SOGANG UNIVERSITY

컴퓨터공학실험II

3장 AND/OR/NOT

Be as proud of Sogang
As Sogang is proud of you

실험 목표

- AND/OR/NOT Gate의 동작의 이해 및 확인
- Verilog를 사용하여 다중입력 AND/OR Gate 및 NOT Gate 구현
- 입력 신호 생성 후 Simulation을 통하여 구현된 각 Gate 동작 확인
- FPGA 통해서 Verilog로 구현된 회로의 동작 확인

GATE

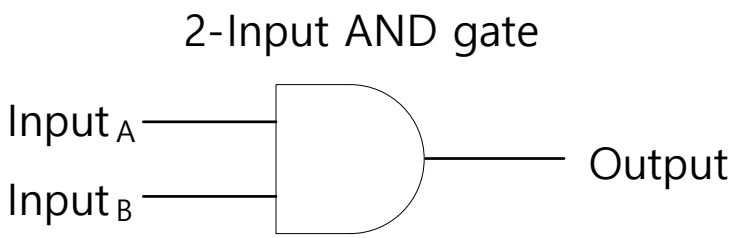
- 디지털 시스템의 회로를 구성하는 가장 기본적인 요소
- 0과 1로 대변되는 논리대수에 의한 연산을 집적회로 형태로 구현
- 이러한 논리대수의 기본이 되는 게이트로는 AND, OR, NOT 등이 있으며, 이들의 조합으로 다양한 형태의 게이트 생성

AND 게이트

- 2개 이상의 입력신호를 받는 게이트로 모든 신호가 High(1)이어야만 High신호를 출력하고 그 이외에는 Low를 출력

- Boolean 식에서 곱셈으로 표현

$$C = A \cdot B$$



2-input AND gate Schematic

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

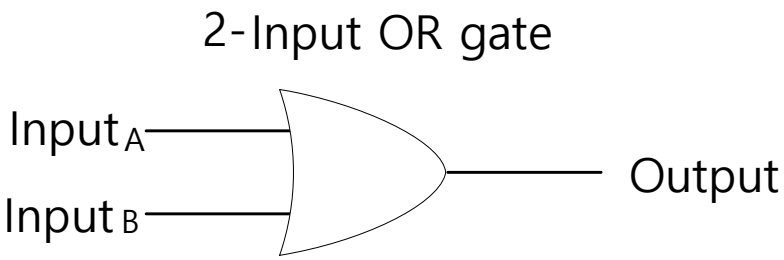
2-input AND gate Truth Table

OR 게이트

- 2개 이상의 신호를 받으며, 두 신호 중 하나라도 High일 때는 High 값을, High값이 하나도 없을 경우 Low값을 출력

- Boolean 식에서 덧셈으로 표현

$$C = A + B$$



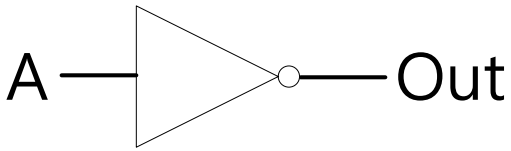
2-input OR gate Schematic

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

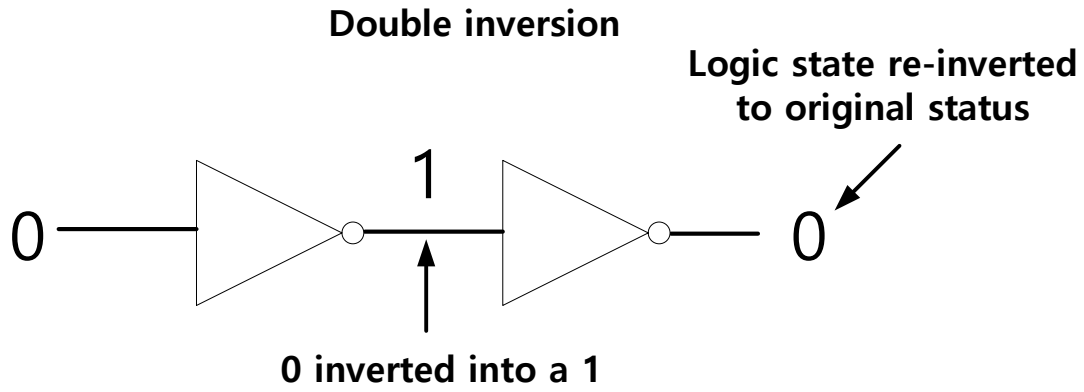
2-input OR gate Truth Table

NOT / BUFFER 게이트

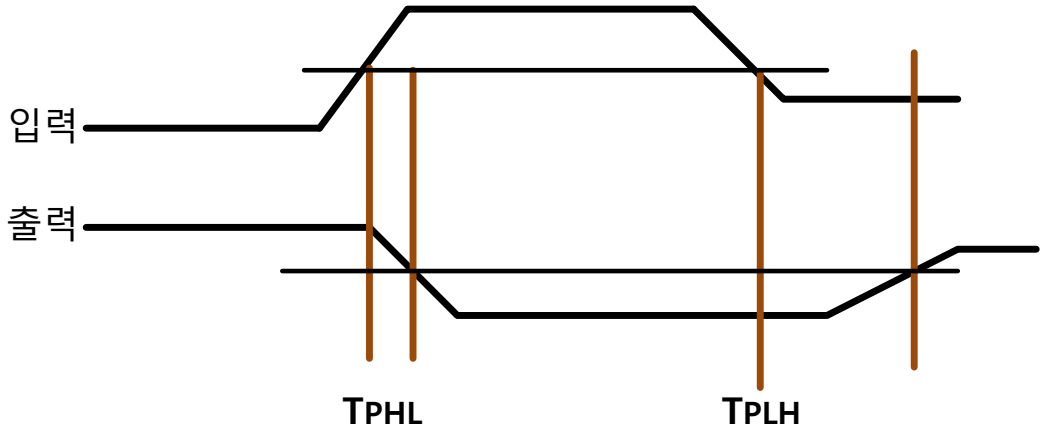
- NOT 게이트는 Inverter라고도 하며, 입력값을 반전시켜 출력
- BUFFER 게이트는 NOT gate 두 개로 이루어져있고, 신호가 변하진 않지만 신호의 세기를 강화



Input	Output
0	1
1	0



전파지연(Propagation Delay Time)



- 신호 값의 변화가 입력에서 출력까지 전달되는데 걸리는 평균시간
- 논리 게이트의 지연시간 및 개수 등에 영향
- $T_{PD} = (T_{PHL} + T_{PLH}) * 1/2$

다단계/다중입력(Boolean 함수)

- 항등원 법칙 (a) x+0=x (b) x*1=x
- 보수 법칙 (a) x+x′=1 (b) x*x′=0
- 등역 법칙 (a) x+x=x (b) x*x=x
- 경계 법칙 (a) x+1=1 (b) x*0=0
- 대합 법칙 (a) (x′)′=x
- 교환 법칙 (a) x+y=y+x (b) xy=yx
- 연관 법칙 (a) x+(y+z)=(x+y)+z (b) x(yz)=(xy)z
- 분배 법칙 (a) x(y+z)=xy+xz (b) x+yz=(x+y)(x+z)
- 드모르간 법칙 (a) (x+y)′=x′y′ (b) (xy)′=x′+y′
- 흡수 법칙 (a) x+xy=x(1+y)=x*1=x (b) x(x+y)=x

FPGA 동작 검증

◆동작검증 단계

- Verilog coding
- Run synthesis
- Device/Pin assignment
- Synthesis / Implement
- Device configuration

FPGA 동작 검증

- Example(NOT gate)
  - Verilog coding

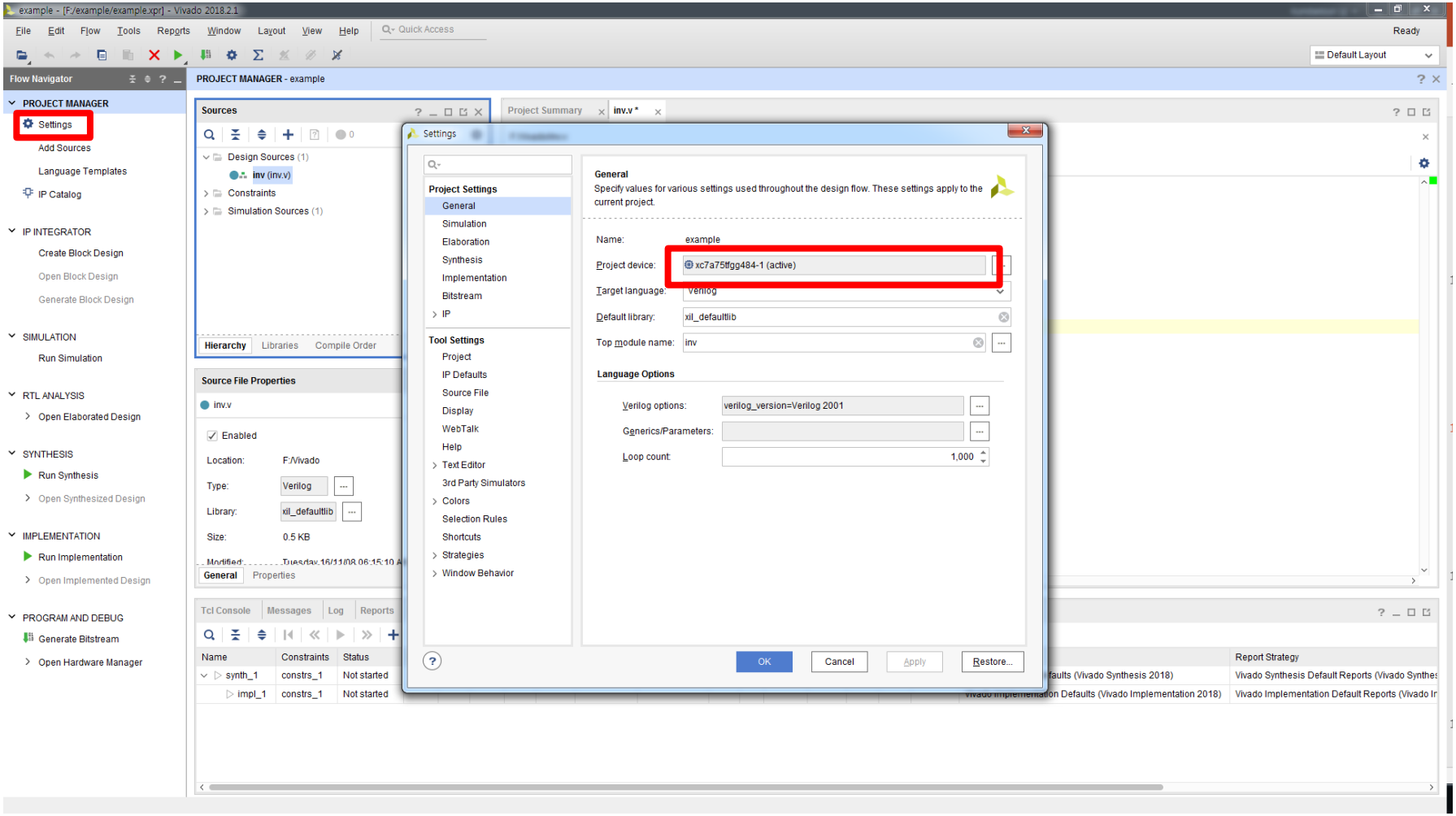
Project Summary | inv.v *

F:/Vivado/inv.v

```
1  `timescale 1ns / 1ps
2
3  module inv(
4      input a,
5      output y
6      );
7
8      assign y = ~a;
9
10 endmodule
11
```

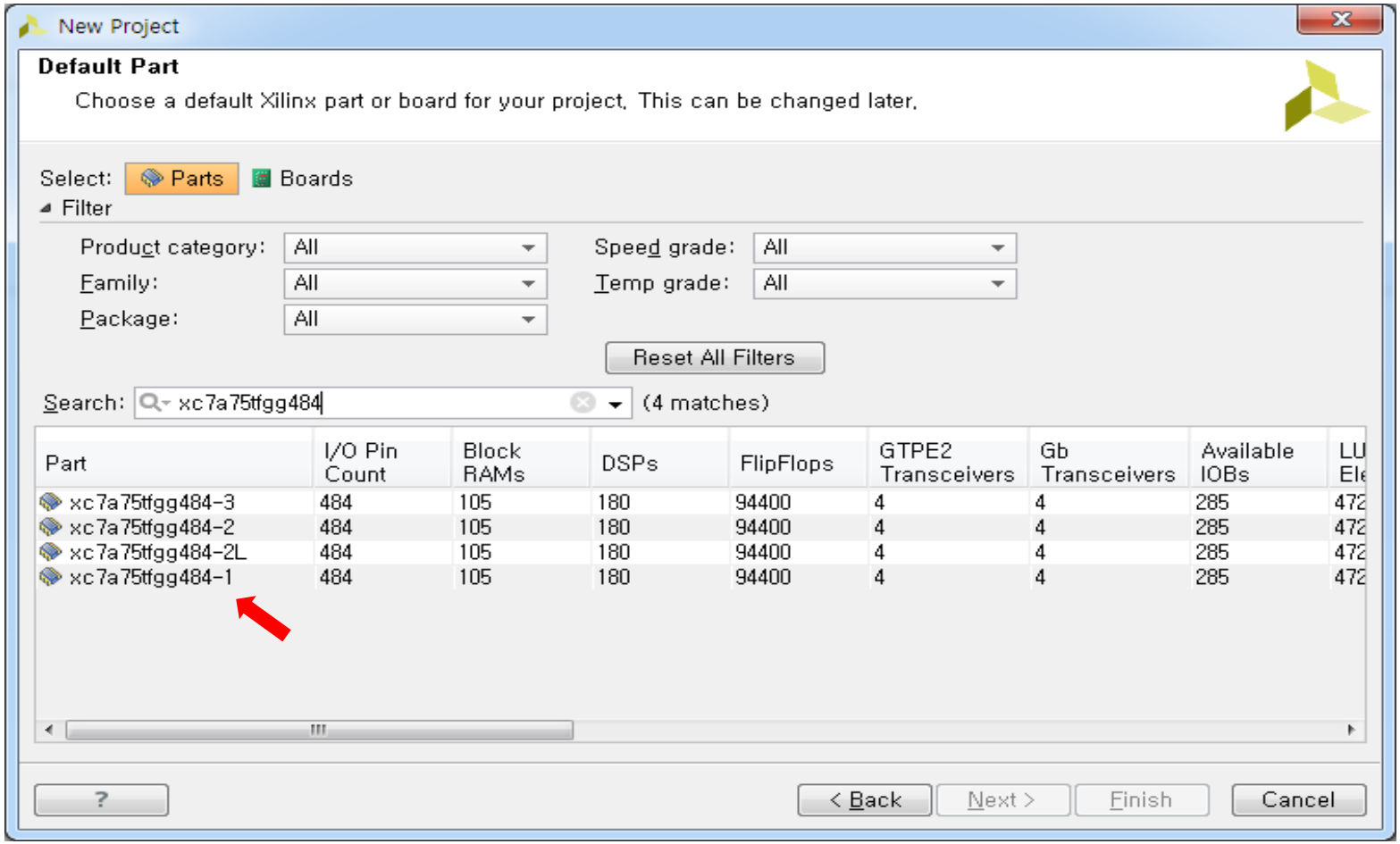
FPGA 동작 검증

- Example(NOT gate)
  - Device assignment



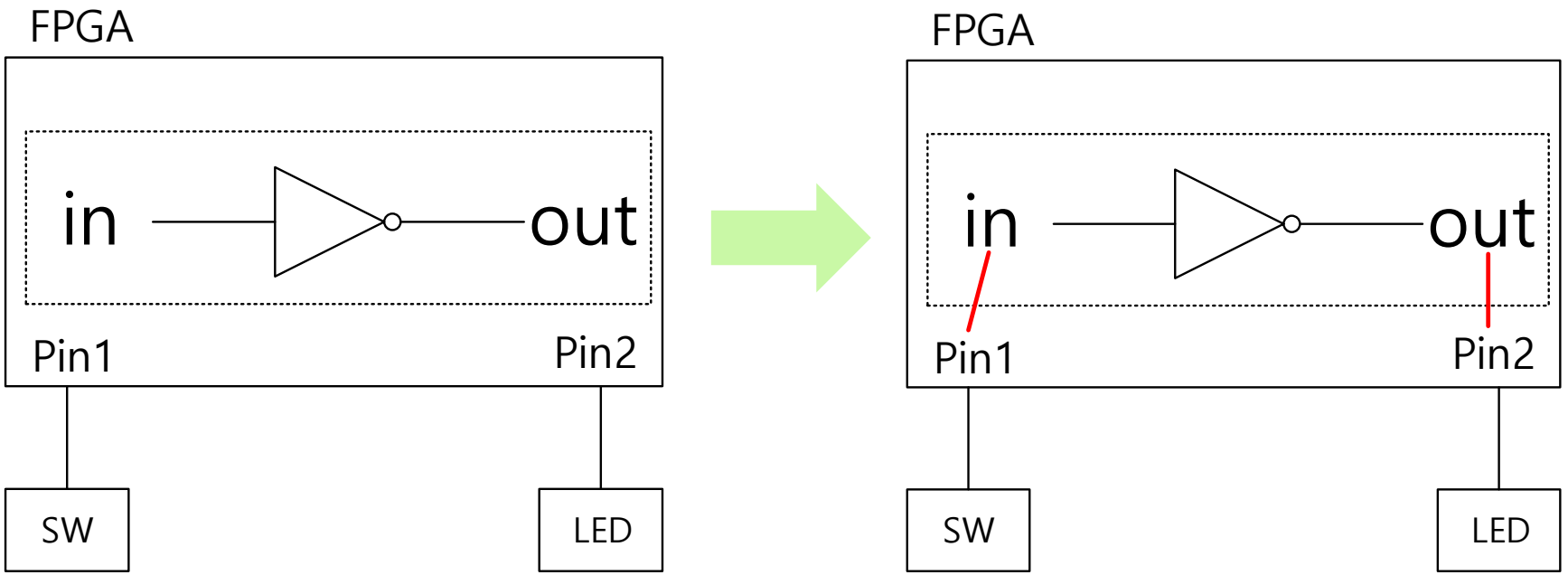
FPGA 동작 검증

- Example(NOT gate)
  - Device assignment
    - ✓ Device : **xc7a75tfgg484(Artix7)**



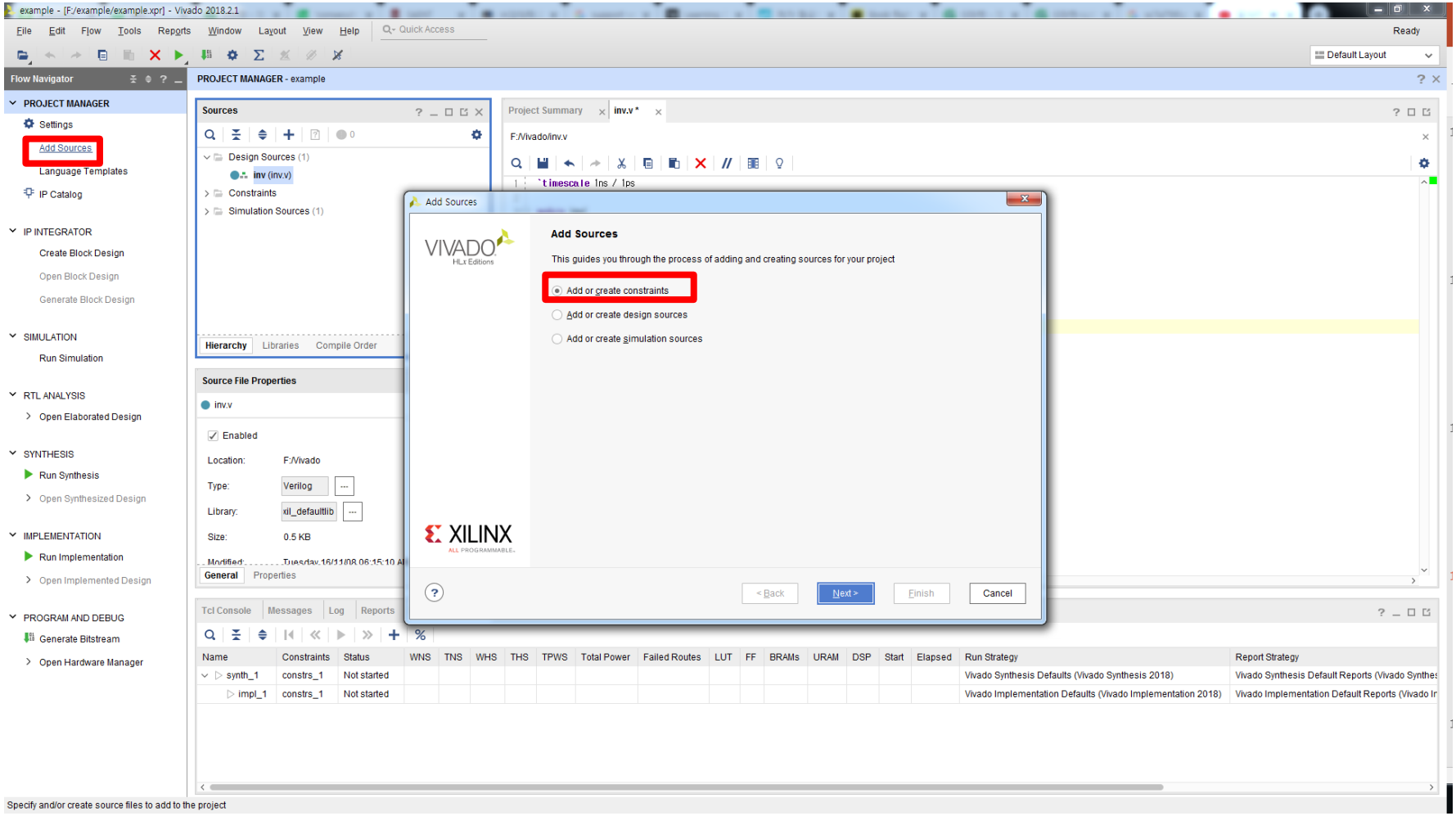
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment



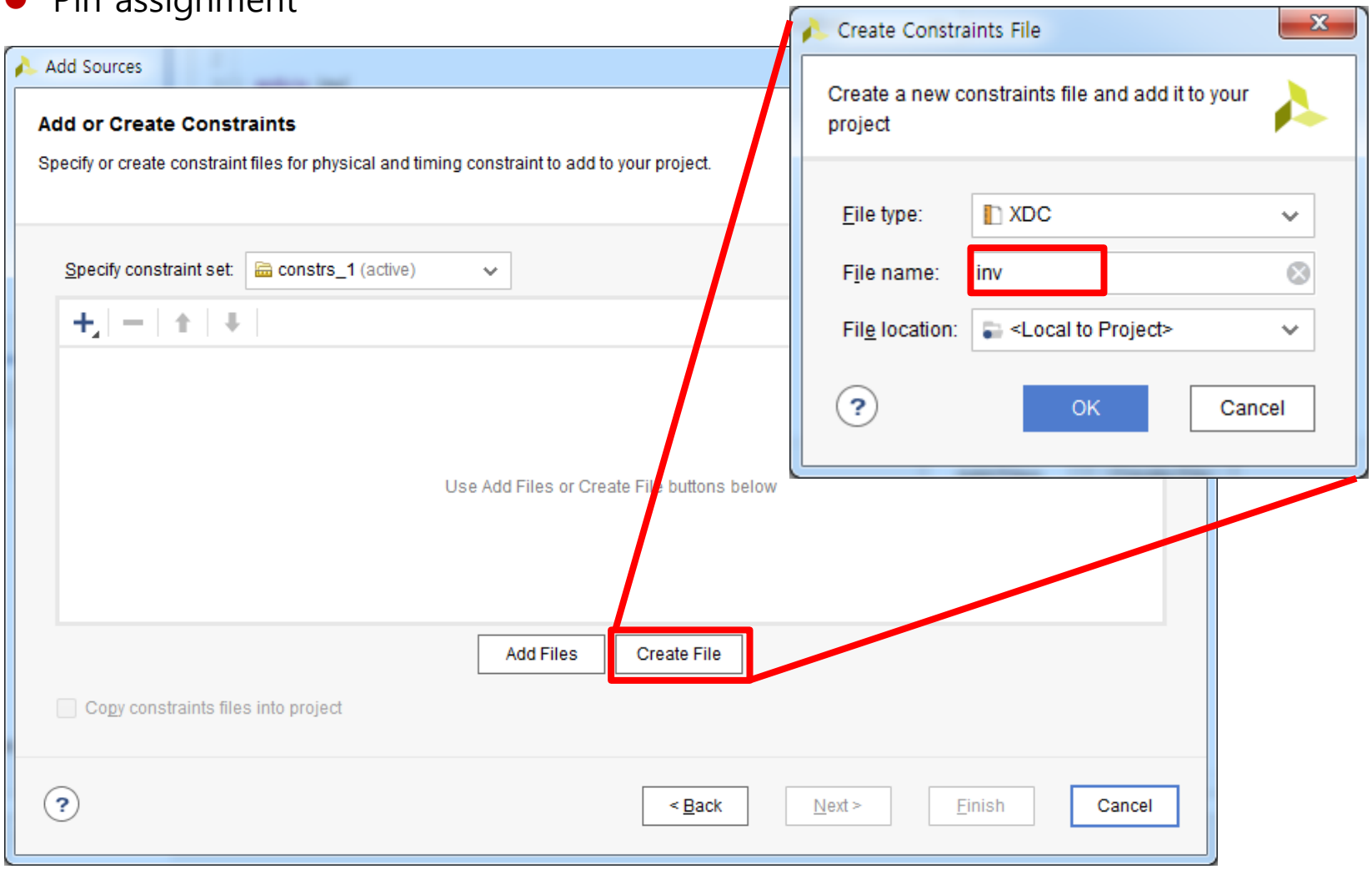
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment



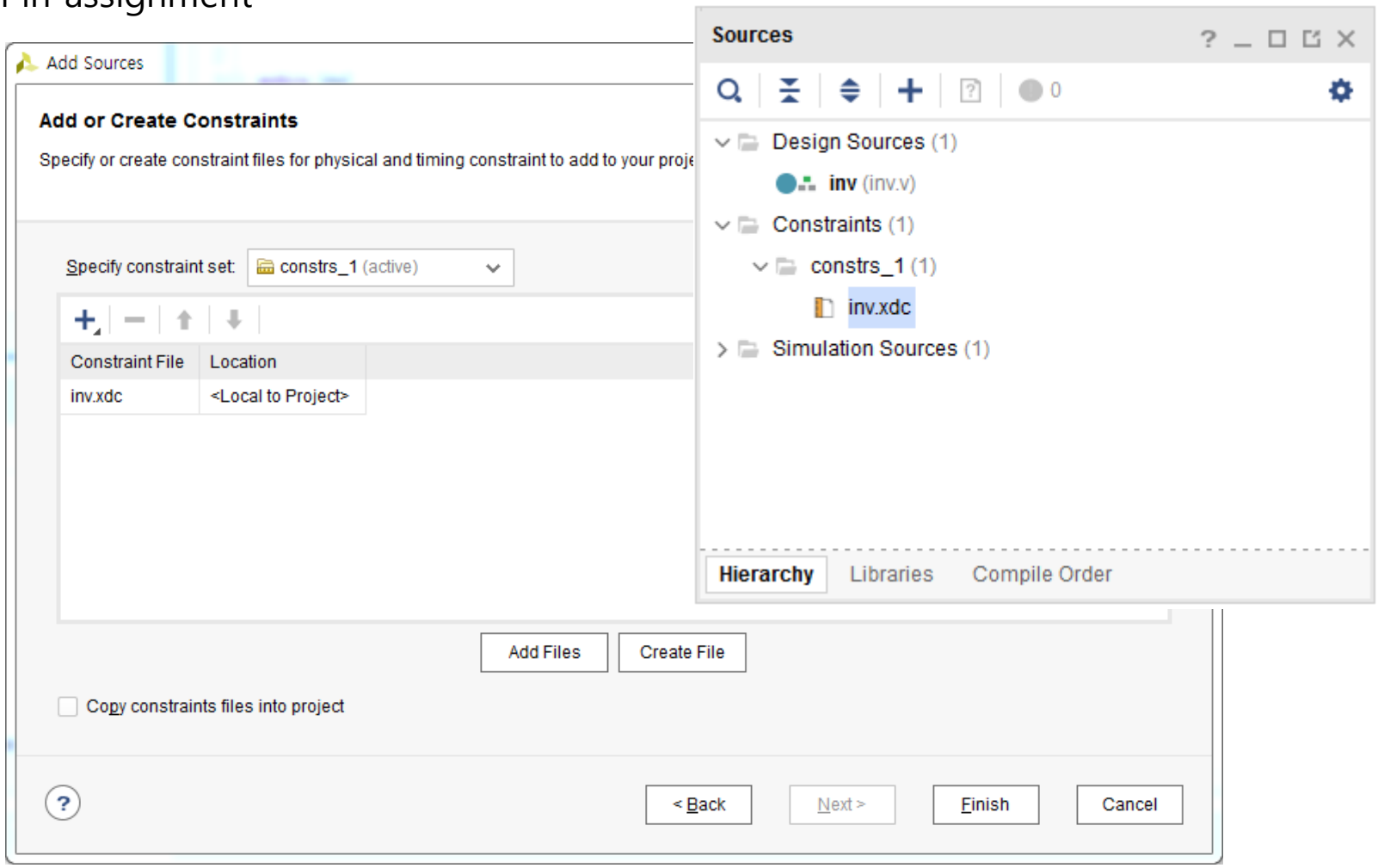
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment



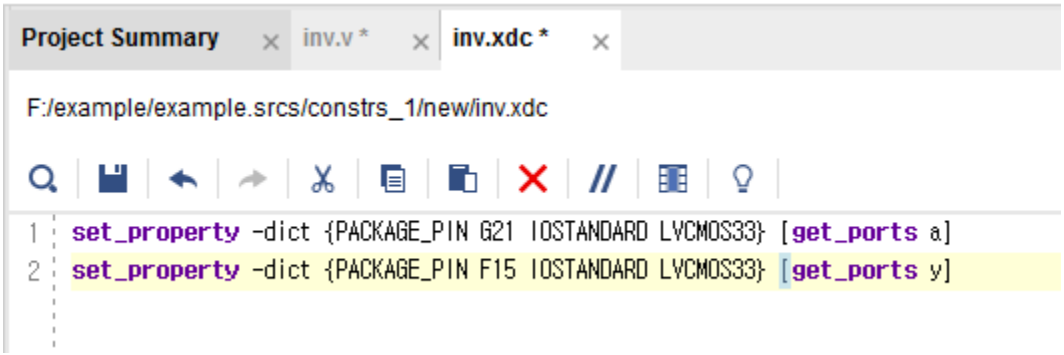
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment



FPGA 동작 검증

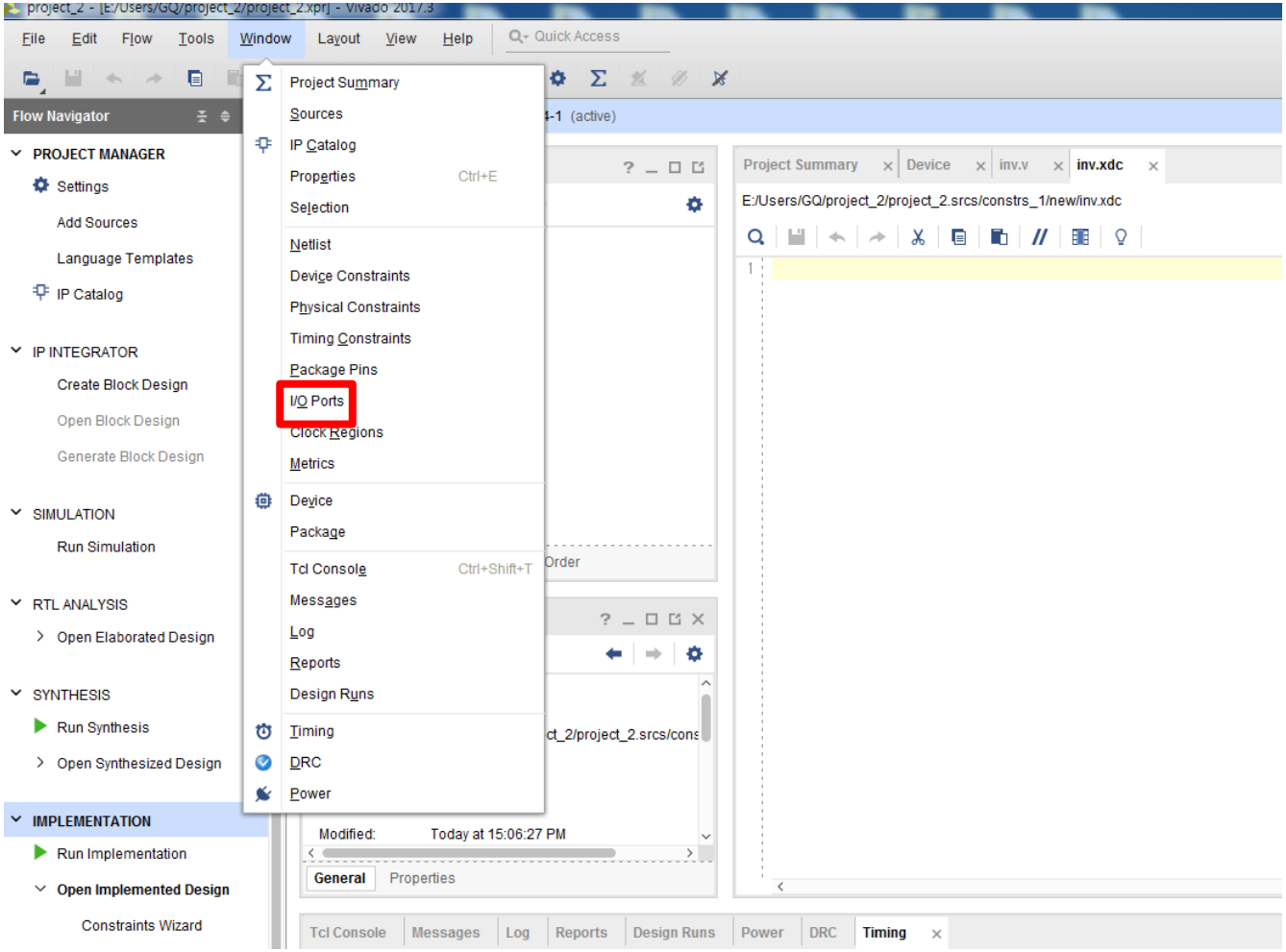
- Example(NOT gate)
  - Pin assignment
    - FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



set_property -dict {PACKAGE_PIN G21 IOSTANDARD LVCMOS33} [get_ports a]
set_property -dict {PACKAGE_PIN F15 IOSTANDARD LVCMOS33} [get_ports y]

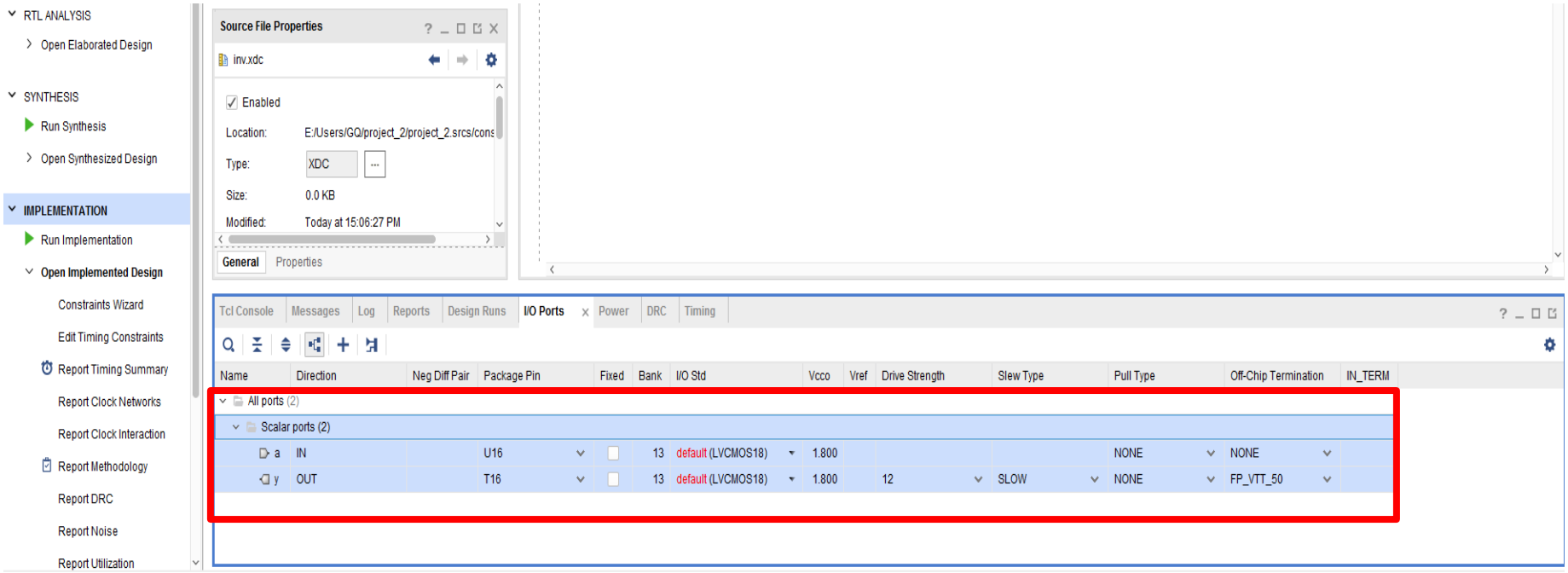
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment



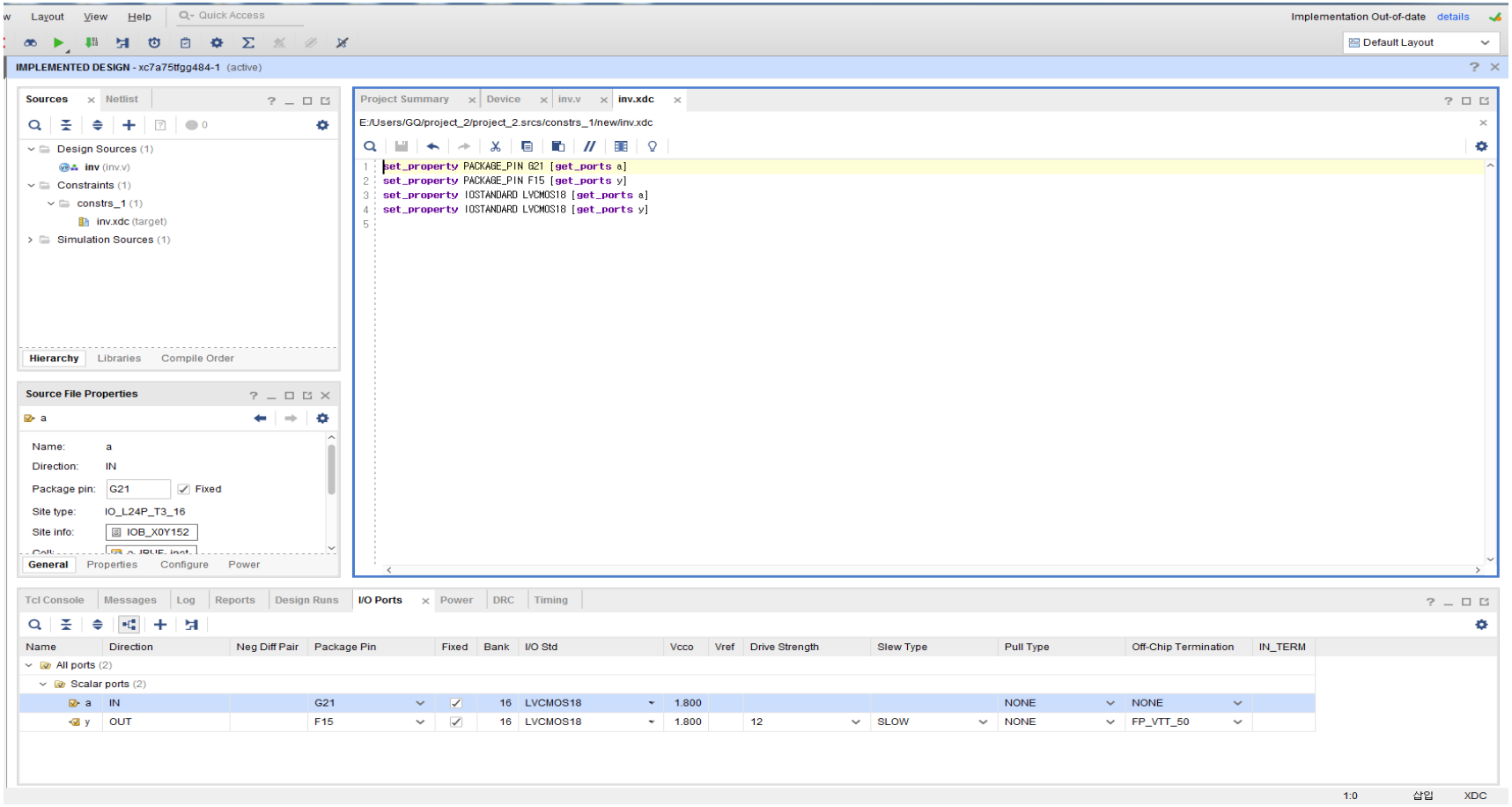
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment
    - FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



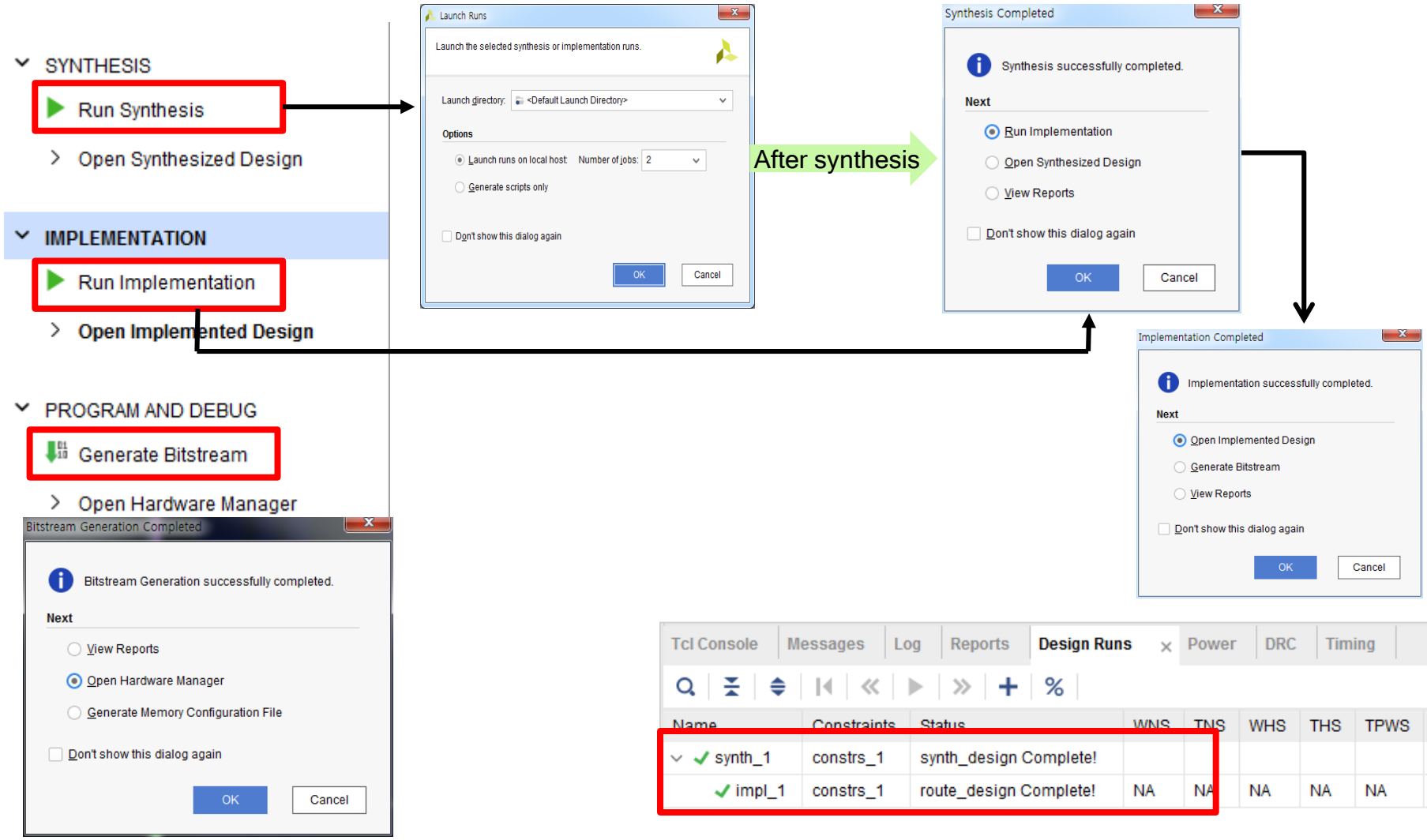
FPGA 동작 검증

- Example(NOT gate)
  - Pin assignment
    - FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



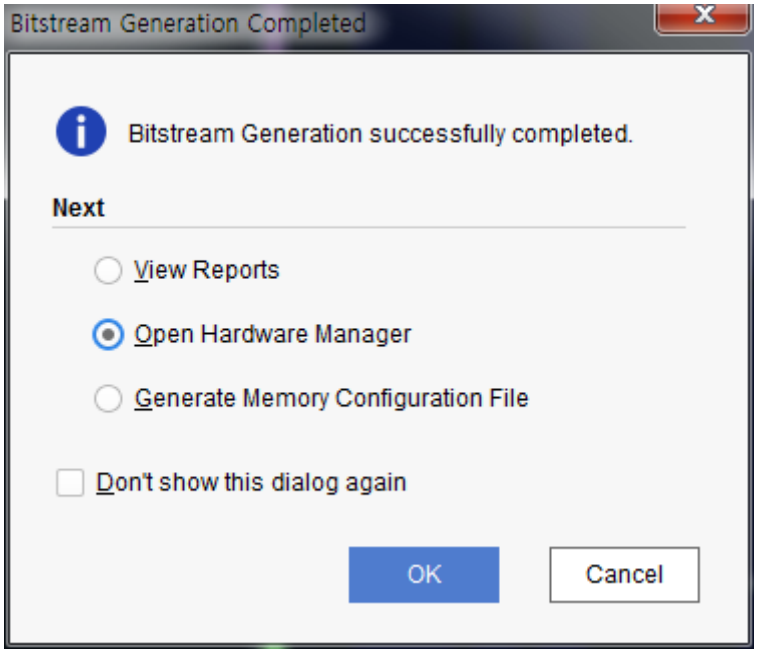
FPGA 동작 검증

- Example(NOT gate)
  - Synthesis / Implementation



FPGA 동작 검증

- Example(NOT gate)
  - Device configuration

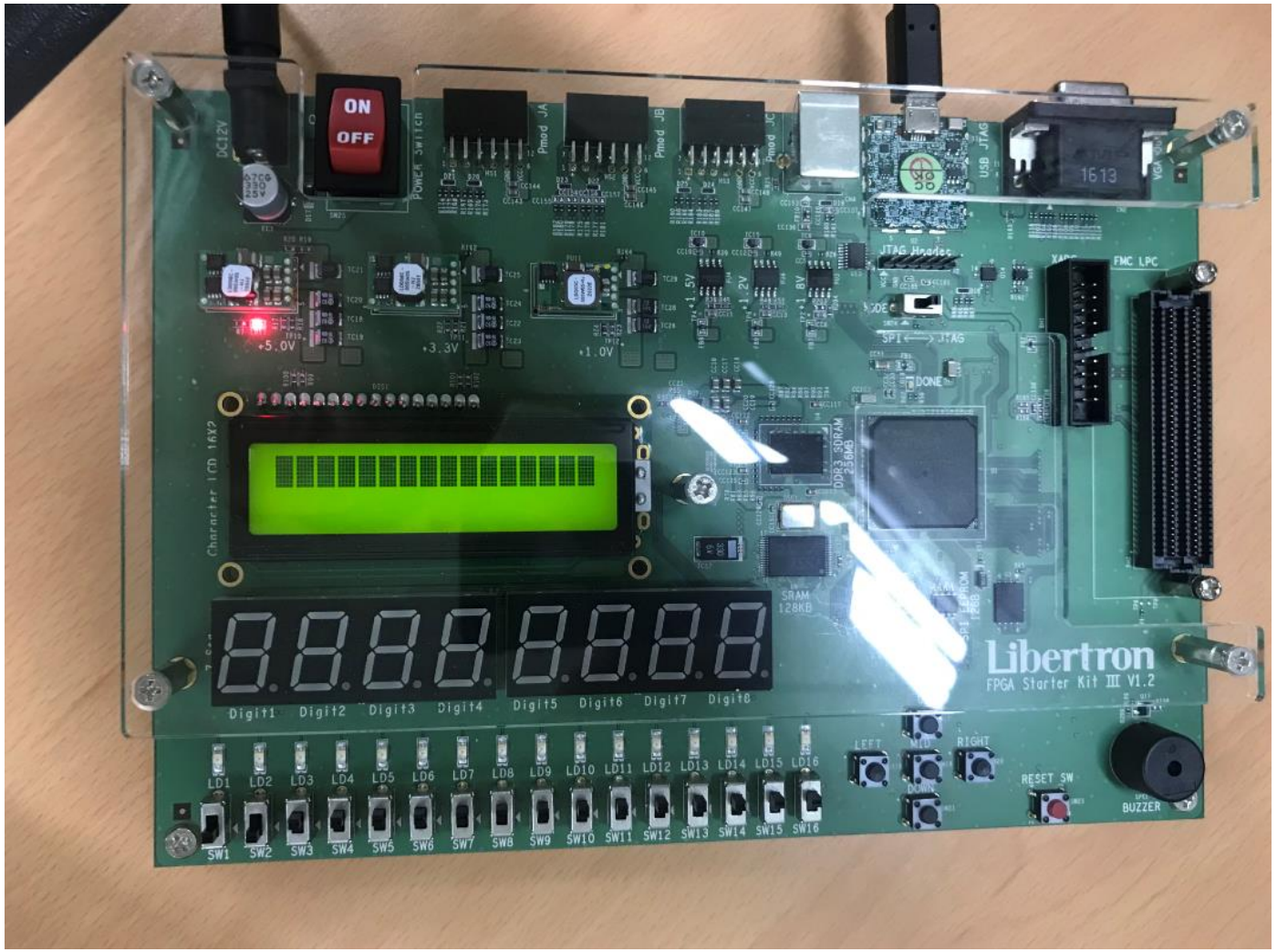


FPGA 동작 검증

- 장비 : FPGA Starter Kit Ⅲ Board
- Cable : Micro USB Cable

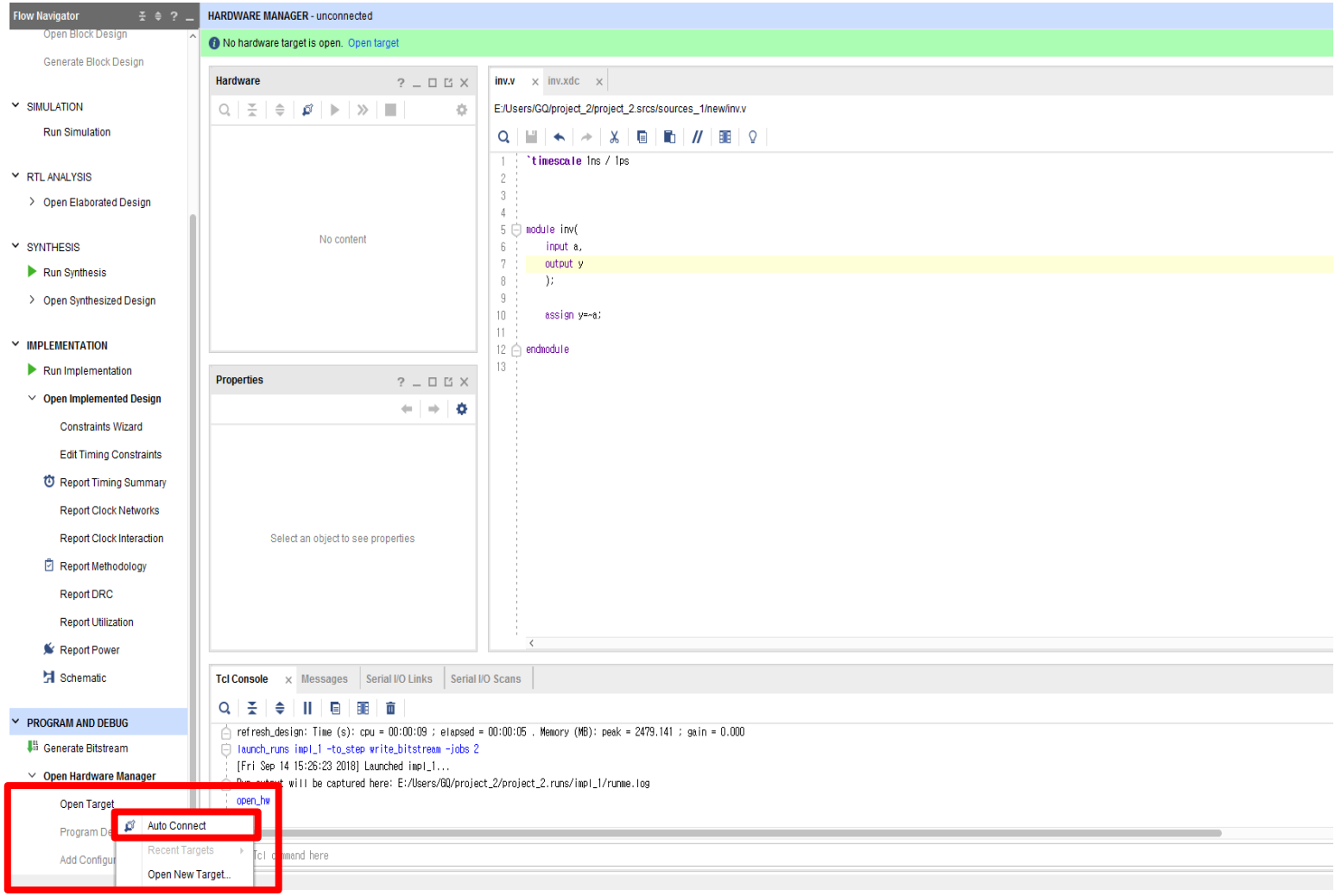
FPGA 동작 검증

- FPGA Starter Kit Ⅲ Board 초기 상태



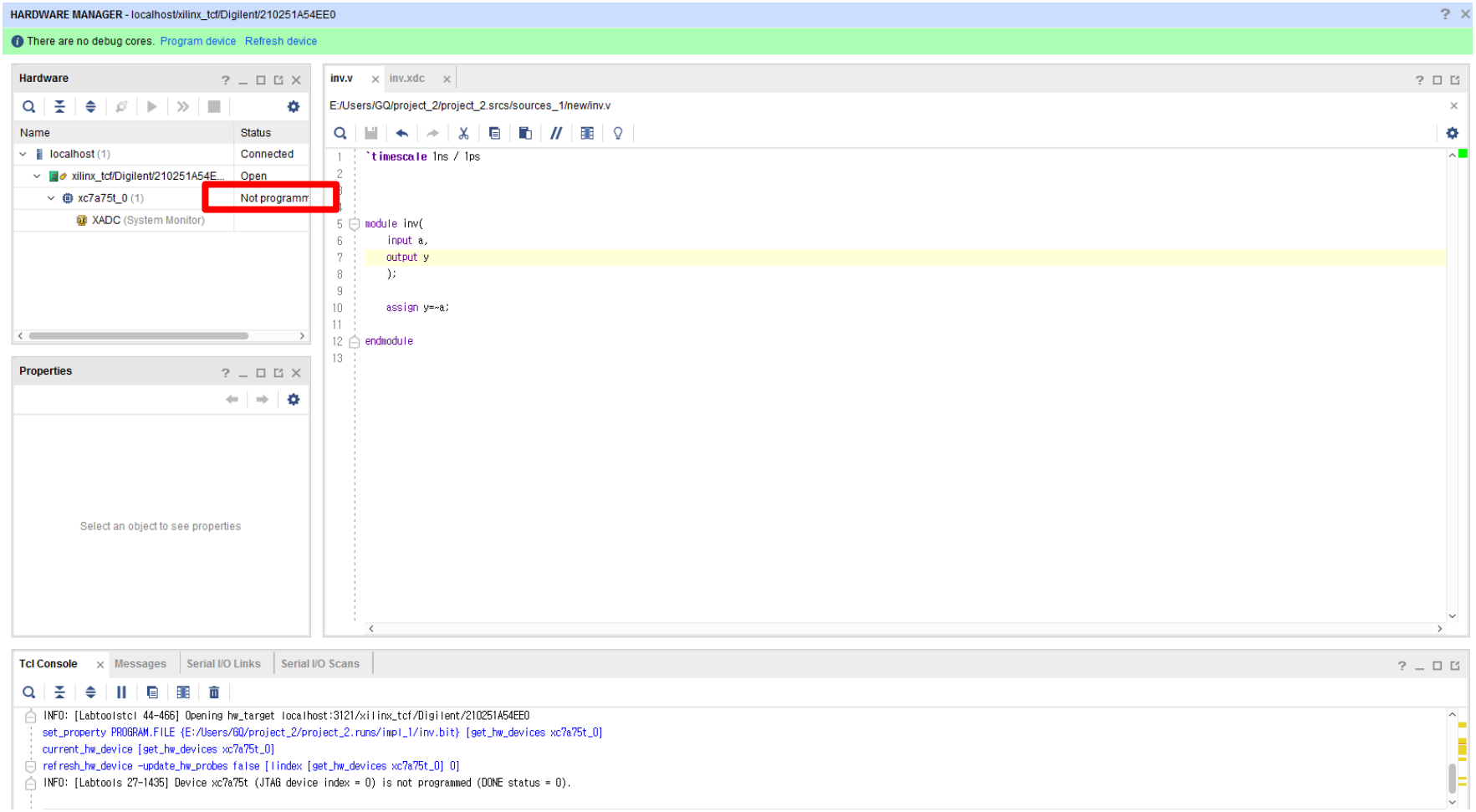
FPGA 동작 검증

- Example(NOT gate)
  - Open Target -> auto connect 클릭



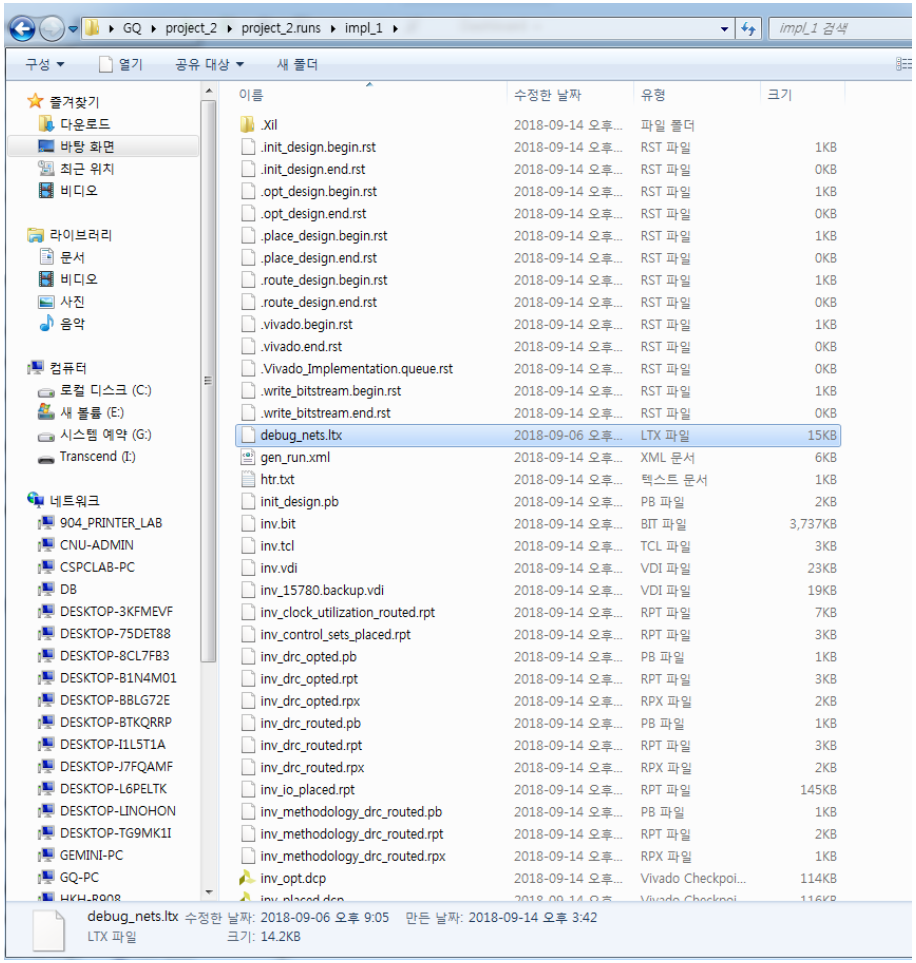
FPGA 동작 검증

- Example(NOT gate)
  - Open Target -> auto connect 클릭



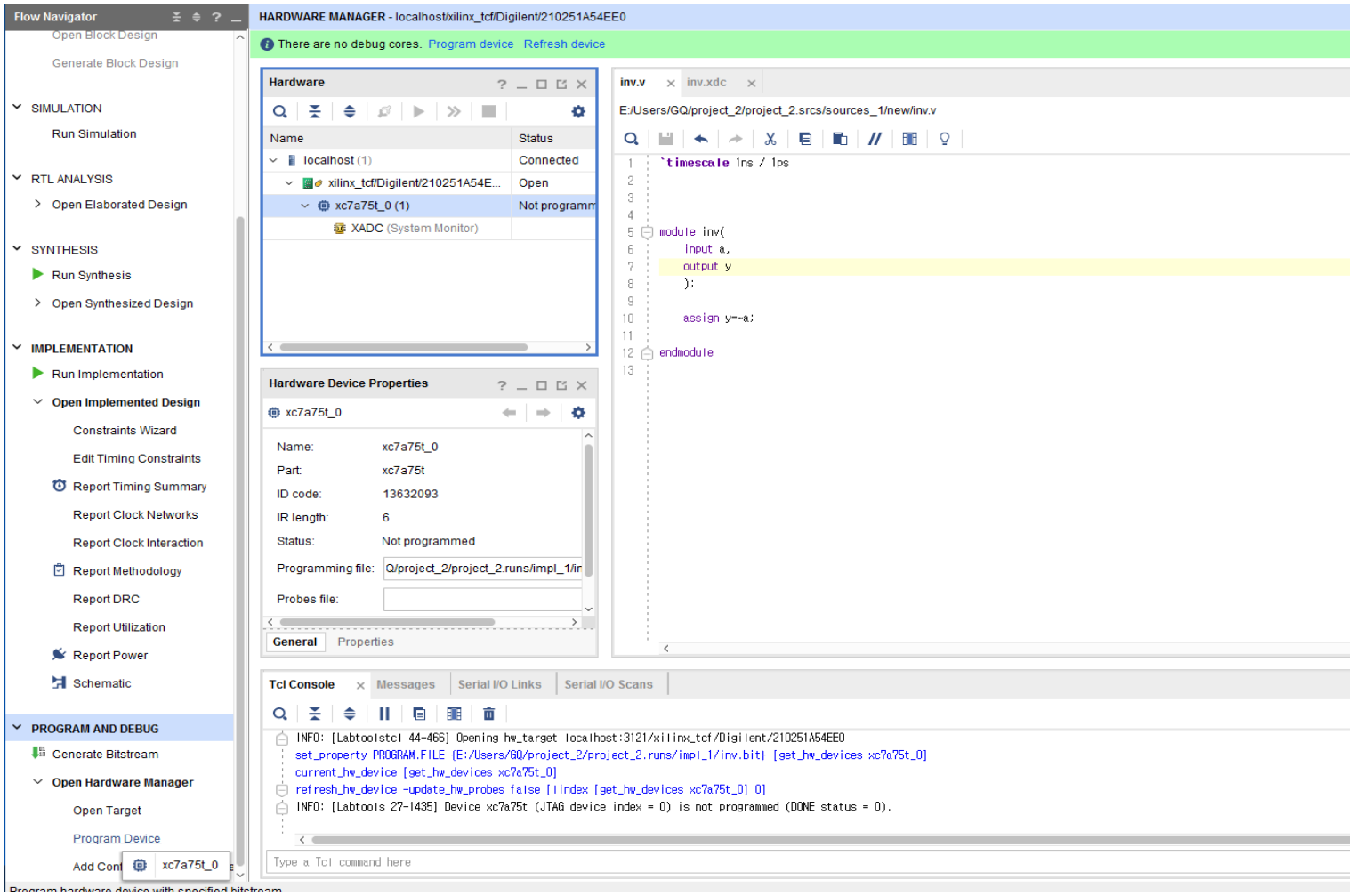
FPGA 동작 검증

- Example(NOT gate)
  - Project 명->project 명_runs->impl_1에 debug_nets.ltx 파일을 넣어줌.



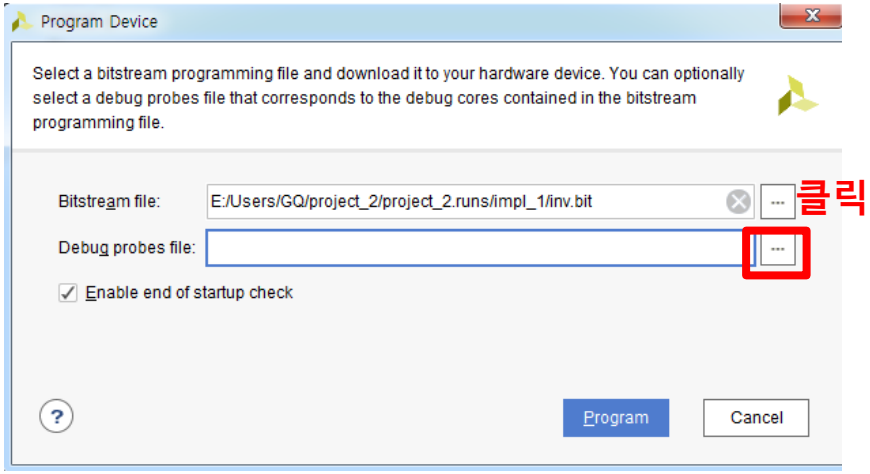
FPGA 동작 검증

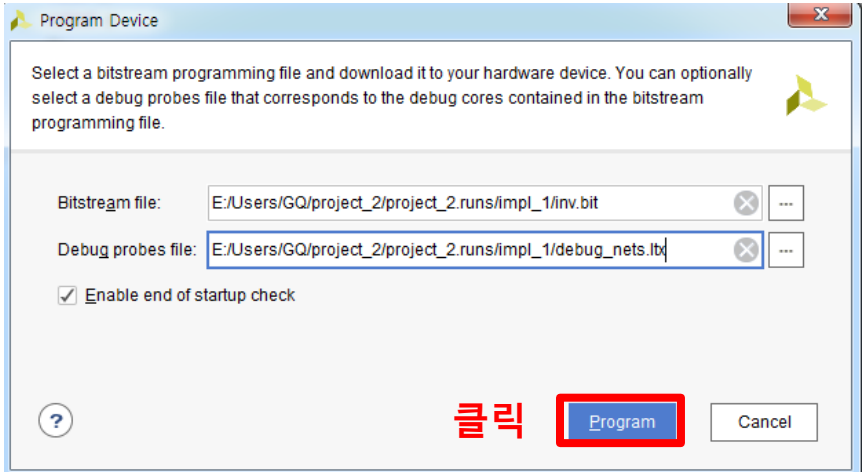
- Example(NOT gate)
  - Program Device -> xc7a75t_0



FPGA 동작 검증

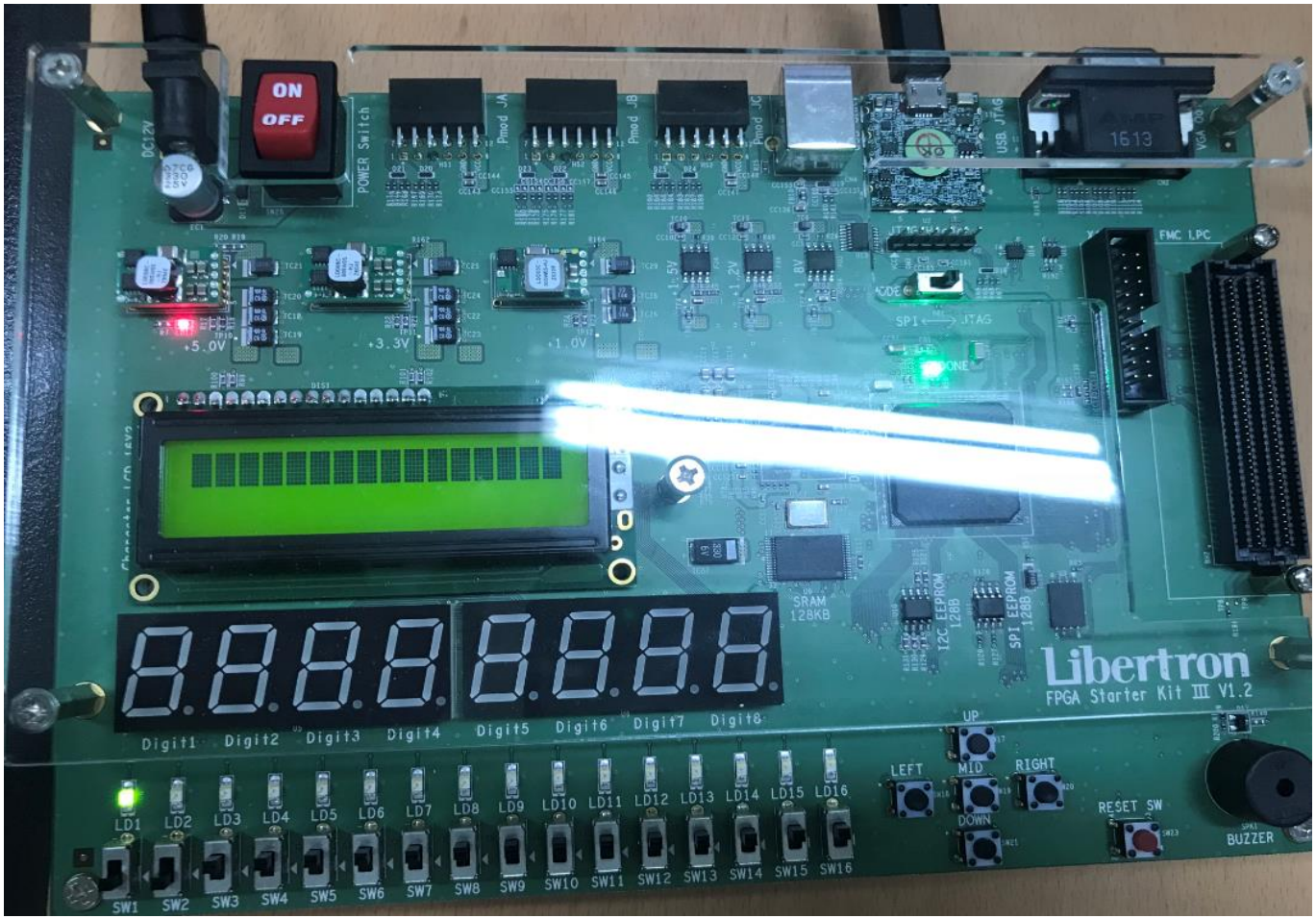
- Example(NOT gate)
  - Program Device





FPGA 동작 검증

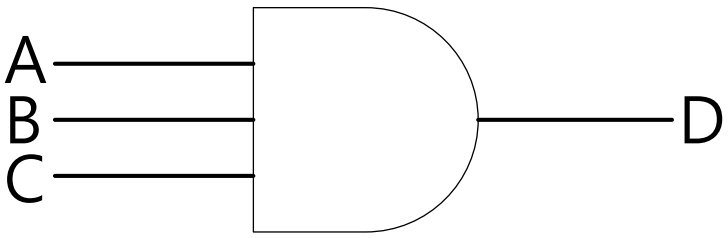
- Example(NOT gate)
  - FPGA 동작 결과



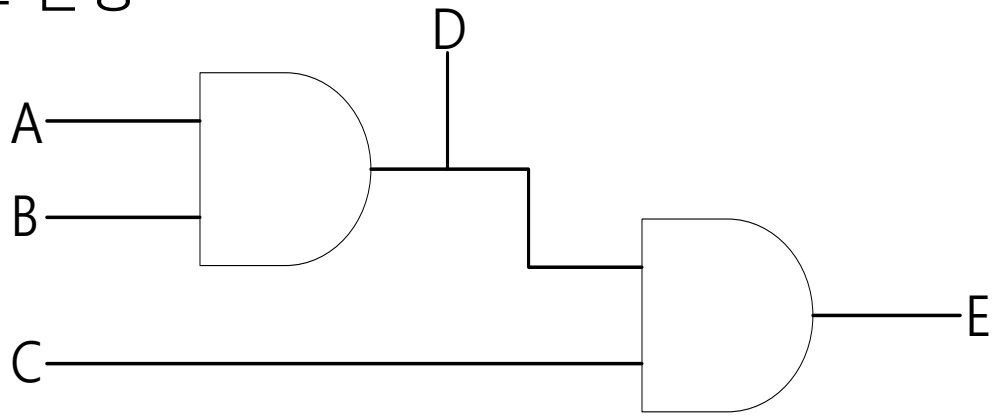
다중입력 AND 게이트(실습)

- 3-input AND gate

  - (A)와 (B)의 Boolean 식을 비교
  - (A)와 (B)의 Verilog 코딩
  - (A)와 (B)의 Simulation을 통해 출력 결과 비교
  - (A)와 (B)의 동작을 FPGA의 동작 시켜 비교
  - 3-input AND gate의 진리표 완성



(A)



(B)

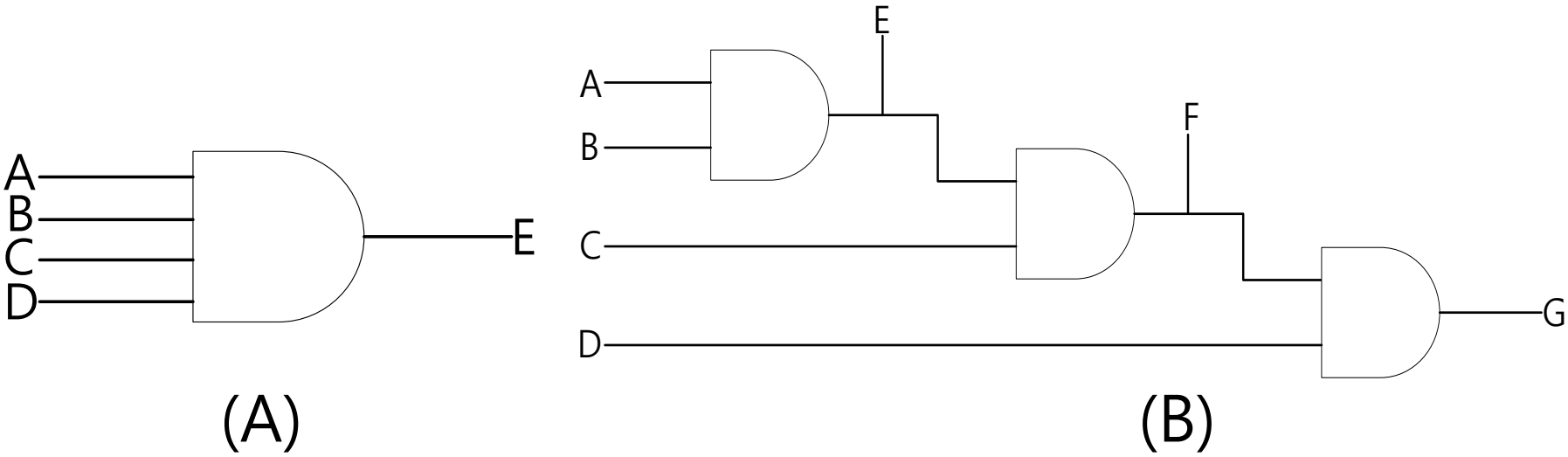
다중입력 AND 게이트(실습)

- 3-input AND gate 진리표

In A	In B	In C	Out D	Out E
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

다중입력 AND 게이트(실습)

- 4-input AND gate
  - (A)와 (B)의 Boolean 식을 비교
  - (A)와 (B)의 Verilog 코딩
  - (A)와 (B)의 Simulation을 통해 출력 결과 비교
  - (A)와 (B)의 동작을 FPGA의 동작 시켜 비교
  - 4-input AND gate의 진리표 완성



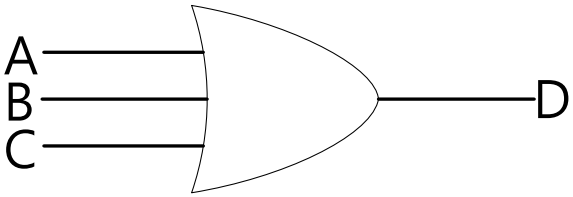
다중입력 AND 게이트(실습)

◆ 4-input AND gate 진리표

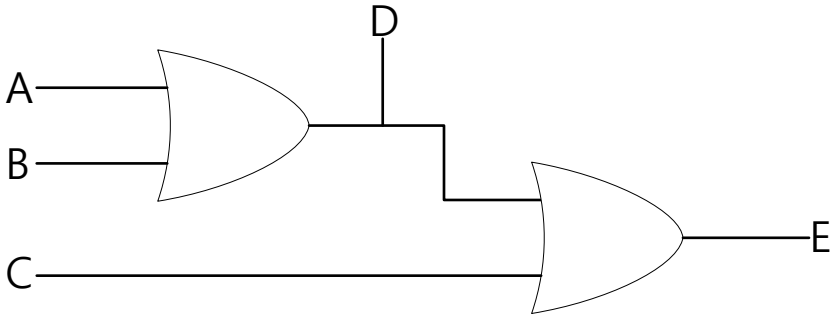
In A	In B	In C	In D	Out E	Out F	Out G
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

다중입력 OR 게이트(실습)

- 3-input OR gate
  - (A)와 (B)의 Boolean 식을 비교
  - (A)와 (B)의 Verilog 코딩
  - (A)와 (B)의 Simulation을 통해 출력 결과 비교
  - (A)와 (B)의 동작을 FPGA의 동작 시켜 비교
  - 3-input OR gate의 진리표 완성



(A)



(B)

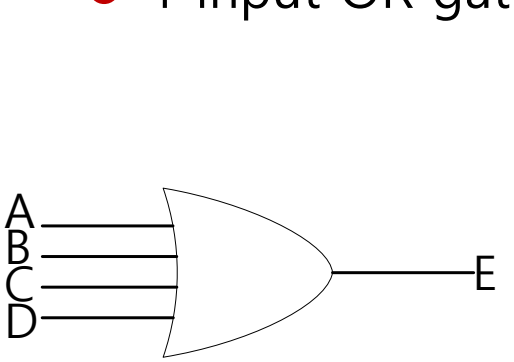
다중입력 OR 게이트(실습)

- 3-input OR gate 진리표

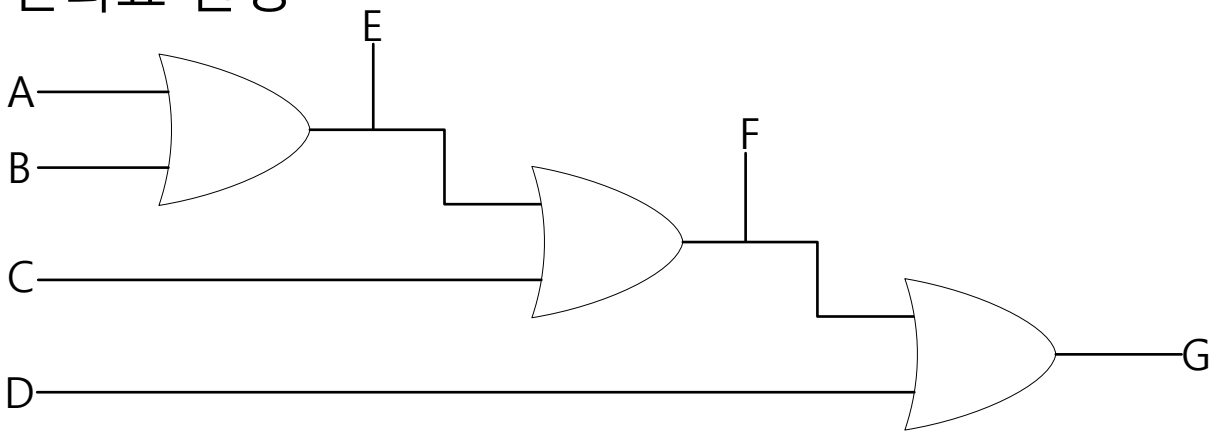
In A	In B	In C	Out D	Out E
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

다중입력 OR 게이트(실습)

- 4-input OR gate
  - (A)와 (B)의 Boolean 식을 비교
  - (A)와 (B)의 Verilog 코딩
  - (A)와 (B)의 Simulation을 통해 출력 결과 비교
  - (A)와 (B)의 동작을 FPGA의 동작 시켜 비교
  - 4-input OR gate의 진리표 완성



(A)



(B)

다중입력 OR 게이트(실습)

◆ 4-input OR gate 진리표

In A	In B	In C	In D	Out E	Out F	Out G
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			