

# 컴퓨터공학실험II

## 3장 AND/OR/NOT



*Be as proud of Sogang  
As Sogang is proud of you*

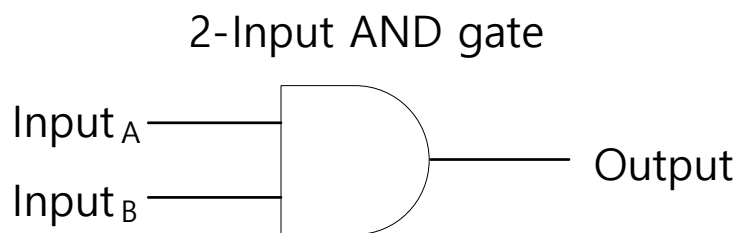
- ◆ AND/OR/NOT Gate의 동작의 이해 및 확인
- ◆ Verilog를 사용하여 다중입력 AND/OR Gate 및 NOT Gate 구현
- ◆ 입력 신호 생성 후 Simulation을 통하여 구현된 각 Gate 동작 확인
- ◆ FPGA 통해서 Verilog로 구현된 회로의 동작 확인

- ◆ 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

◆ 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

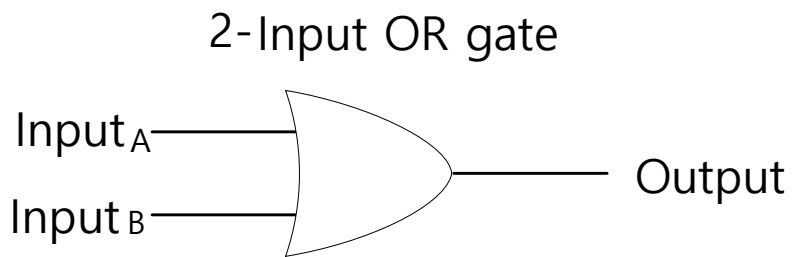
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

2-input AND gate Truth Table

◆ 2개 이상의 신호를 받으며, 두 신호 중 하나라도 High일 때는 High값을, High값이 하나도 없을 경우 Low값을 출력

◆ Boolean 식에서 덧셈으로 표현

$$C = A + B$$

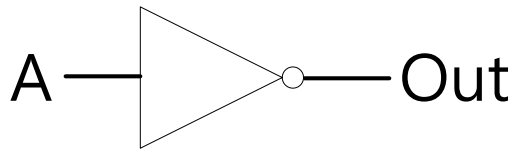


2-input OR gate Schematic

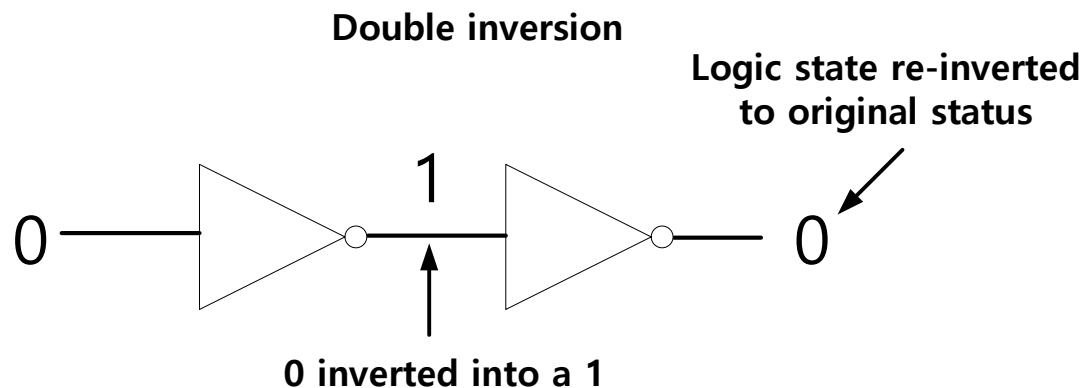
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

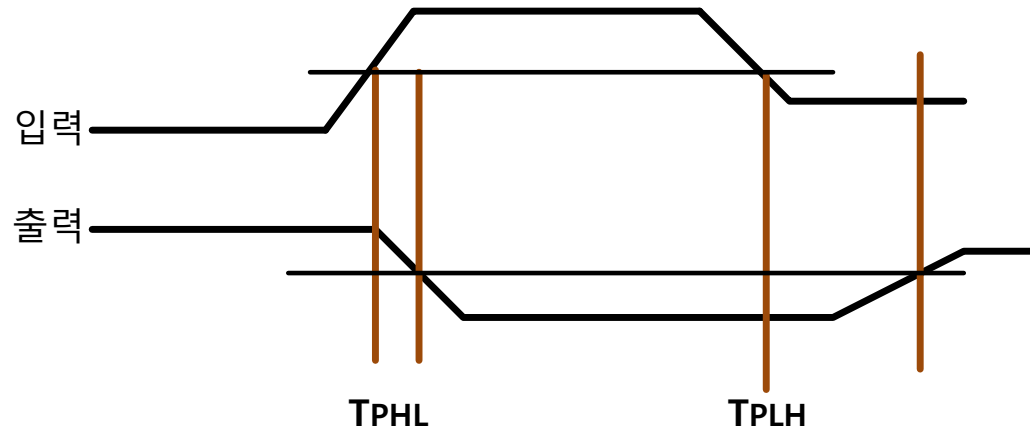
2-input OR gate Truth Table

- ◆ NOT 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





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

- ◆ 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)  $x(x+y)=x$

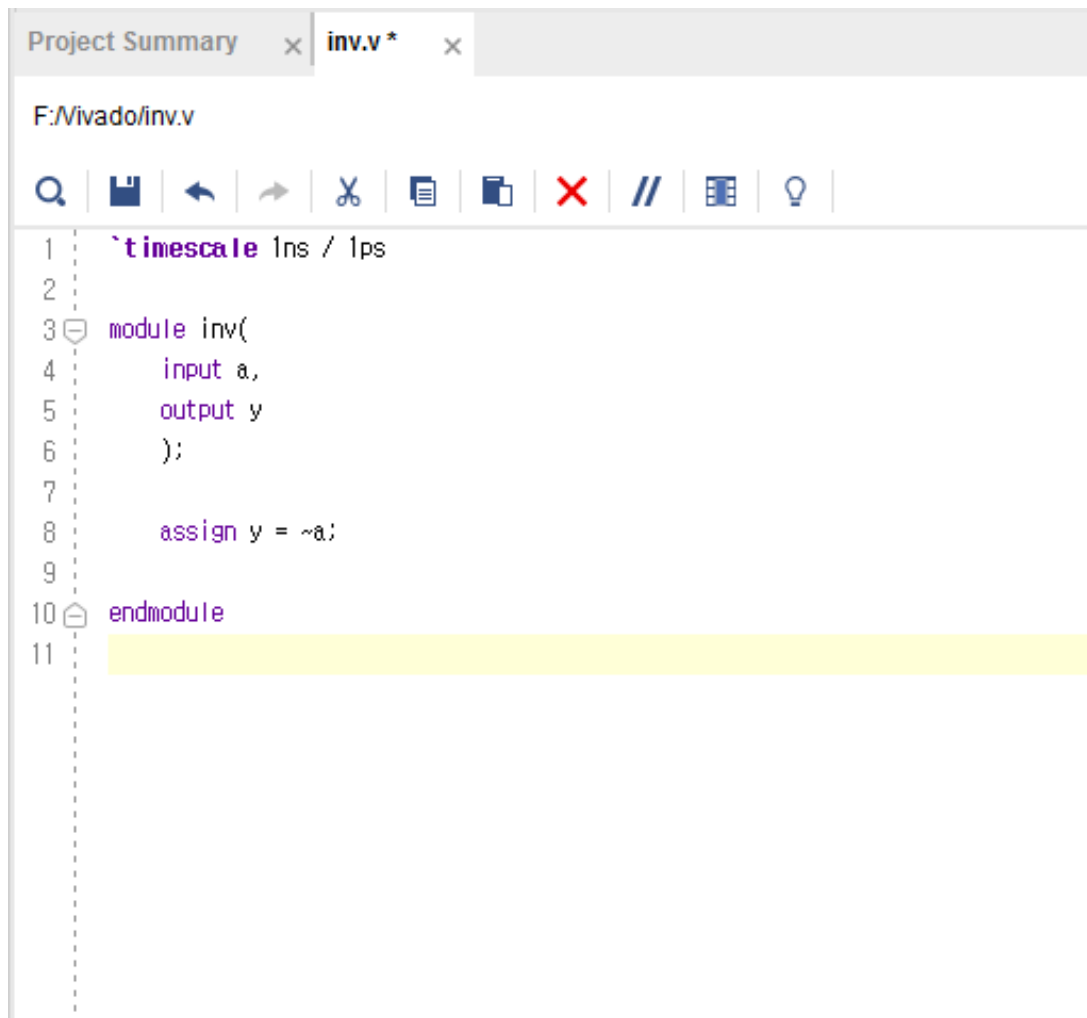


## ◆ 동작검증 단계

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

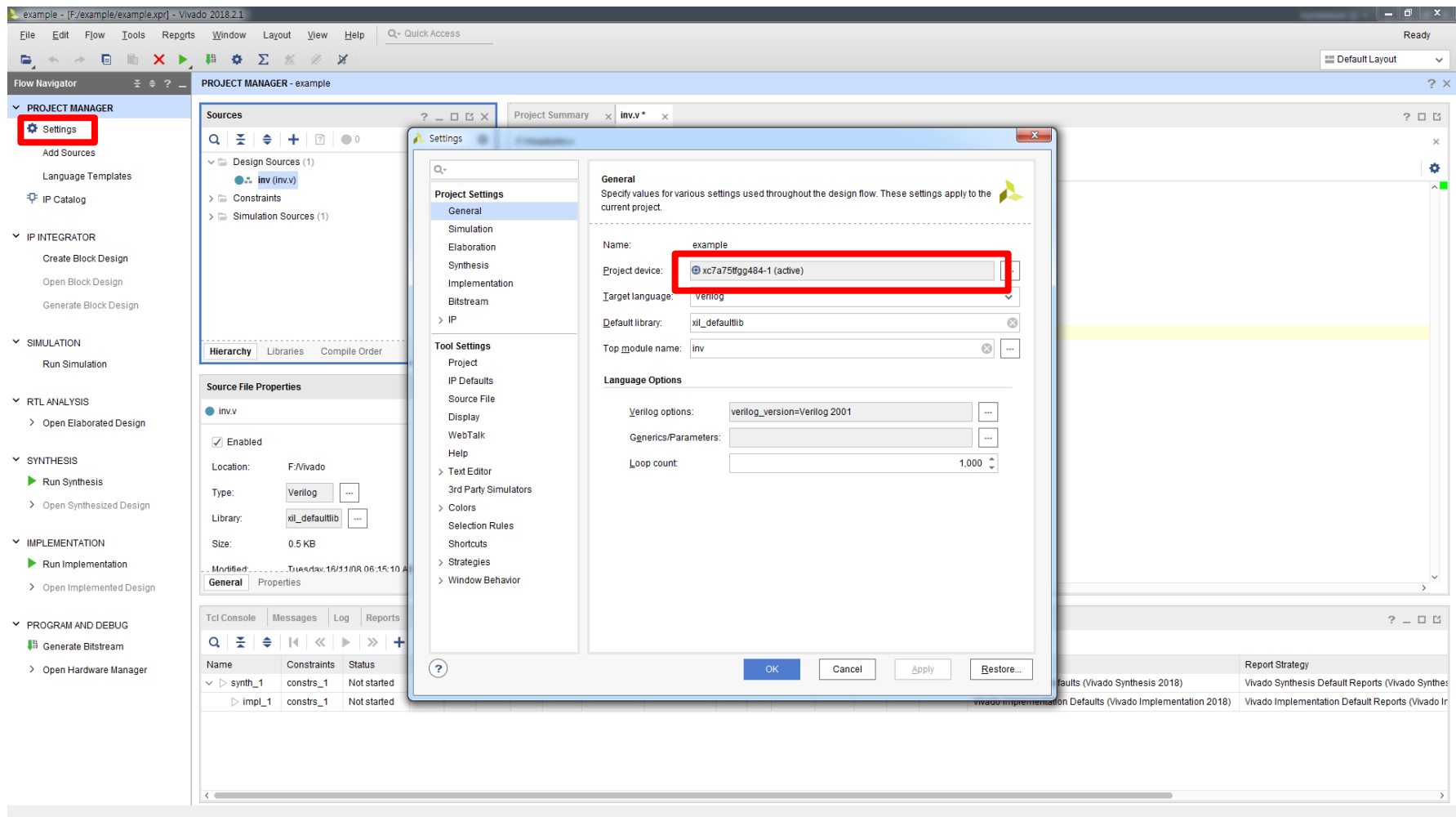
## ◆ Example(NOT gate)

- Verilog coding



```
Project Summary x inv.v * x
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 |
```

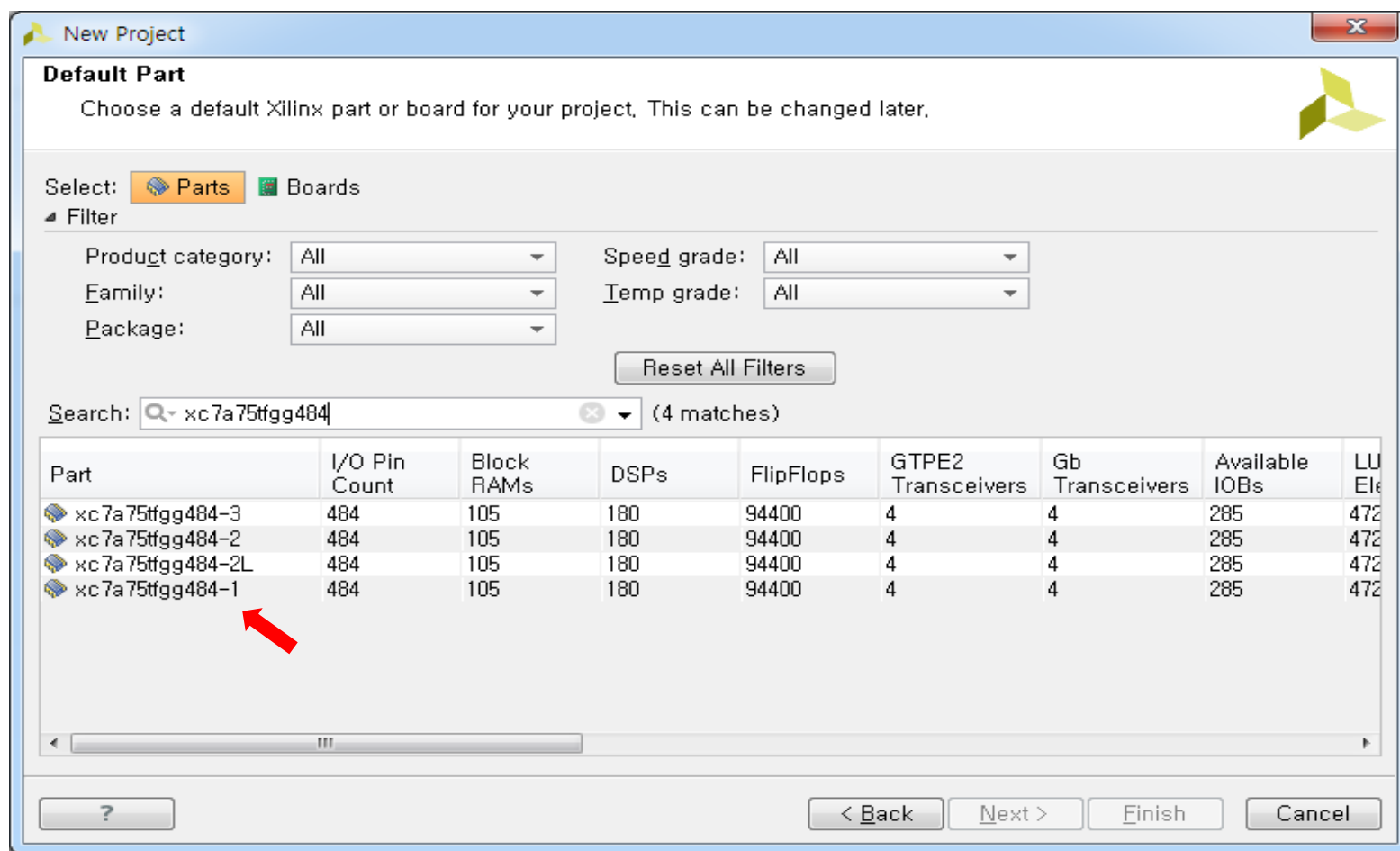
- ◆ Example(NOT gate)
- Device assignment



## ◆ Example(NOT gate)

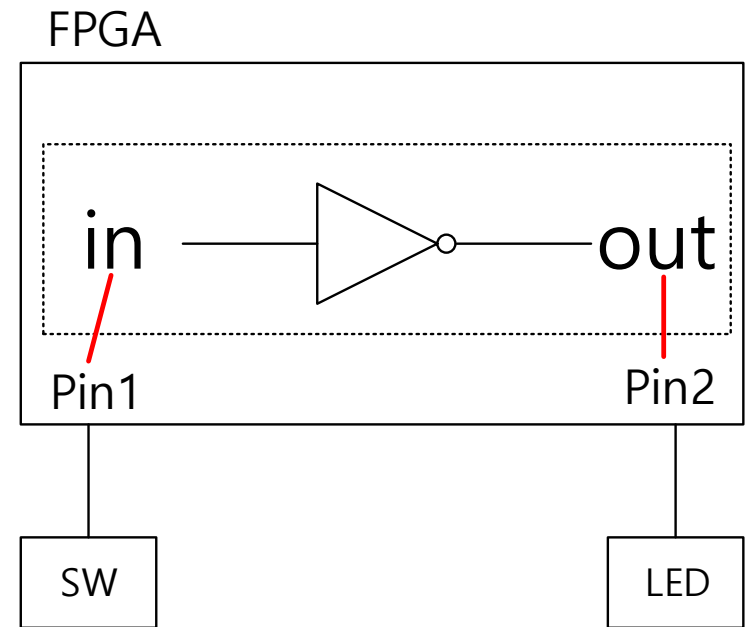
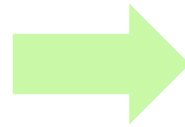
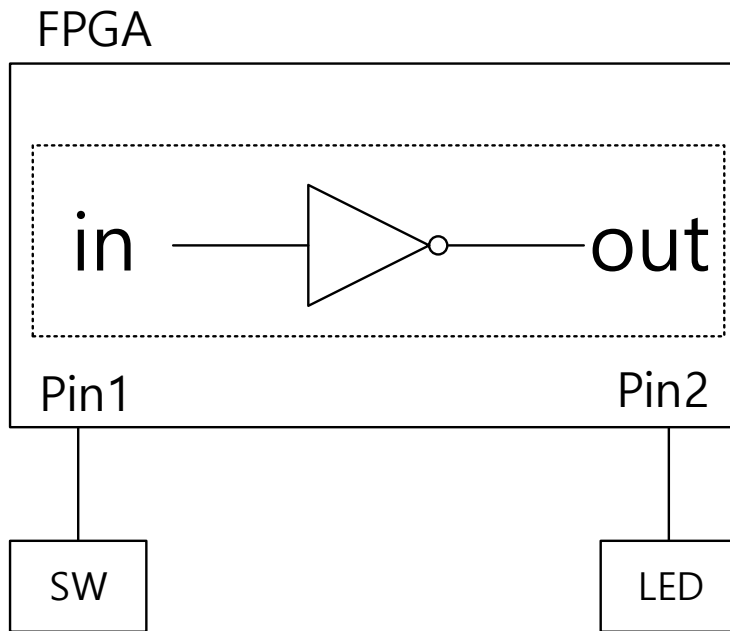
### ● Device assignment

✓ Device : **xc7a75tfgg484(Artix7)**



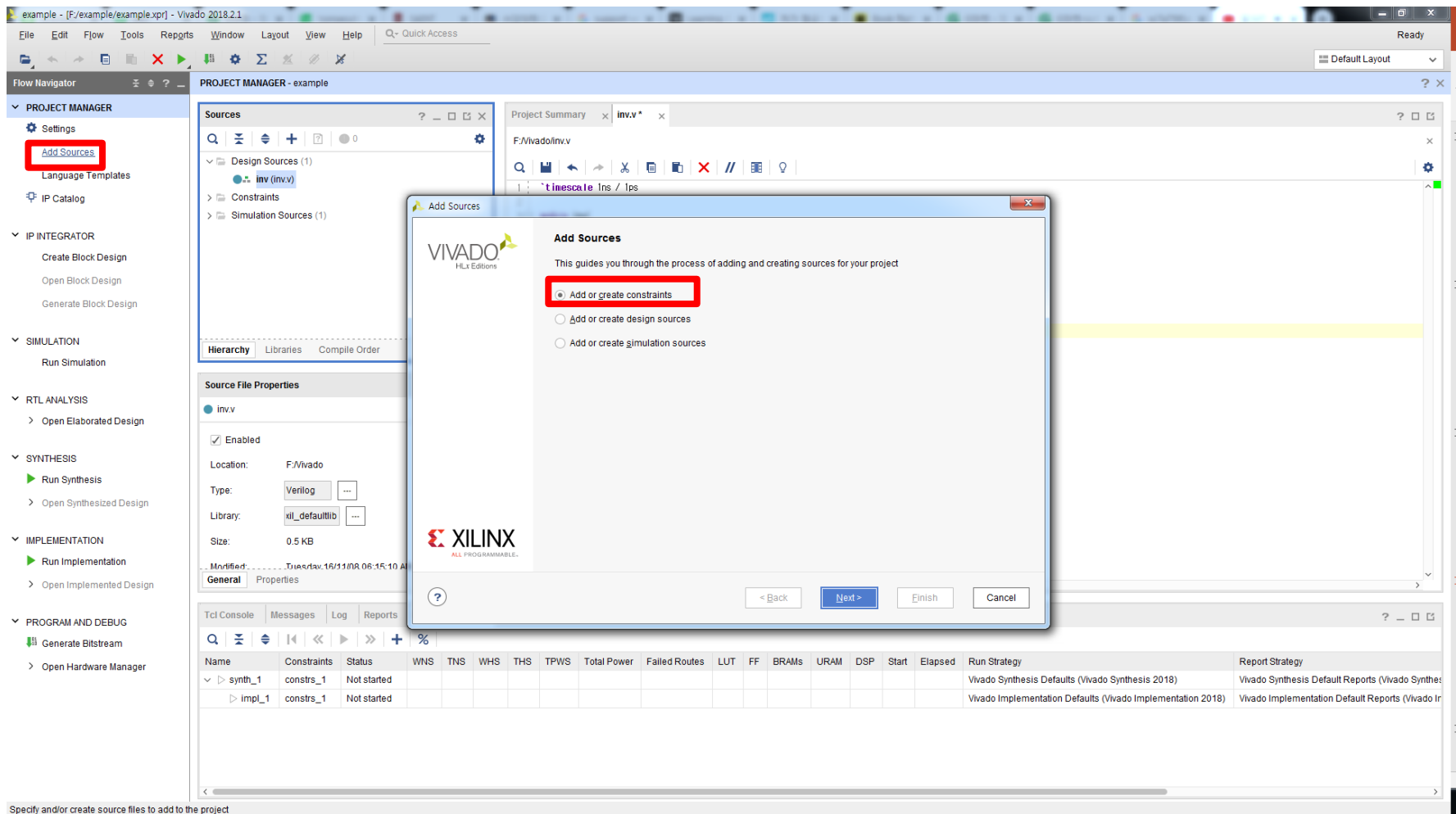
## ◆ Example(NOT gate)

- Pin assignment



## ◆ Example(NOT gate)

### ● Pin assignment



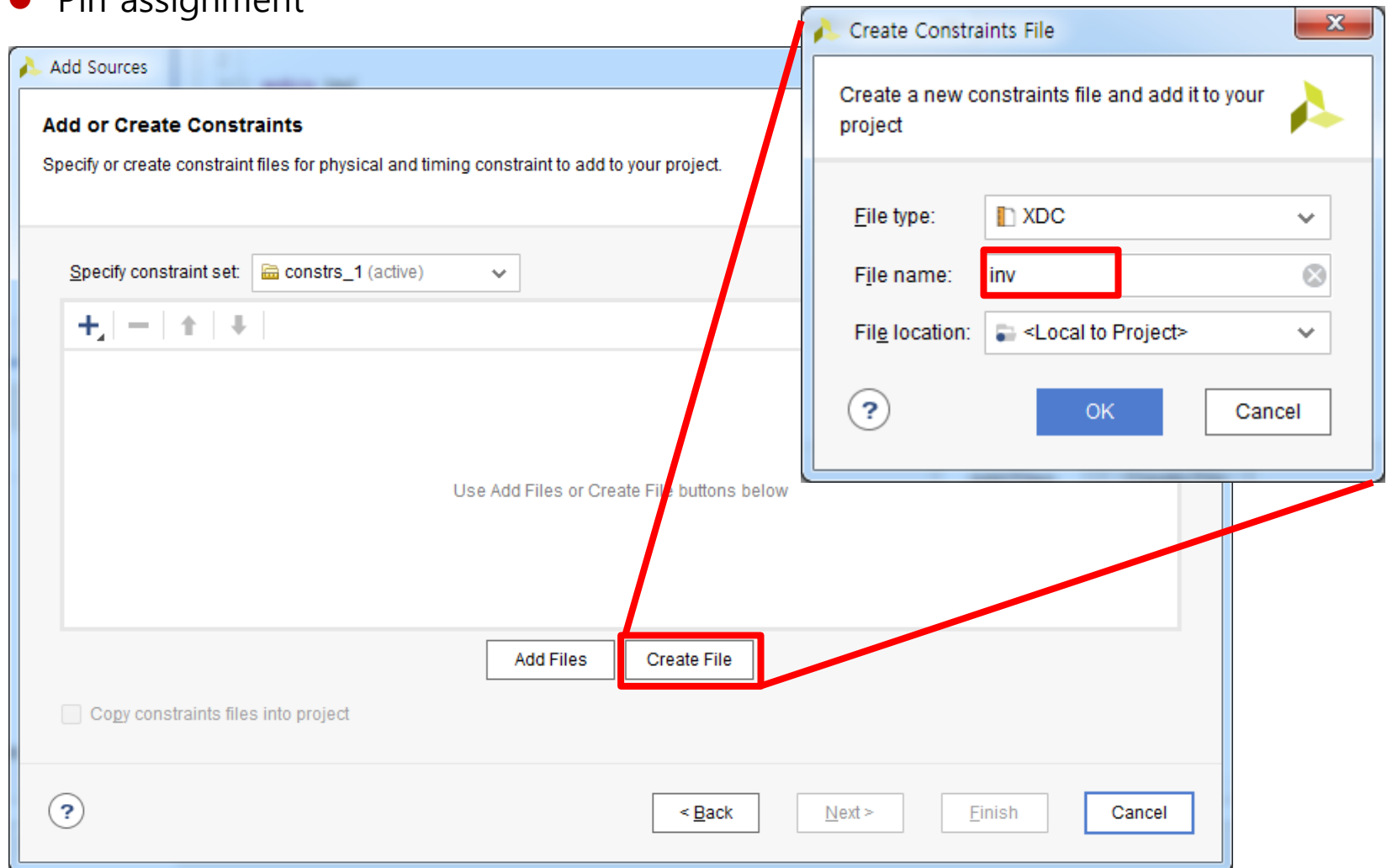
The screenshot shows the Xilinx Vivado 2018.2.1 interface. The 'Add Sources' dialog box is open, guiding the user through adding sources to the project. The 'Add or create constraints' option is selected and highlighted with a red box. The background shows the project manager with 'inv.v' as the design source and a table of synthesis results.

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2018)	Vivado Synthesis Default Reports (Vivado Synthesis 2018)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementation 2018)	Vivado Implementation Default Reports (Vivado Implementation 2018)

Specify and/or create source files to add to the project

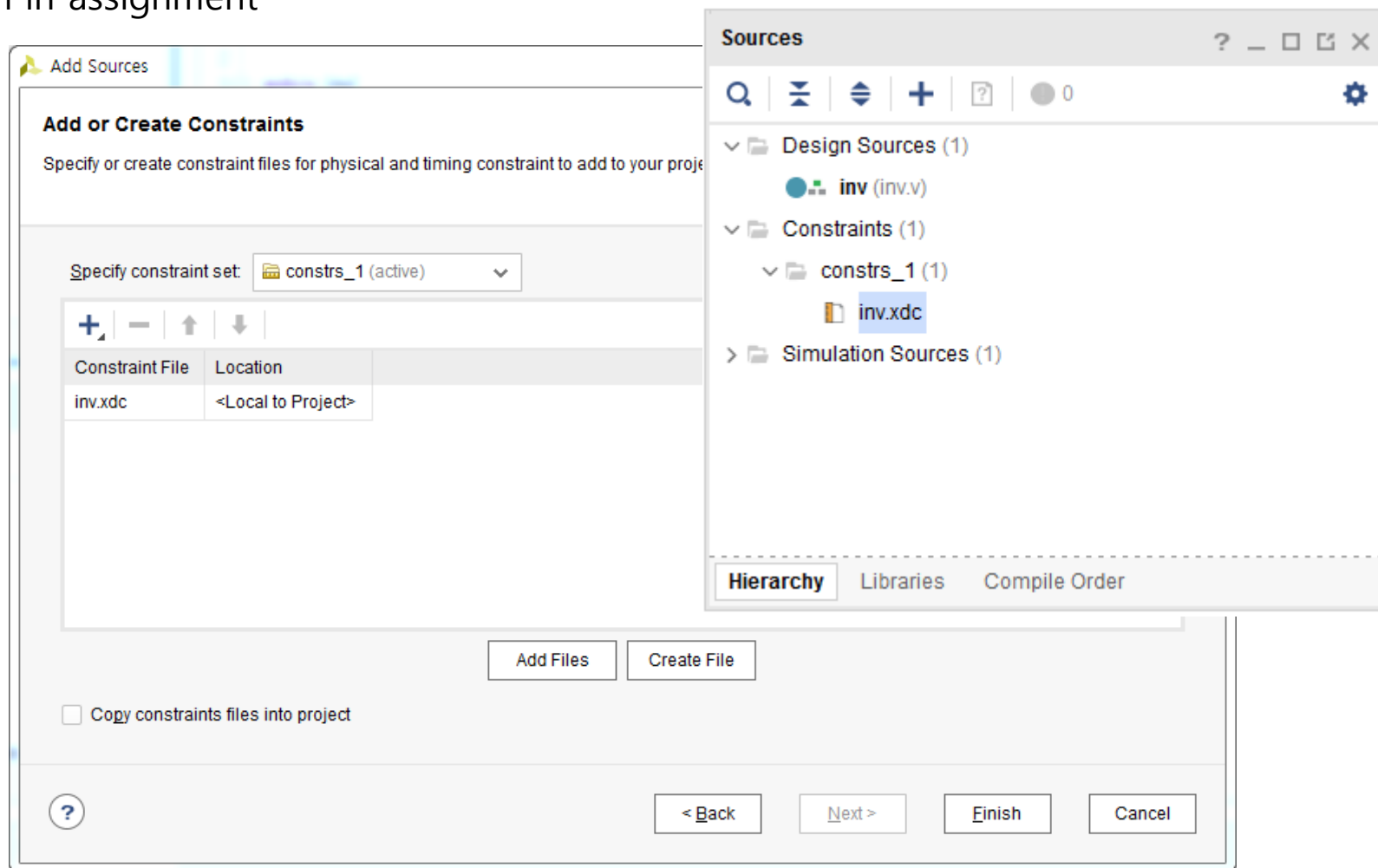
## ◆ Example(NOT gate)

- Pin assignment



## ◆ Example(NOT gate)

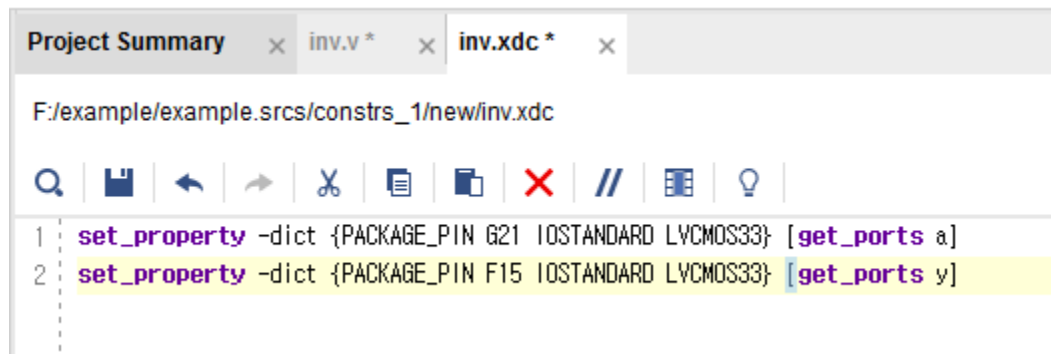
- Pin assignment





## ◆ Example(NOT gate)

- Pin assignment
  - FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



The screenshot shows an IDE window with three tabs: 'Project Summary', 'inv.v \*', and 'inv.xdc \*'. The active tab is 'inv.xdc \*', which contains the following code:

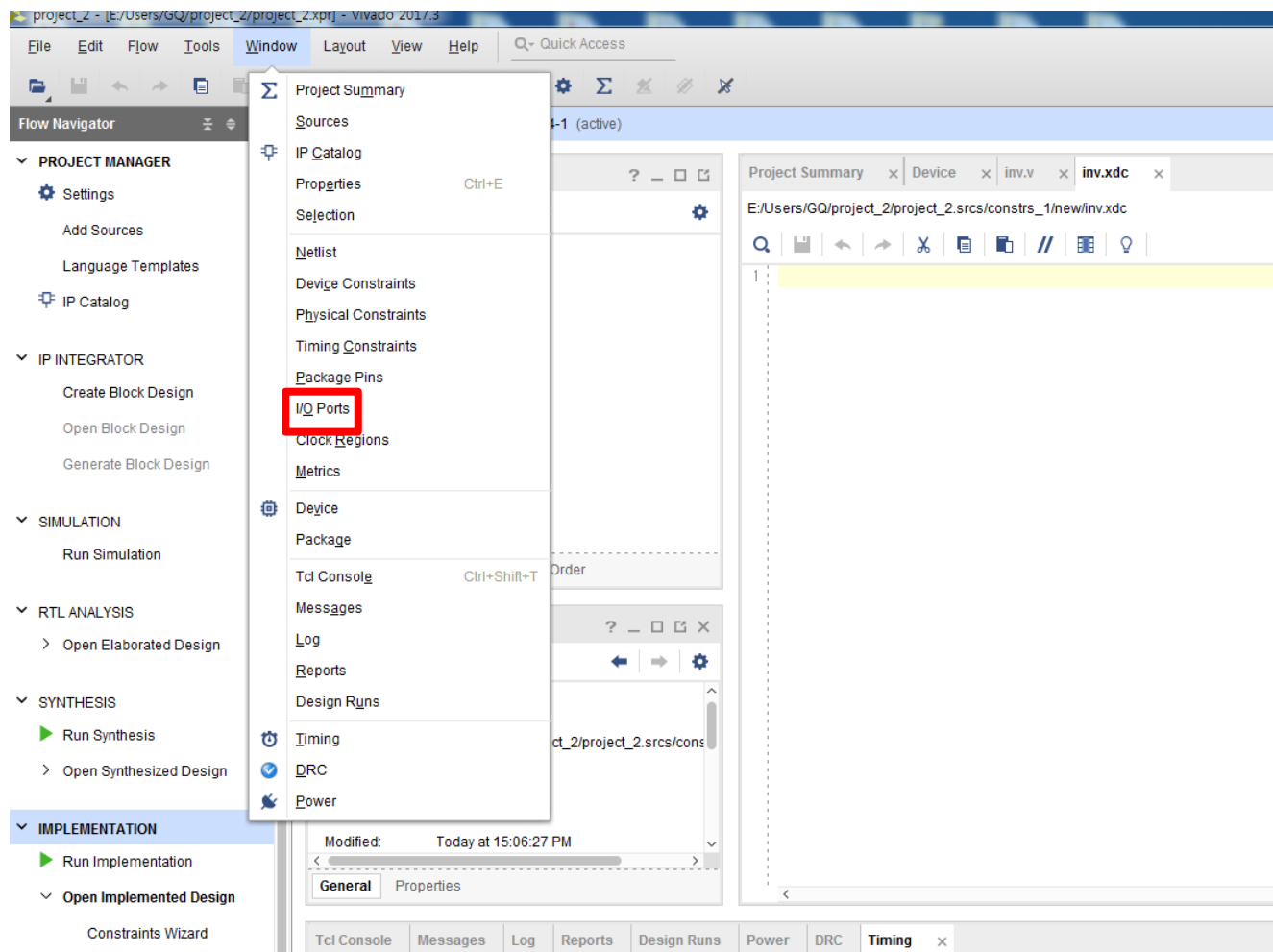
```
F:/example/example.srcs/constrs_1/new/inv.xdc

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

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

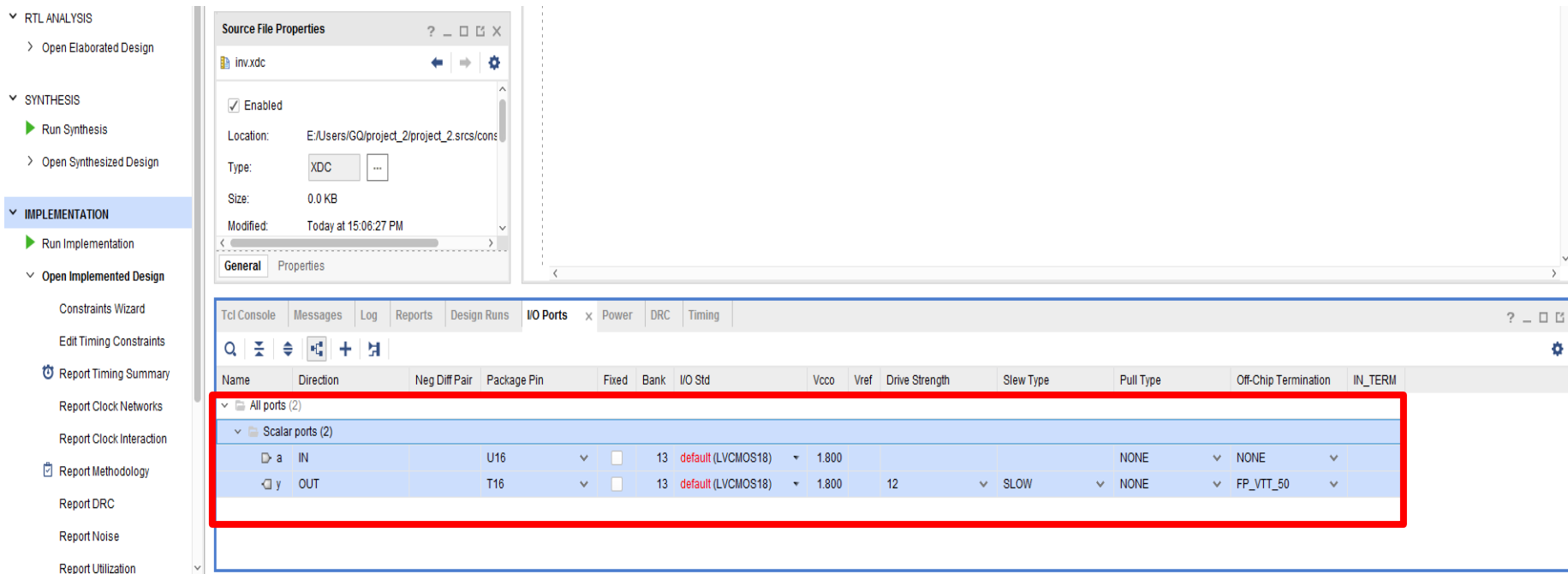
## ◆ Example(NOT gate)

### ● Pin assignment



## ◆ Example(NOT gate)

- Pin assignment
  - FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



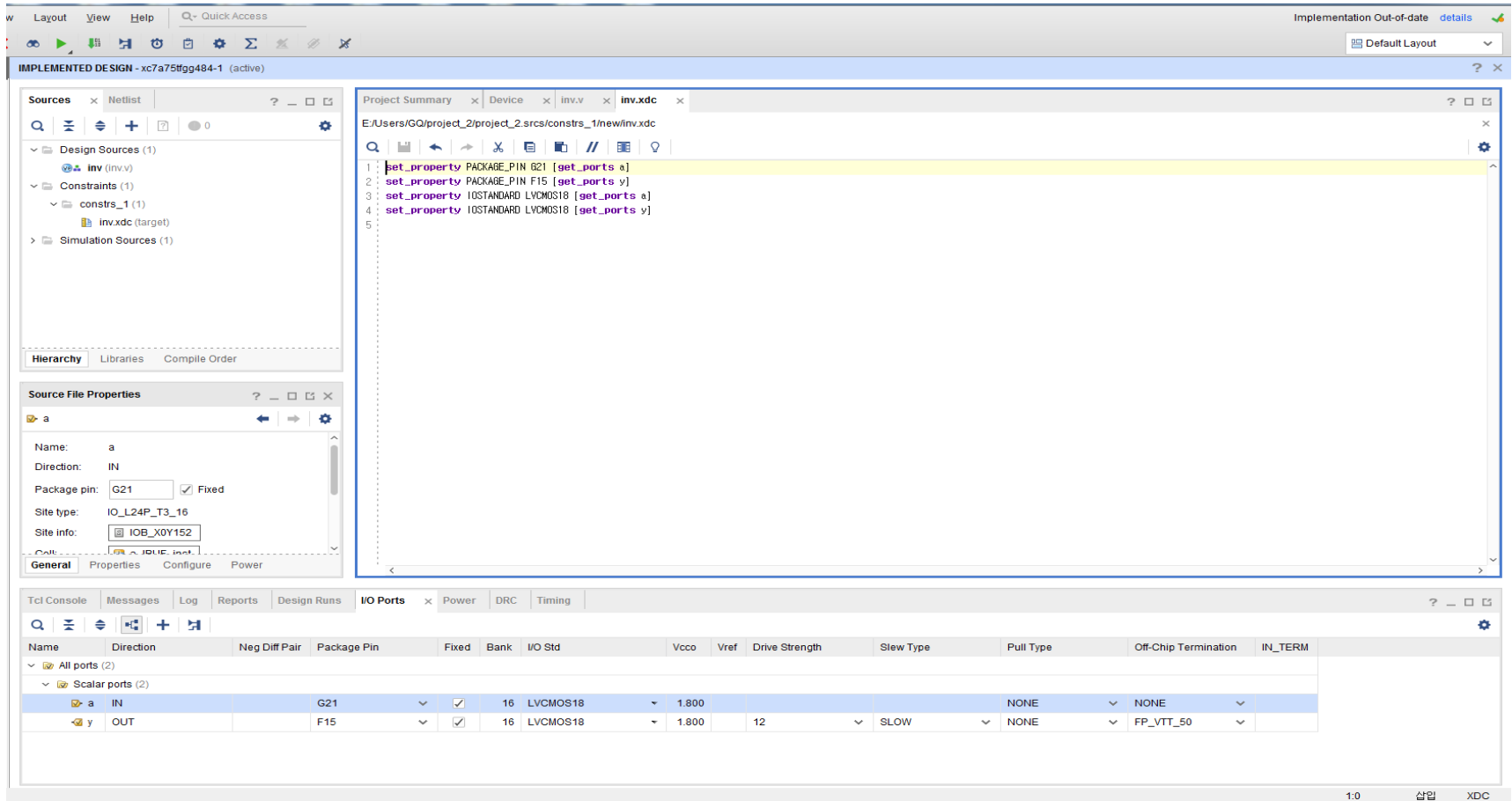
The screenshot shows the Xilinx IDE interface. On the left, the 'IMPLEMENTATION' tab is selected, with 'Open Implemented Design' chosen. The 'Source File Properties' window is open for 'inv.xdc', showing it is enabled and located at 'E:/Users/GQ/project\_2/srcs/cons...'. The 'IO Ports' window is also open, displaying a table of scalar ports. A red box highlights the 'Scalar ports (2)' section of the table.

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
▼ All ports (2)													
▼ Scalar ports (2)													
a	IN		U16	▼		13	default (LVCMOS18)	▼	1.800		NONE	▼	NONE
y	OUT		T16	▼		13	default (LVCMOS18)	▼	1.800	12	▼	SLOW	▼
											▼	NONE	▼
											▼	FP_VTT_50	▼

## ◆ Example(NOT gate)

### ● Pin assignment

- FPGA pin list에서 할당하고 싶은 Pin과 Verilog 소스의 port를 링크



The screenshot shows the Xilinx Vivado IDE interface for a project named 'xc7a75fpg484-1'. The main editor displays the 'inv.xdc' file with the following Verilog code:

```
1 set_property PACKAGE_PIN G21 [get_ports a]
2 set_property PACKAGE_PIN F15 [get_ports y]
3 set_property IOSTANDARD LVCMOS18 [get_ports a]
4 set_property IOSTANDARD LVCMOS18 [get_ports y]
```

The 'Source File Properties' window for port 'a' is open, showing the following configuration:

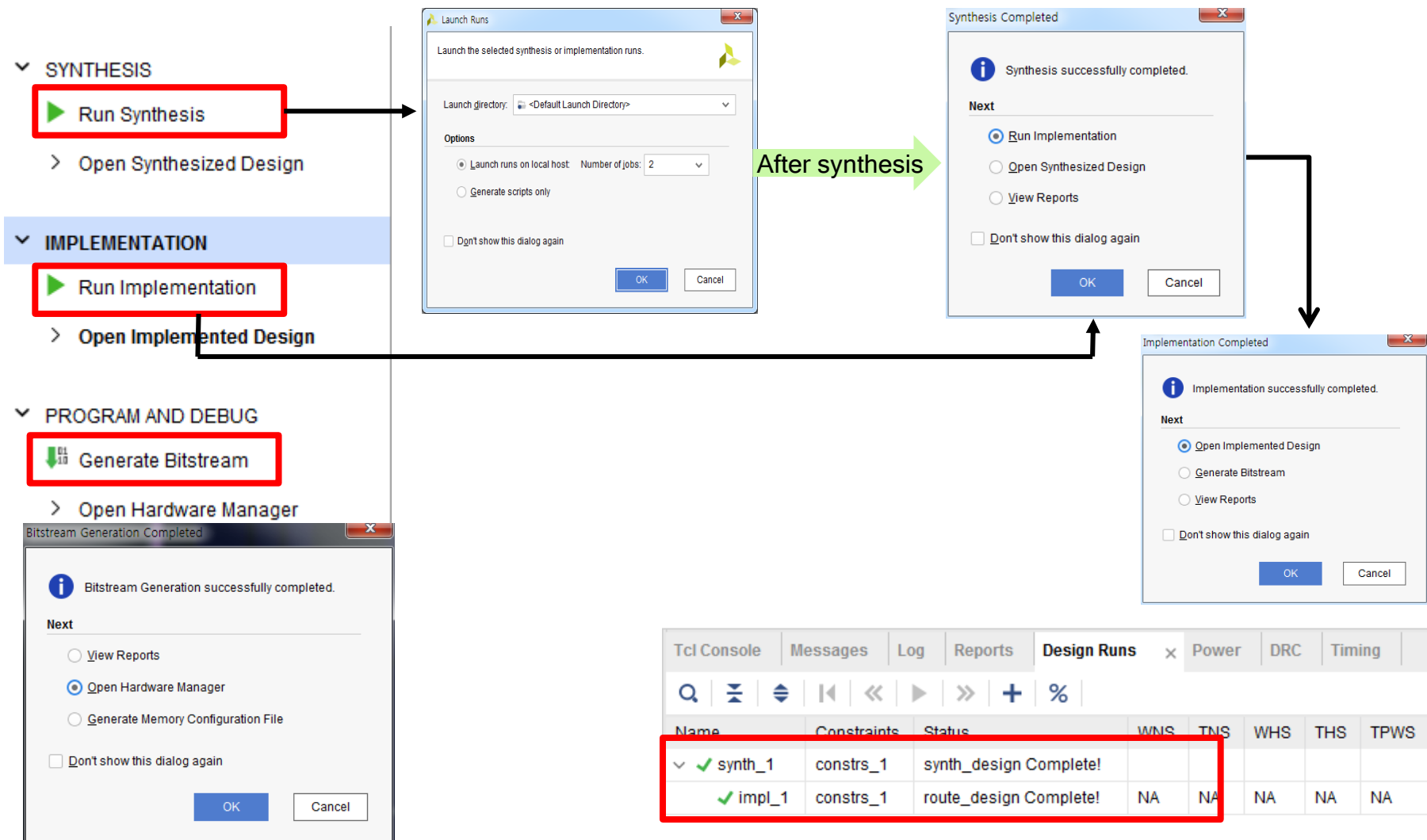
- Name: a
- Direction: IN
- Package pin: G21 (Fixed)
- Site type: IO\_L24P\_T3\_16
- Site info: IOB\_X0Y152

The 'IO Ports' window at the bottom shows the pin assignment table:

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
a	IN		G21	<input checked="" type="checkbox"/>	16	LVCMOS18	1.800				NONE	NONE	
y	OUT		F15	<input checked="" type="checkbox"/>	16	LVCMOS18	1.800	12		SLOW	NONE	FP_VTT_50	

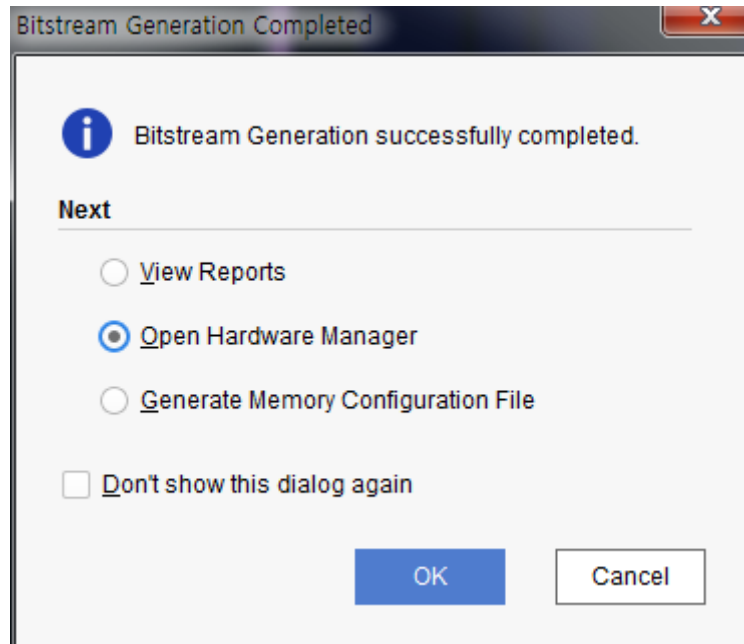
## ◆ Example(NOT gate)

### ● Synthesis / Implementation



After synthesis

- ◆ Example(NOT gate)
  - Device configuration

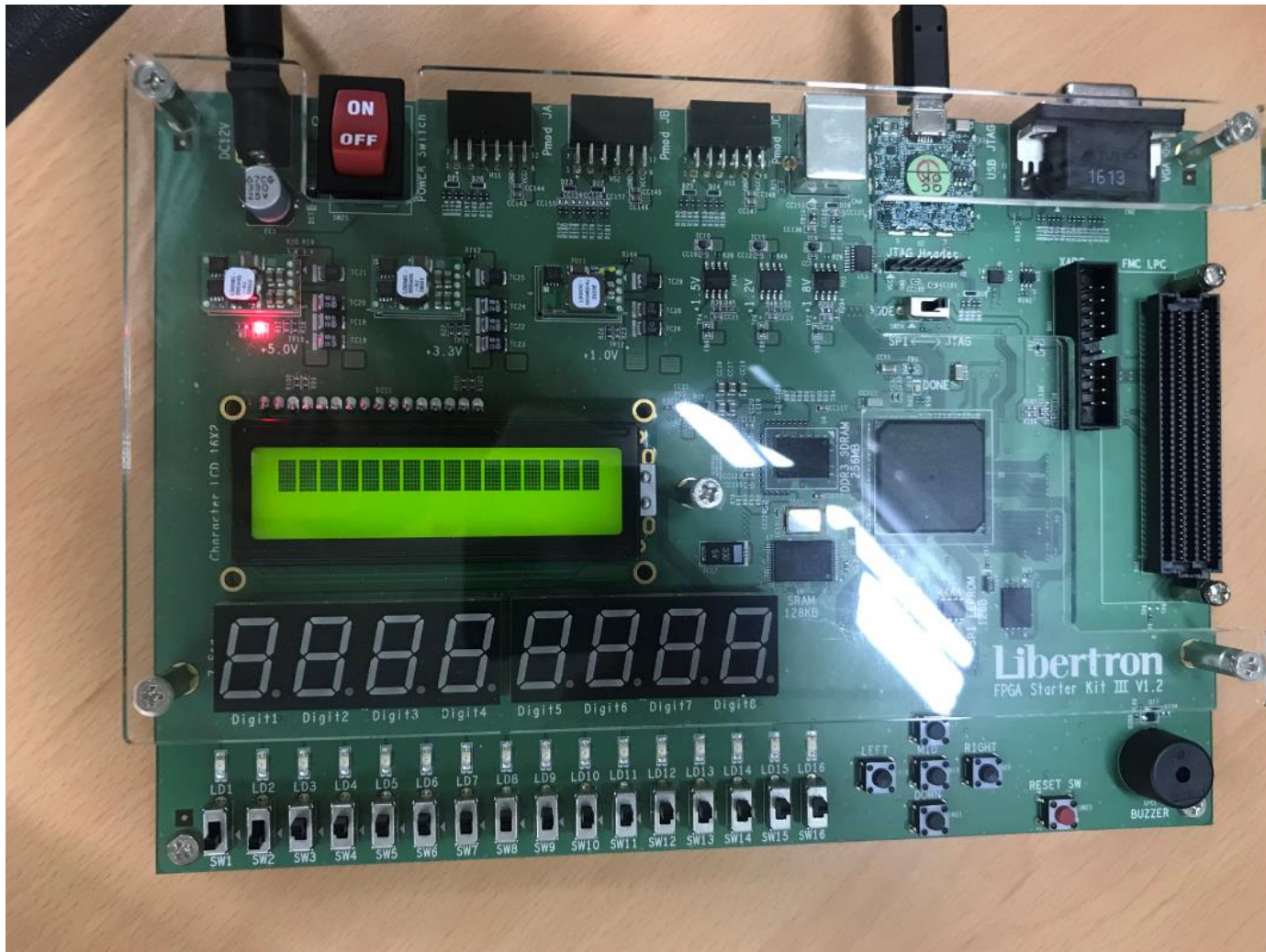


- ◆ 장비 : FPGA Starter Kit III Board
- ◆ Cable : Micro USB Cable





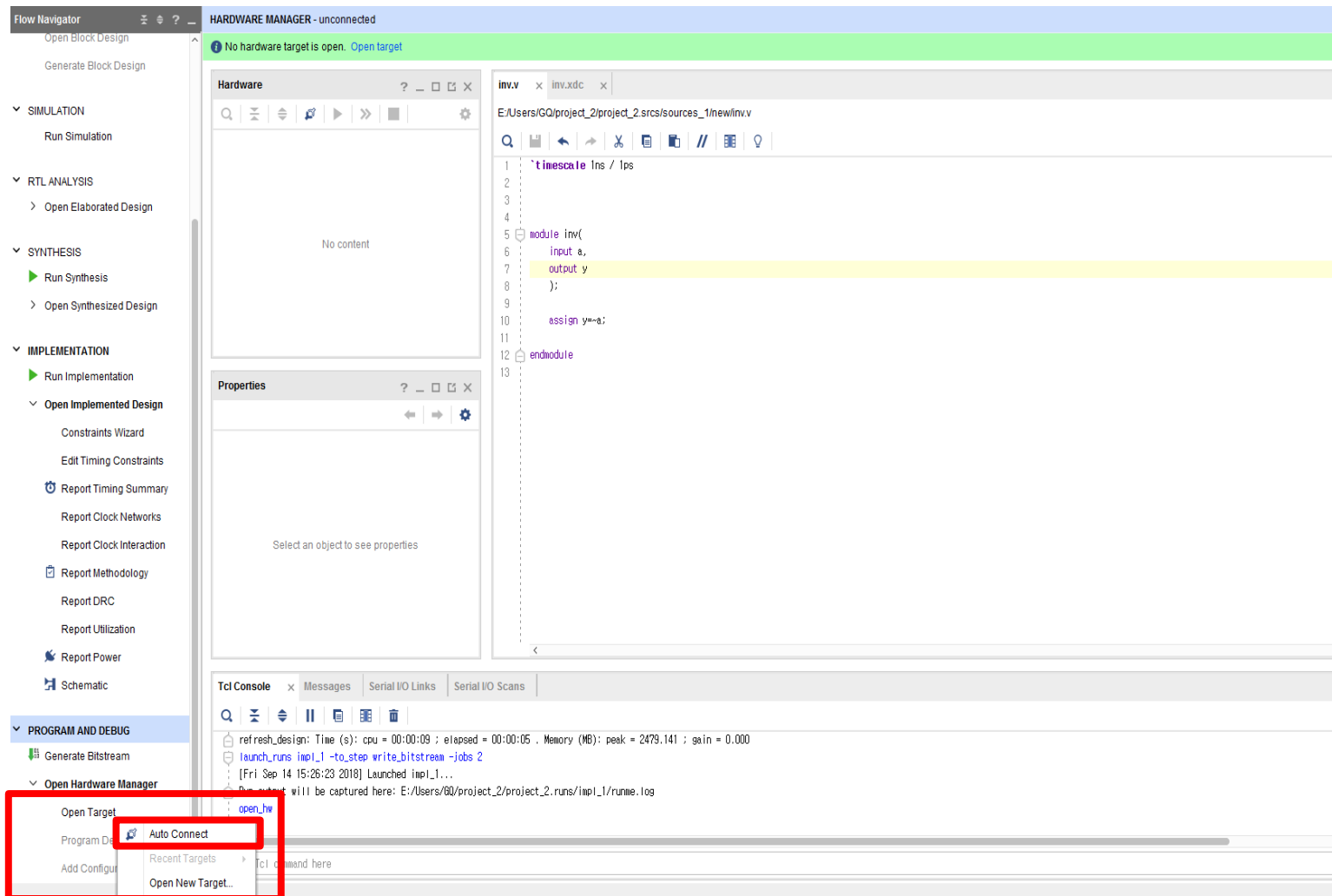
## ◆ FPGA Starter Kit III Board 초기 상태





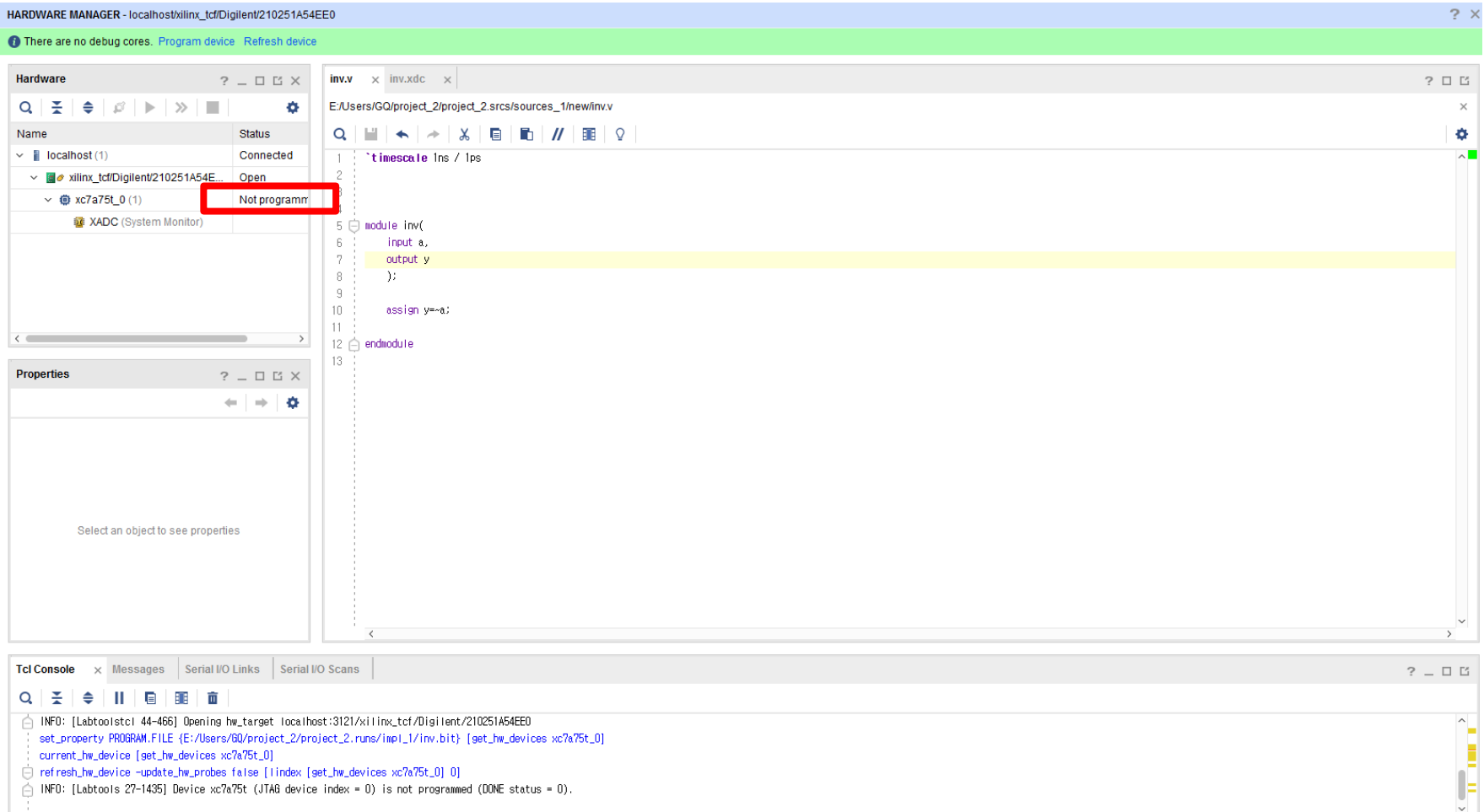
## ◆ Example(NOT gate)

- Open Target -> auto connect 클릭



## ◆ Example(NOT gate)

- Open Target -> auto connect 클릭



The screenshot displays the Xilinx Hardware Manager and IDE interface. The Hardware Manager on the left shows the target device 'xc7a75t\_0 (1)' with a status of 'Not programmed', which is highlighted by a red box. The IDE on the right shows the Verilog code for a NOT gate (inverter) implementation:

```

1  *timescale 1ns / 1ps
2
3
4
5  module inv(
6      input a,
7      output y
8  );
9
10     assign y=~a;
11
12 endmodule
13

```

The Tcl Console at the bottom shows the following log messages:

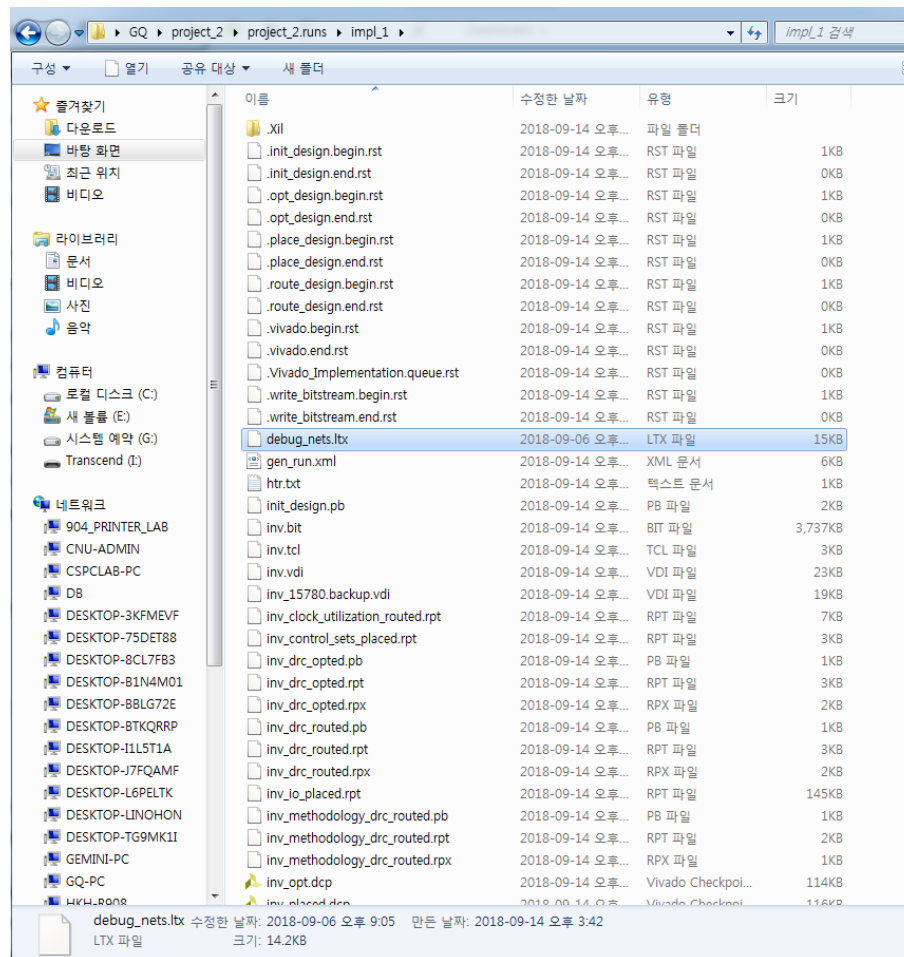
```

INFO: [Labtools] 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digilent/210251A54EE0
set_property PROGRAM.FILE {E:/Users/GQ/project_2/project_2.runs/impl_1/inv.bit} [get_hw_devices xc7a75t_0]
current_hw_device [get_hw_devices xc7a75t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a75t_0] 0]
INFO: [Labtools 27-1435] Device xc7a75t (JTAG device index = 0) is not programmed (DONE status = 0).

```

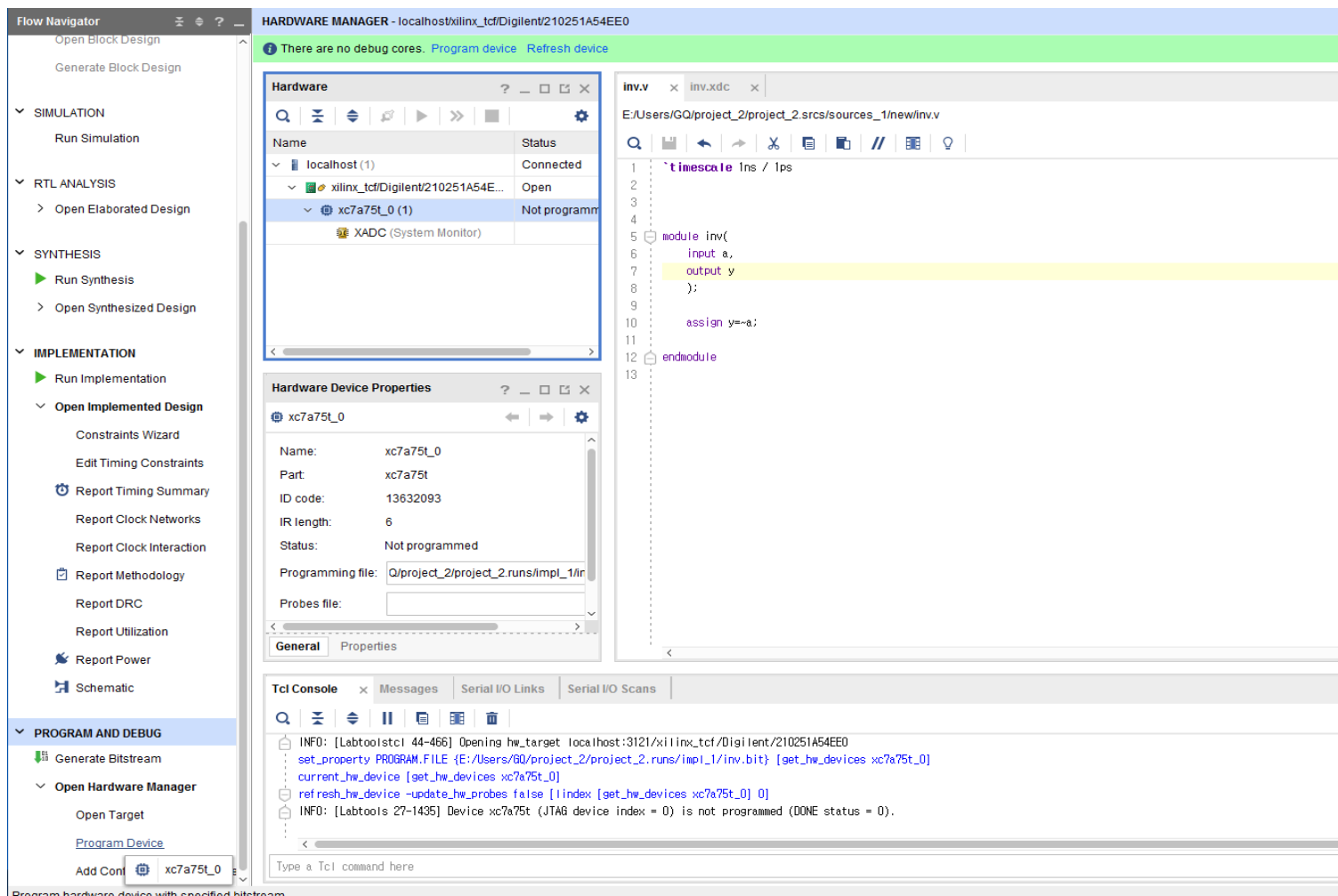
## ◆ Example(NOT gate)

- Project 명->project 명\_runs->impl\_1에 debug\_nets.ltx 파일을 넣어줌.



## ◆ Example(NOT gate)

- Program Device -> xc7a75t\_0



The screenshot displays the Xilinx Vivado IDE interface. On the left is the **Flow Navigator** pane with the **IMPLEMENTATION** section expanded, showing options like **Run Implementation** and **Open Implemented Design**. The **PROGRAM AND DEBUG** section is also visible, with **Open Hardware Manager** selected.

The central **HARDWARE MANAGER** pane shows a table of hardware devices:

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210251A54EE0	Open
xc7a75t_0 (1)	Not programmed
XADC (System Monitor)	

Below this table, the **Hardware Device Properties** for **xc7a75t\_0** are shown:

- Name: xc7a75t\_0
- Part: xc7a75t
- ID code: 13632093
- IR length: 6
- Status: Not programmed
- Programming file: Q:/project\_2/project\_2.runs/impl\_1/inv
- Probes file:

The right pane shows the source code for **inv.v**:

```

1  `timescale 1ns / 1ps
2
3
4
5  module inv(
6      input a,
7      output y
8  );
9
10     assign y=~a;
11
12 endmodule
13
  
```

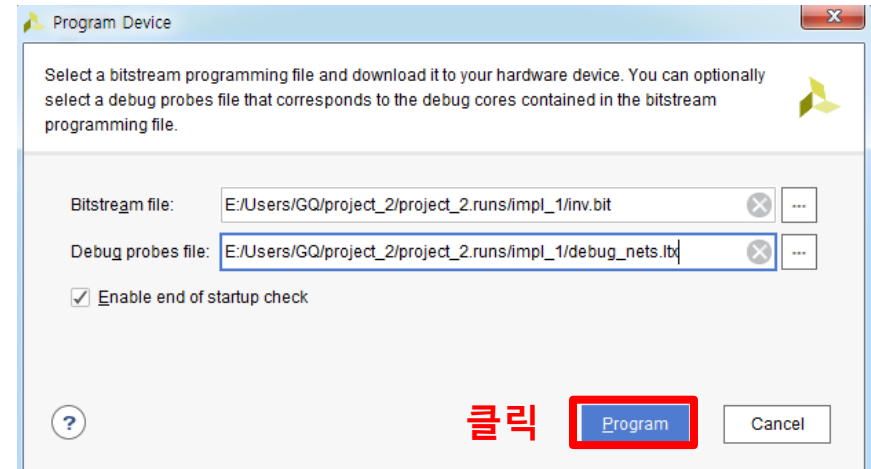
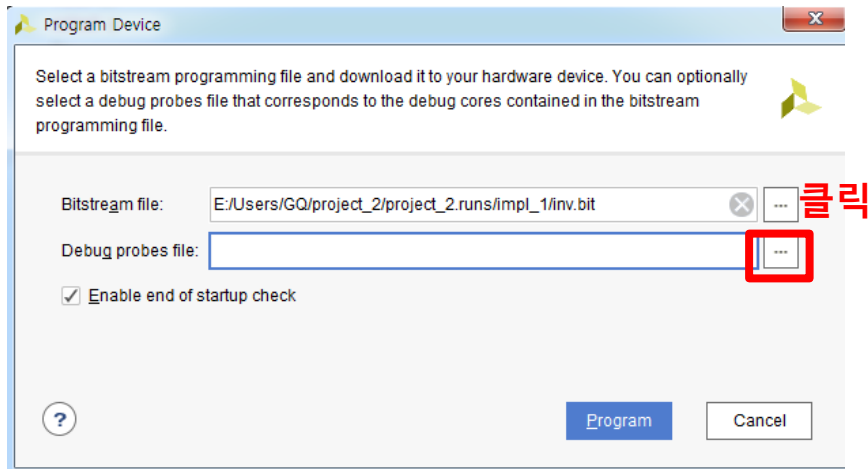
At the bottom, the **Tcl Console** shows the following commands and output:

```

INFO: [Labtoolstcl 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digilent/210251A54EE0
set_property PROGRAM.FILE {E:/Users/GQ/project_2/project_2.runs/impl_1/inv.bit} [get_hw_devices xc7a75t_0]
current_hw_device [get_hw_devices xc7a75t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a75t_0] 0]
INFO: [Labtools 27-1435] Device xc7a75t (JTAG device index = 0) is not programmed (DONE status = 0).
  
```

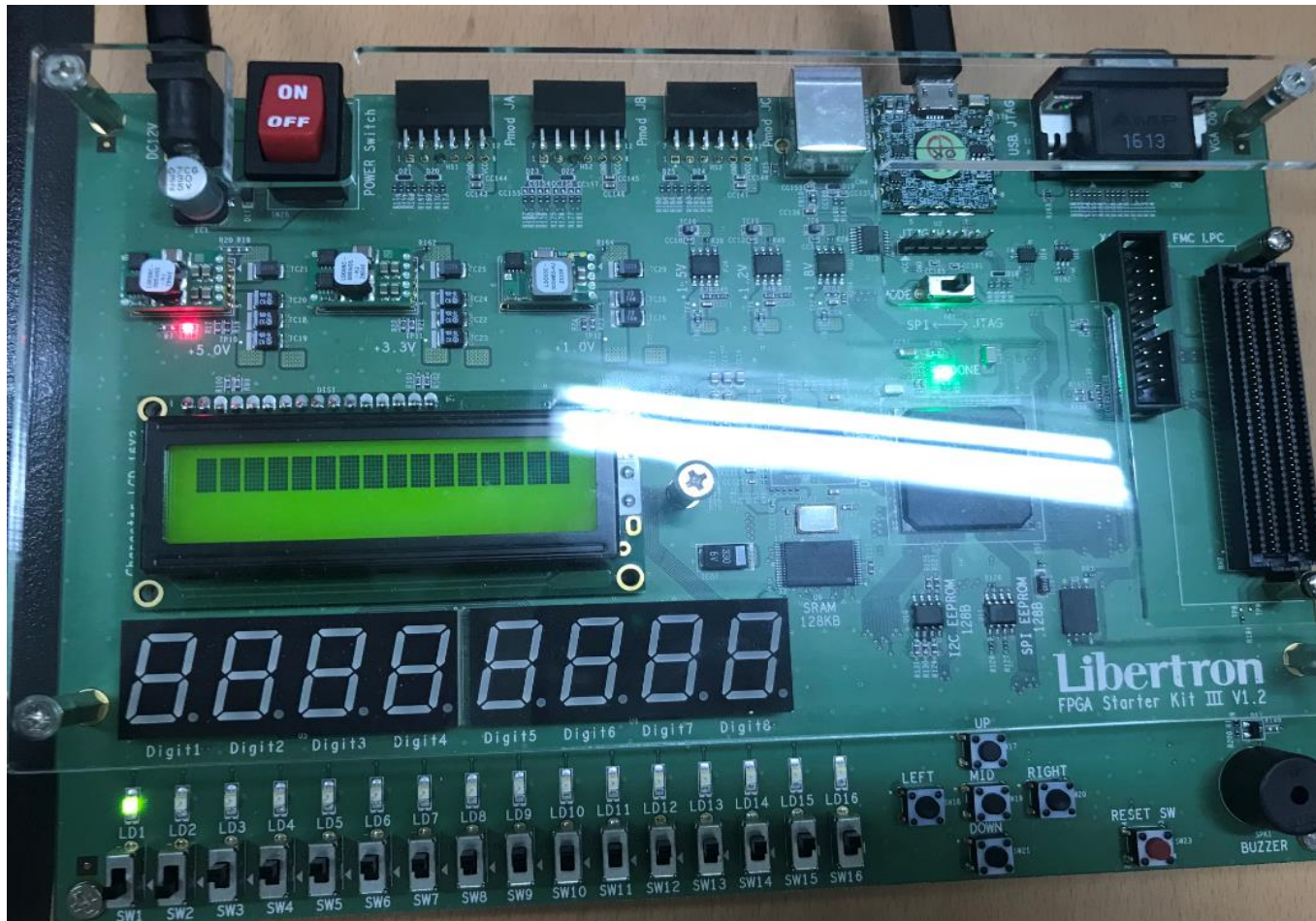
## ◆ Example(NOT gate)

### ● Program Device



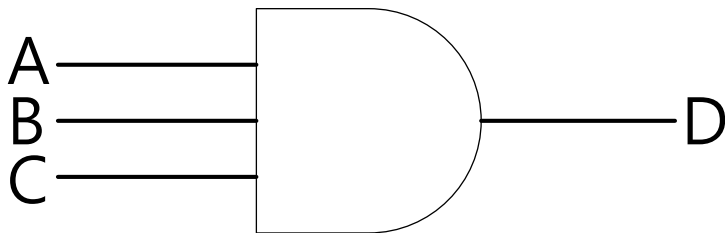
## ◆ Example(NOT gate)

- FPGA 동작 결과

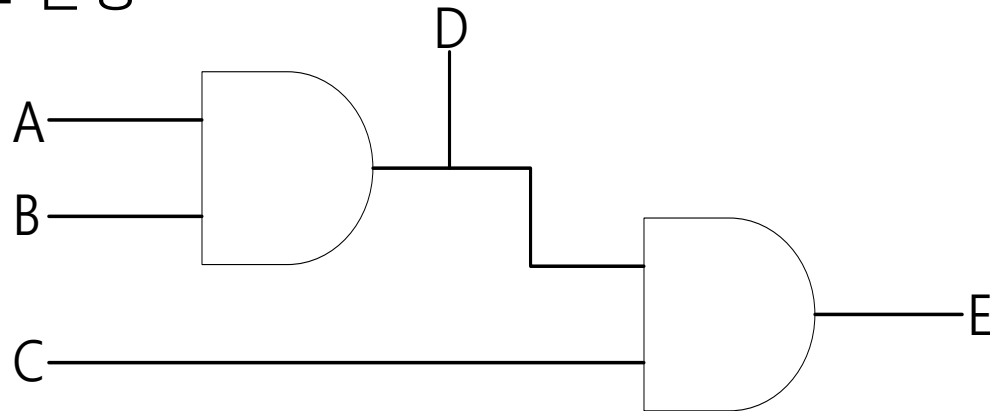


## ◆ 3-input AND gate

- (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)

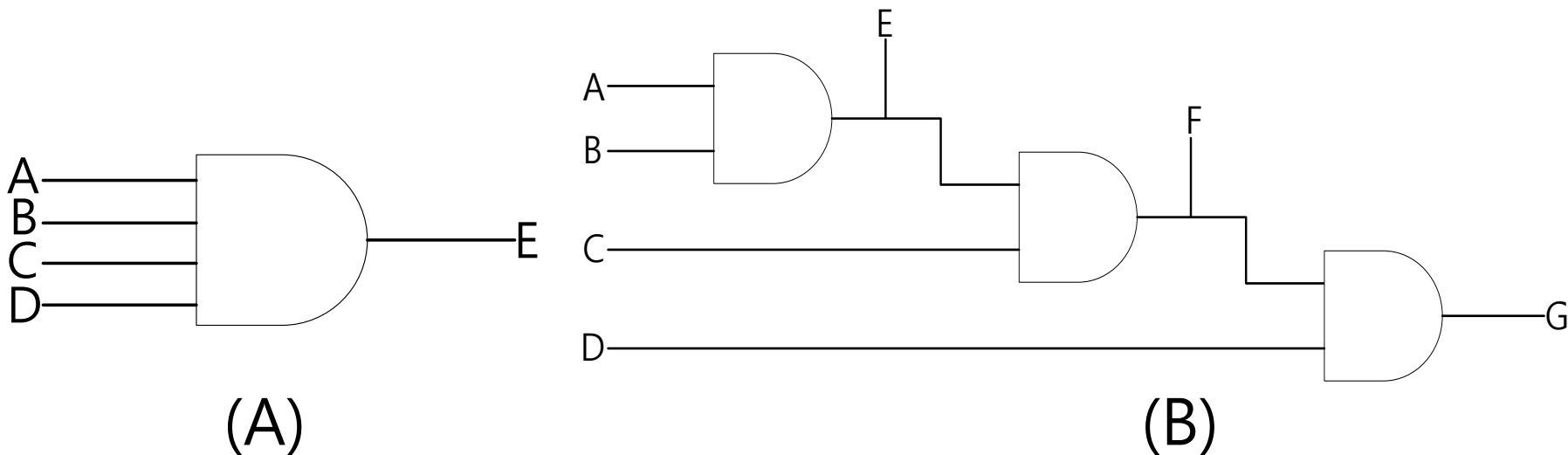
◆ 3-input AND gate 진리표

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		



## ◆ 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

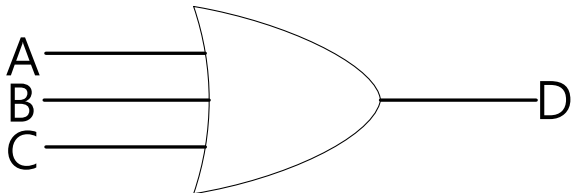


## ◆ 4-input AND gate 진리표

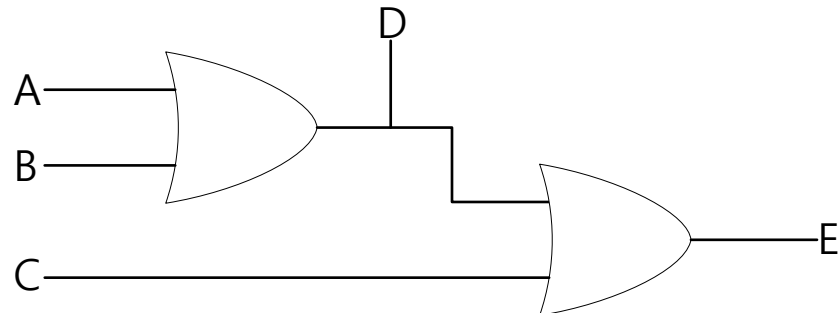
In A	In B	In C	In D	Out E	Out F	Out G
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

## ◆ 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)



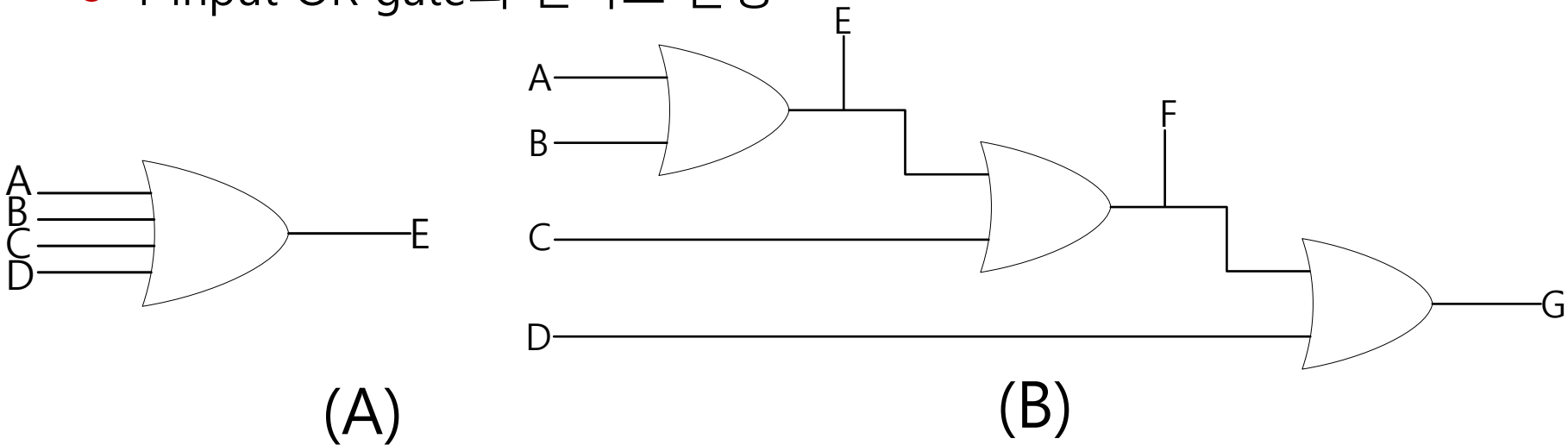
(B)

◆ 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		

## ◆ 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



## ◆ 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			