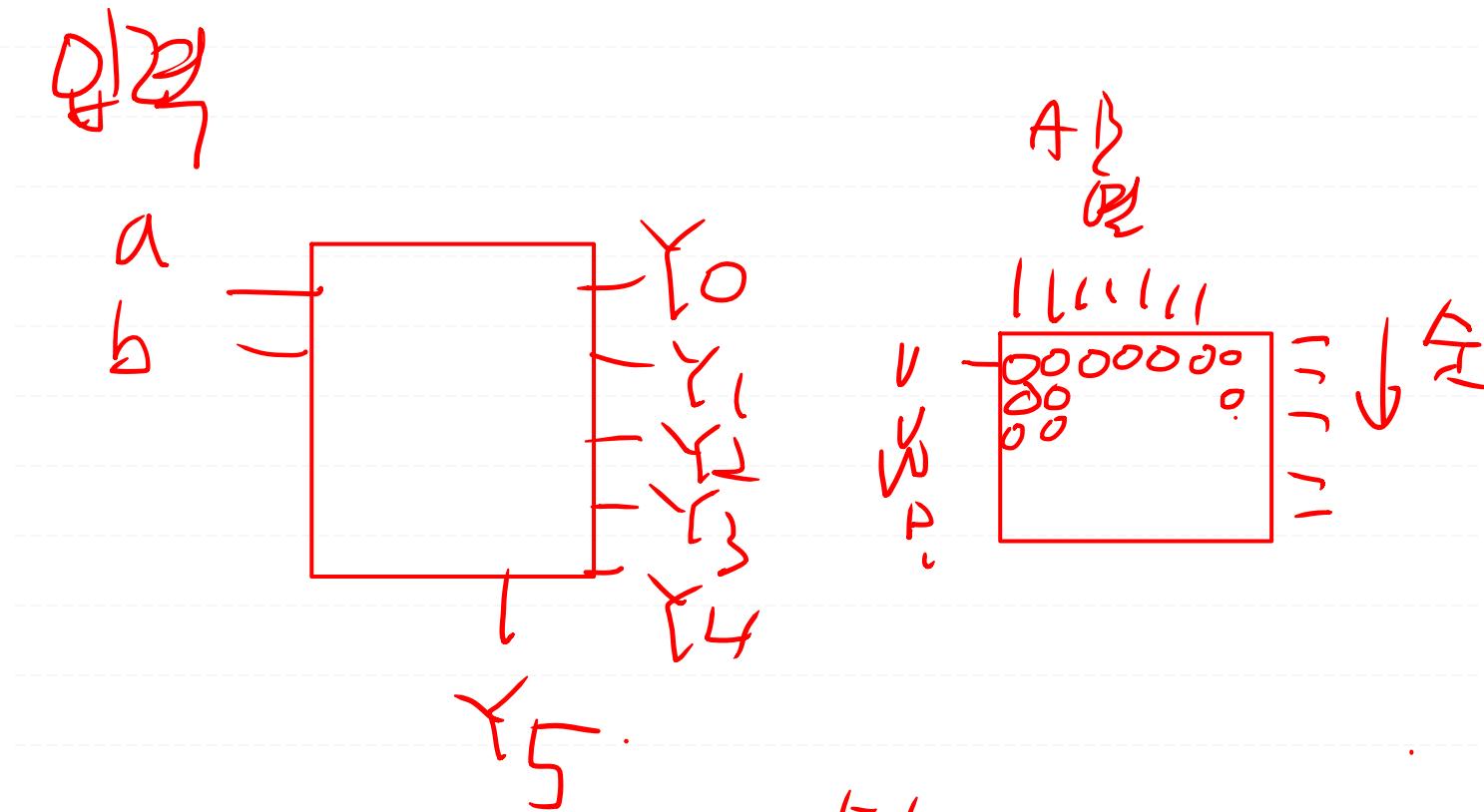
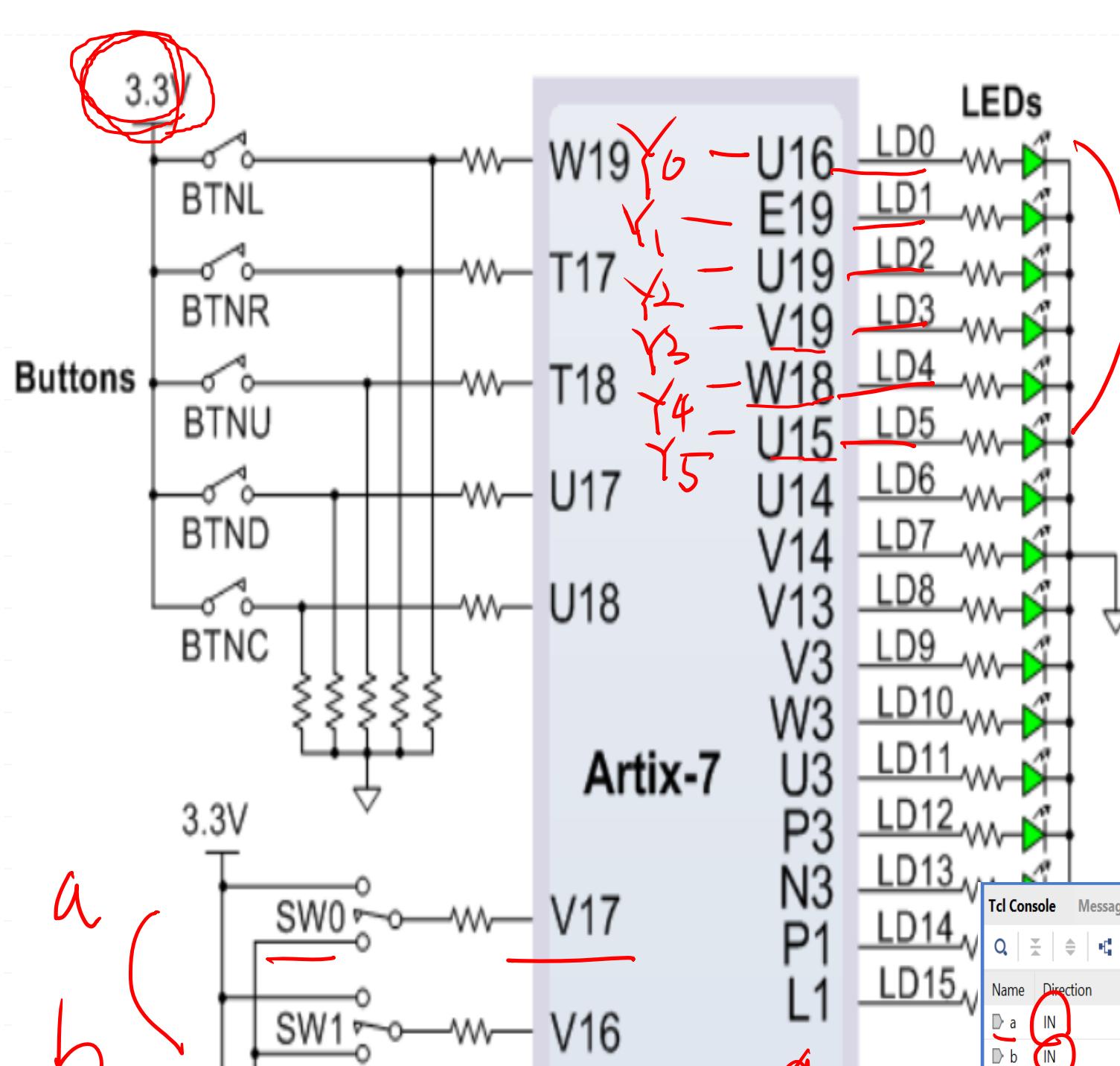
IMPLEMENTATION





Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod connector(s)	10	Programming mode jumper
3	Analog signal Pmod connector (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper





Constraints

PROJECT MANAGER - 250227\_gates

Sources

Design Sources (1)  
gates (gates.i)

Constraints (1)  
constrs\_1 (1)  
gates.xdc (target) *(highlighted)*

Simulation Sources (1)  
sim\_1 (1)

Utility Sources

Hierarchy Libraries Compil

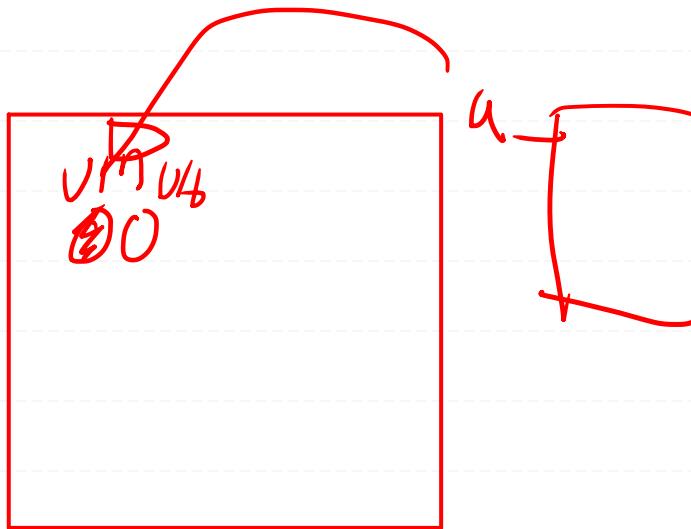
Source File Properties

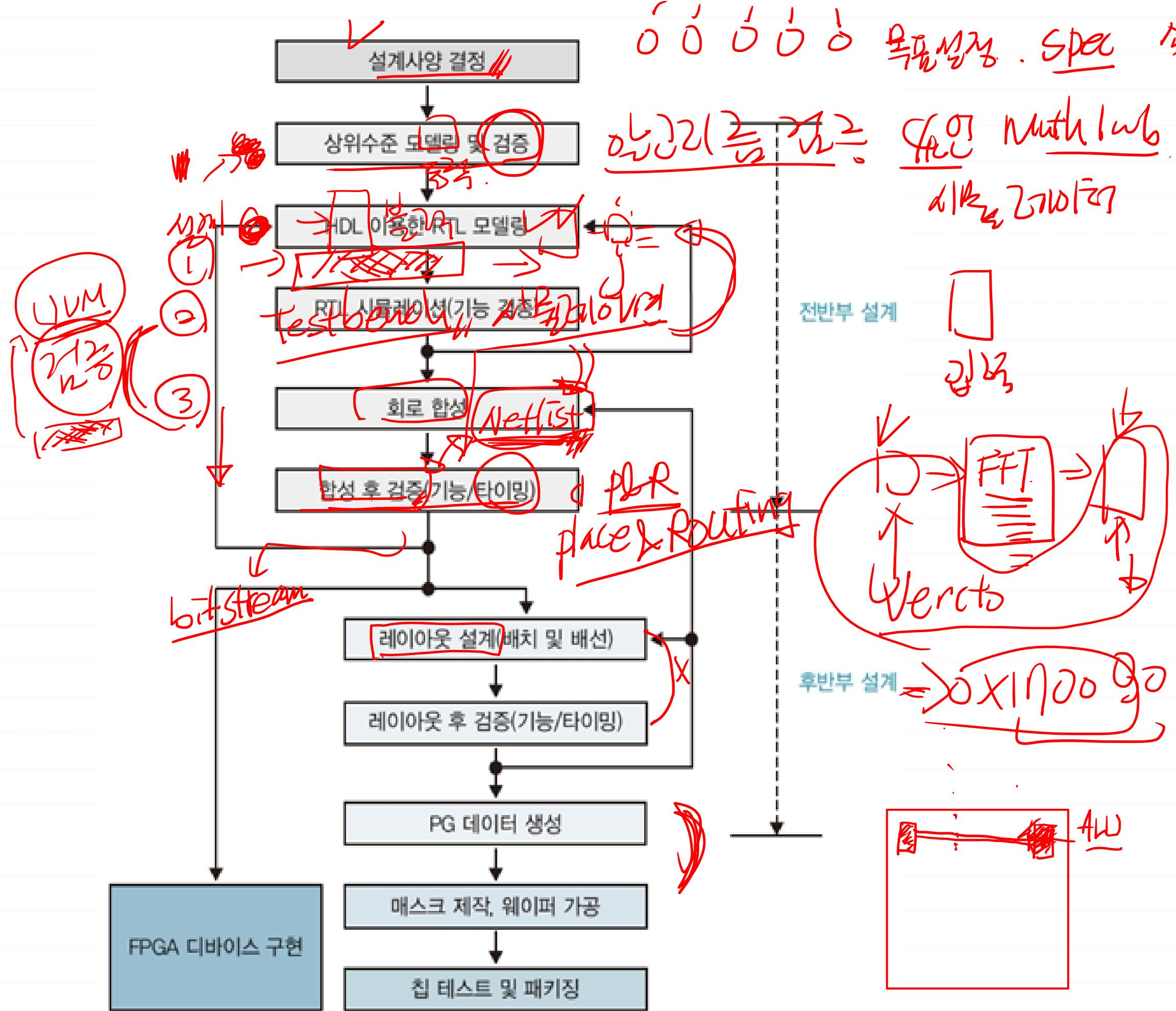
gates.xdc

gates.xdc (C:\Working\FPGA\_Harman\_25\_1\25...\\250227\_gates.srscs\constrs\_1\new) - GVIM1

파일(F) 편집(E) 도구(T) 문법(S) 버퍼(B) 창(W) 도움말(H)

1 set\_property PACKAGE\_PIN V17 [get\_ports a]  
2 set\_property PACKAGE\_PIN V16 [get\_ports b]  
3 set\_property PACKAGE\_PIN U16 [get\_ports y0]  
4 set\_property PACKAGE\_PIN E19 [get\_ports y1]  
5 set\_property PACKAGE\_PIN U19 [get\_ports y2]  
6 set\_property PACKAGE\_PIN V19 [get\_ports y3]  
7 set\_property PACKAGE\_PIN W18 [get\_ports y4]  
8 set\_property PACKAGE\_PIN U15 [get\_ports y5]  
9 set\_property IOSTANDARD LVCMOS33 [get\_ports a]  
10 set\_property IOSTANDARD LVCMOS33 [get\_ports b]  
11 set\_property IOSTANDARD LVCMOS33 [get\_ports y0]  
12 set\_property IOSTANDARD LVCMOS33 [get\_ports y1]  
13 set\_property IOSTANDARD LVCMOS33 [get\_ports y2]  
14 set\_property IOSTANDARD LVCMOS33 [get\_ports y3]  
15 set\_property IOSTANDARD LVCMOS33 [get\_ports y4]  
16 set\_property IOSTANDARD LVCMOS33 [get\_ports y5]  
17 set\_property SLEW SLOW [get\_ports y0]





# 6G Modern Standard

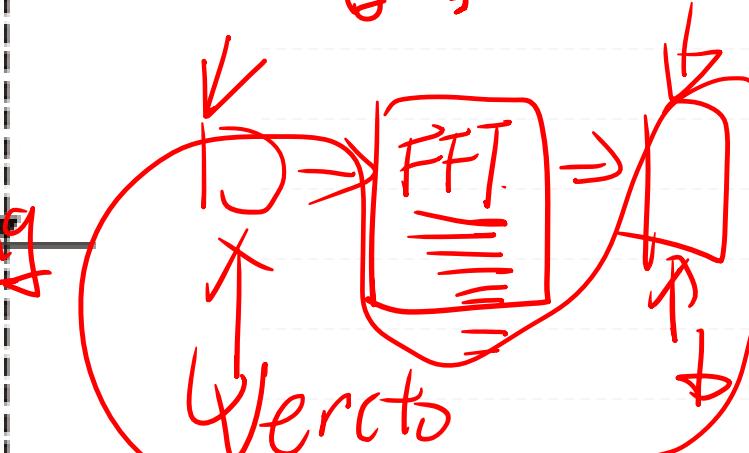
7) 7. x. 1996

# Frac Math Lab

$d \mid k_n, 2710 \mid \bar{c}_7$

전반부 설계

四



후반부 설계

$\Rightarrow \text{Oxido}$

30      35

A red hand-drawn diagram on lined paper. On the left, the number '40' is written above a rectangle. Inside the rectangle, there are several parallel diagonal lines from top-left to bottom-right, creating a hatched pattern. To the right of the rectangle is a large circle containing the word 'GOOM'.

A hand-drawn diagram in red ink on white paper. On the left, there is a large circle representing a sphere. Inside this circle, a smaller circle is drawn near the top right. A grid of intersecting lines is drawn across the sphere's face, with some lines being horizontal and others vertical or diagonal. To the right of the sphere, there is a separate circle containing the number '105'. Above the sphere, there is some handwritten text that appears to be '105'.

A hand-drawn diagram of a car's front end, likely a Ford Taurus, shown from a front-three-quarter perspective. The drawing includes the hood, grille with the Ford oval logo, headlights, and a bumper. Several red annotations are present:

- A large number "6" is written above the front left corner of the car.
- A red bracket on the right side of the diagram groups the word "bumper" and the number "424".
- A red bracket on the left side groups the word "grille" and the number "UVM".
- A red bracket at the bottom groups the words "Hood eng" and "G G G".
- A red checkmark is placed next to the "Hood eng" bracket.
- A red "X" is placed next to the "UVM" label.

### [그림 1-1] HDL 기반 시스템 반도체(디지털 시스템) 설계과정

2진수  
2진수  
bit

2진수  
2진수  
bit

$$\begin{array}{r} a \\ + b \\ \hline 0 - s \end{array}$$

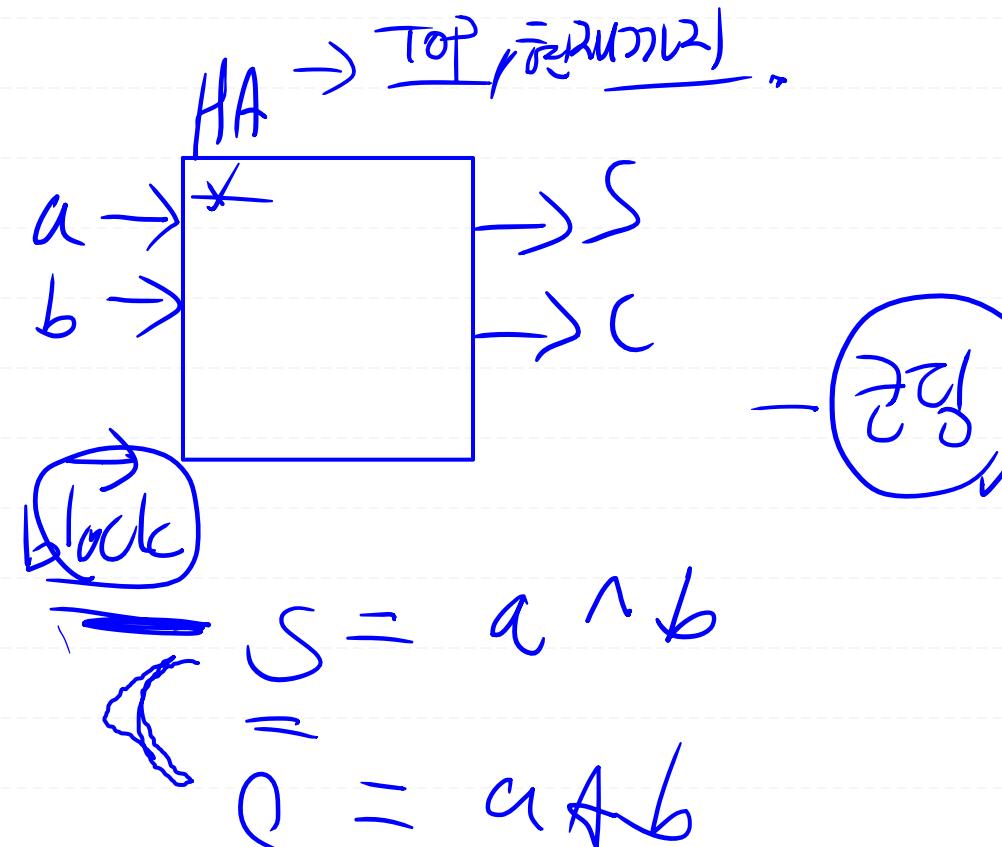
2진수  
2진수  
bit

$$\begin{array}{c|cc|c} & a & b & s & c \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{array}$$

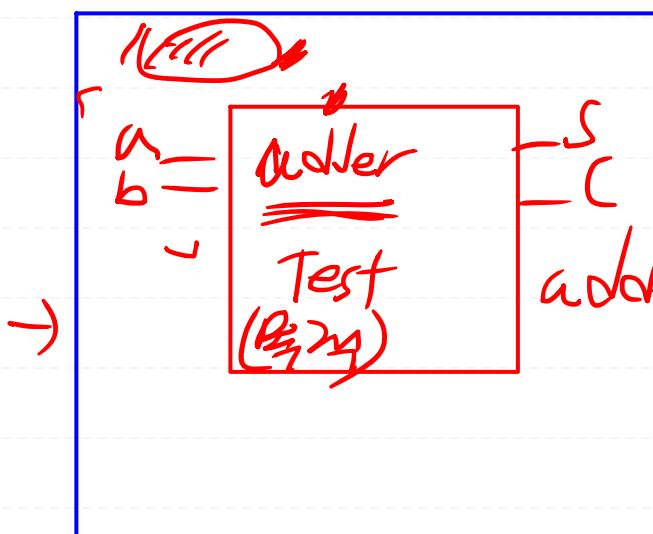
2진수  
2진수  
bit

$$\begin{array}{l} S = a \wedge b \\ Q = a \vee b \end{array}$$

## half Adder



⇒ tb\_adder ← Top

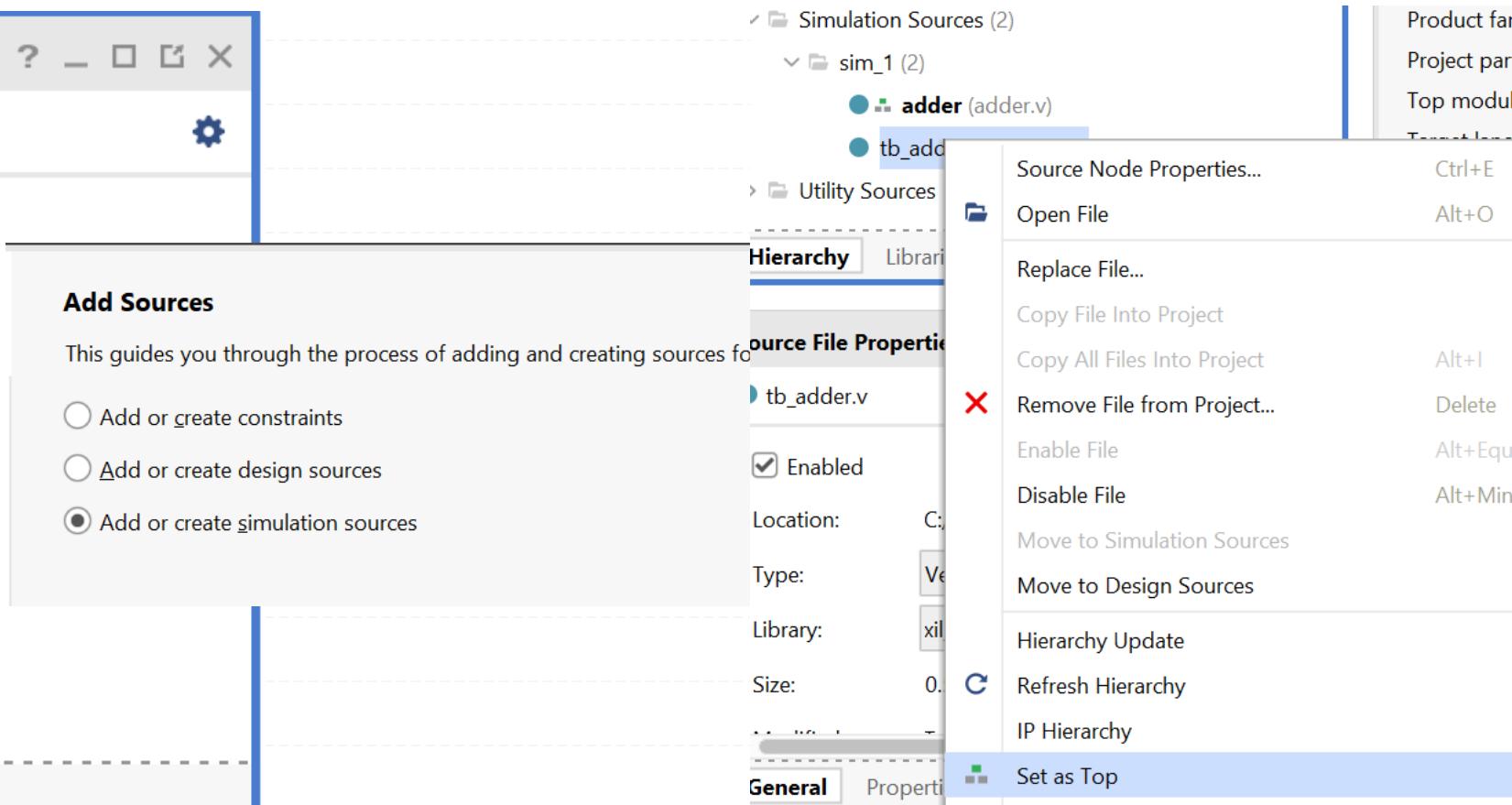
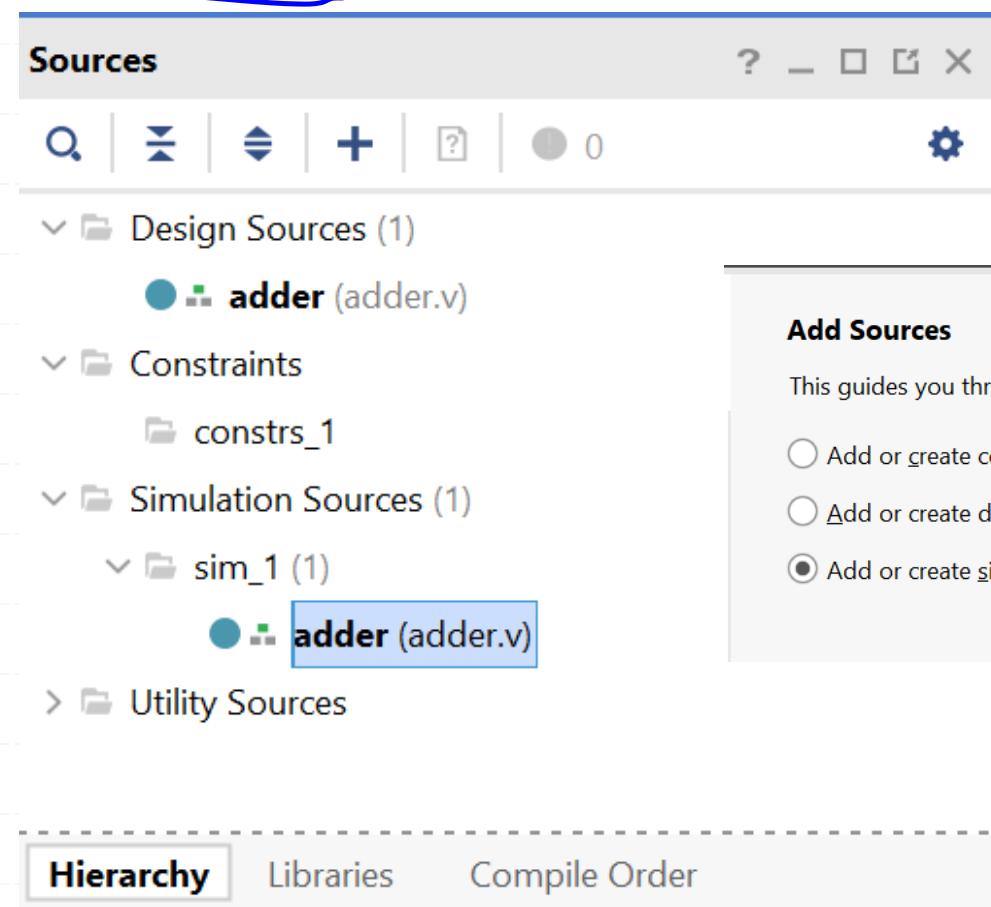


Design module

입수 a  
입수 b  
출력 s  
출력 c

## testbench

= simulation.



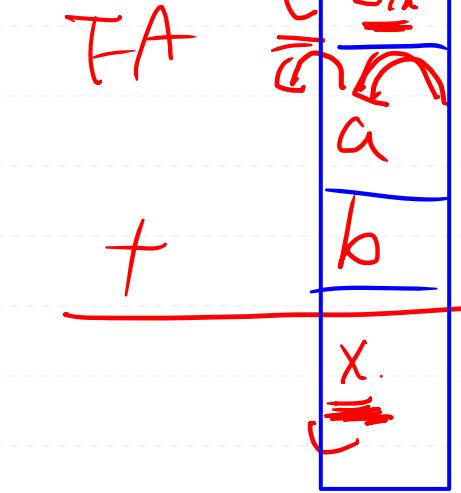
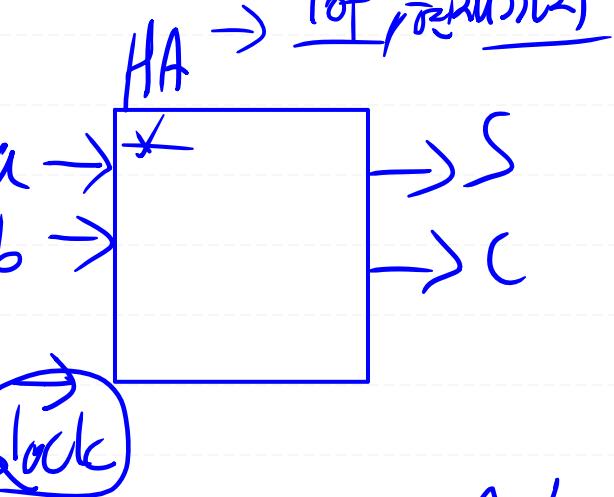
adder u\_half\_adder  
.a a,  
.b b,  
.s s,  
.c c  
);  
tb\_adder  
reg a, b;  
wire s, c;

// reg, 저장하다.  
// 연결용 wire.

시그널.  
**initial :**  
**begin**  
     $a = 0; b = 0;$   
**end**       $\text{delay } 10\text{ns}$   
    \*  $\text{timescale } 1\text{ns}$

w w  
S. C

half Adder



halfAdder

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

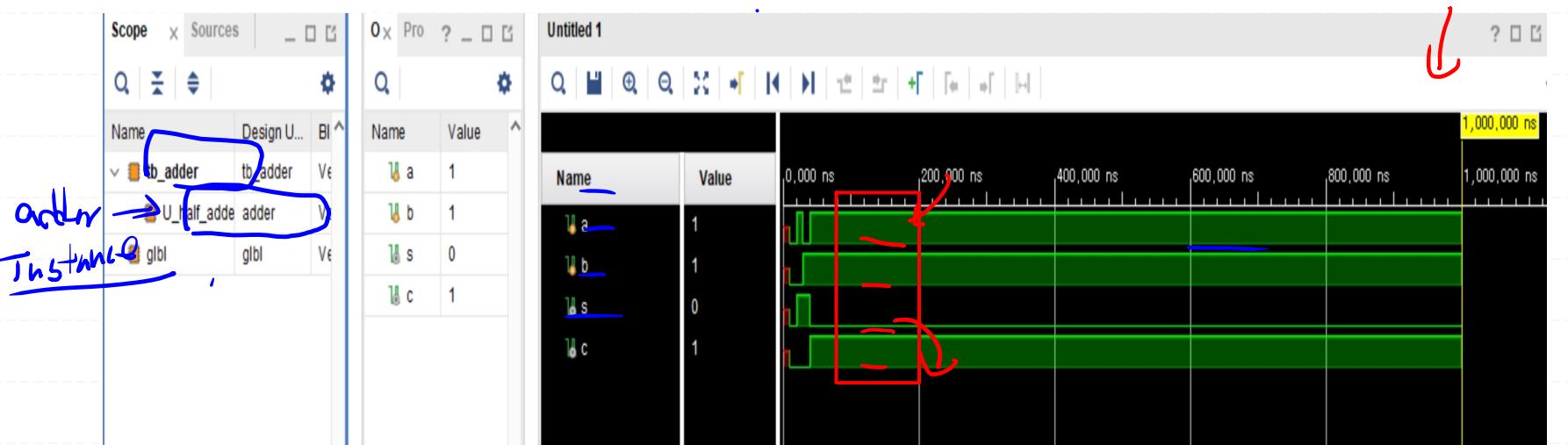
**initial :**  
**begin**  
     $a = 0; b = 0;$   
     $a = 1; b = 0; \text{#10ns}$   
     $a = 0; b = 1; \text{#10ns}$   
     $a = 1; b = 1; \text{#10ns}$   
**end**

$S = a \wedge b$   
 $C = a \oplus b$

1bit FA



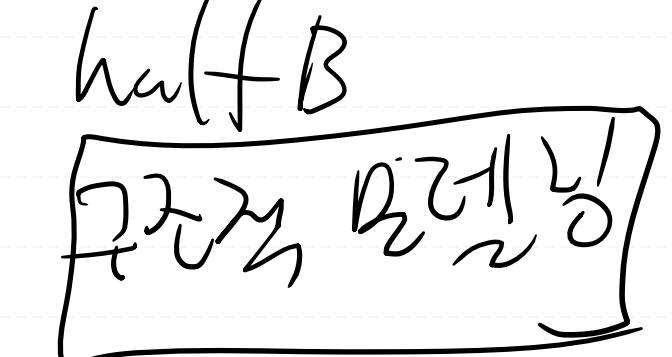
10ns



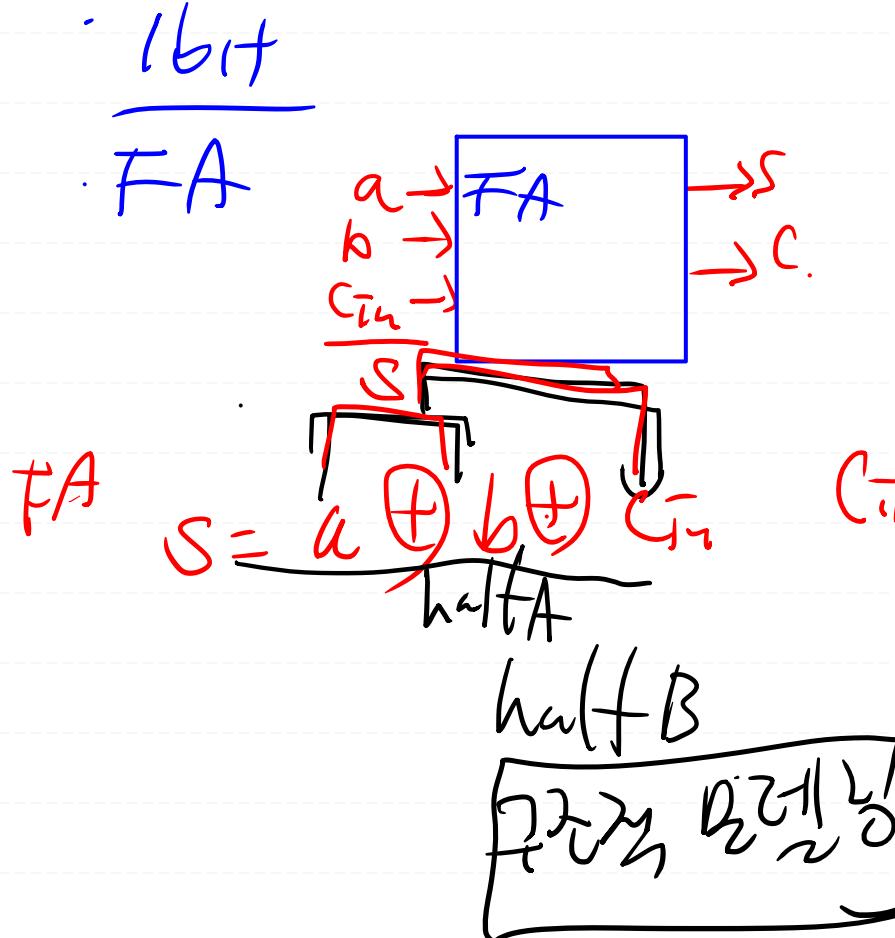
FA

$C_{in}$	a	b	s	c
0	0	0	0	0
0	0	1	1	0
1	0	0	1	0
0	1	0	0	1
1	1	0	1	1

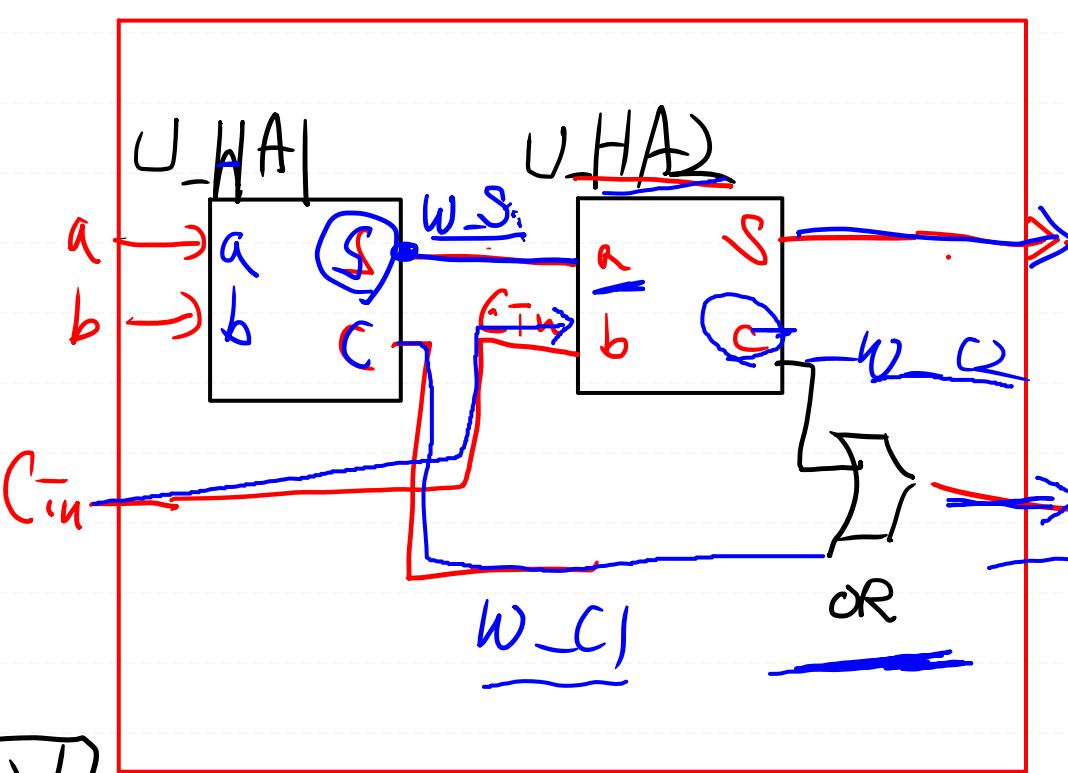
halfA



# Full-Adder -



$C_{in}$	$a$	$b$	$S$	$C$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



```

3 // 1bit FA
4 module full_adder(
5   input a,
6   input b,
7   input cin,
8   output s,
9   output c
10 );
11
12 wire w_s; // wiring U_HA1 out s to U_HA2 in a
13 wire w_c1, w_c2;
14 assign c = w_c1 || w_c2;
15
16
17
18
19 half_adder U_HA1(
20   .a(a),
21   .b(b),
22   .s(w_s),
23   .c(w_c1)
24 );
25
26 half_adder U_HA2(
27   .a(w_s), // from U_HA1 of s
28   .b(cin),
29   .s(s),
30   .c(w_c2) // 
31 );
32
33
34
35 endmodule

```

```

37 module half_adder(
38   input a,
39   input b,
40   output s,
41   output c
42 );
43
44 // half adder 1bit
45 assign s = a ^ b;
46 assign c = a & b;
47
48 endmodule

```

```

3 module tb_adder();
4 reg a, b, cin; // reg.
5 wire s, c; // output wire.
6
7 adder u_full_adder(); // module instance
8   .a(a),
9   .b(b),
10  .cin(cin), // input carry.
11  .s(s),
12  .c(c)
13
14
15
16
17 initial
18 begin
19   #10; a = 0; b = 0; cin = 0;
20   #10; a = 1; b = 0; cin = 0;
21   #10; a = 0; b = 1; cin = 0;
22   #10; a = 1; b = 1; cin = 0;
23   #10; a = 0; b = 0; cin = 1;
24   #10; a = 1; b = 0; cin = 1;
25   #10; a = 0; b = 1; cin = 1;
26   #10; a = 1; b = 1; cin = 1;
27
28 $stop;
29 end

```

\* 4 bit FA M<sub>2</sub> | Test bench

home work