

Lab 2. 시뮬레이션 및 FPGA 프로그래밍

2017 Fall Logic Design Lab

Department of Computer Science and Engineering

Seoul National University

Outline

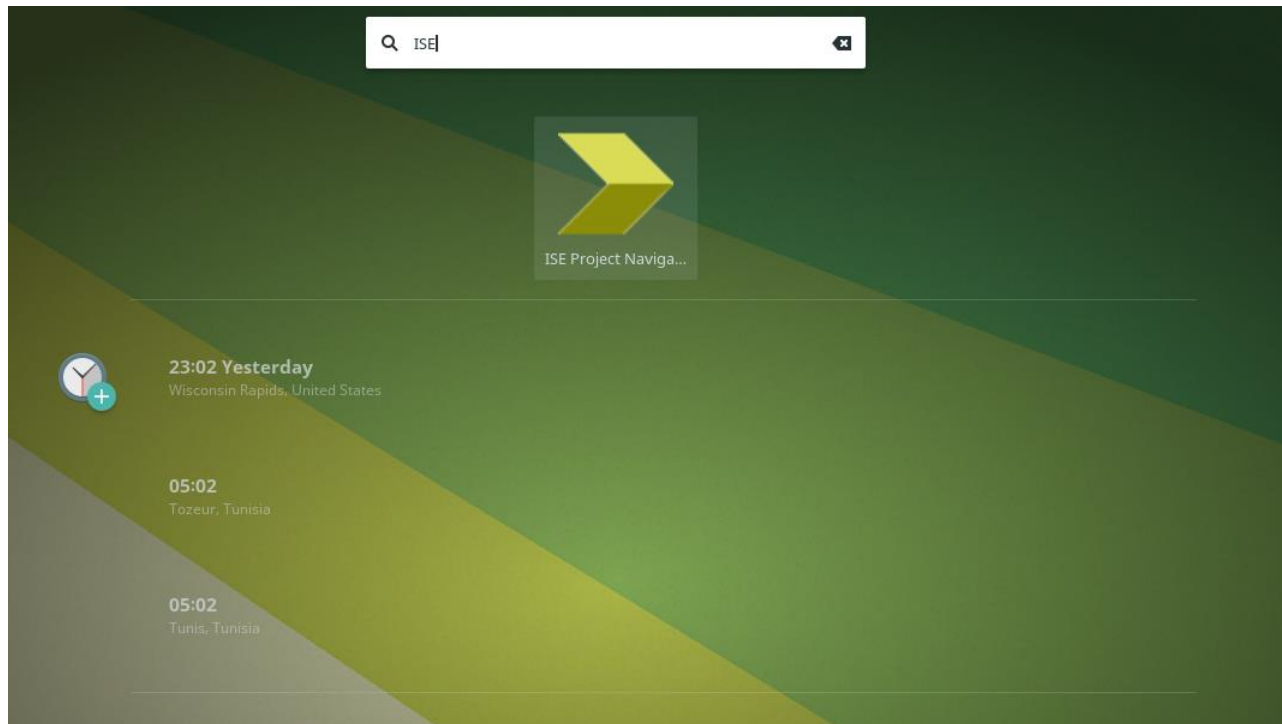
1. FPGA 및 Xilinx ISE 소개
2. Xilinx ISE를 이용한 회로 설계 및 시뮬레이션
3. SNU Logic Design 보드 실습
4. XOR 게이트 설계 및 시뮬레이션
5. 복잡한 논리식 설계 및 시뮬레이션

FPGA 및 Xilinx ISE 소개

- FPGA (Field Programmable Gate Array)
 - 내부 논리 기능 및 연결을 직접 프로그램이 가능한 반도체 소자로 전자 회로 개발 시 설계 및 검증과정에서 주로 사용
 - Spartan-3AN TQG144 : Xilinx 사의 FPGA 모델로, 순차 로직이나 조합 로직을 구현할 수 있는 CLB (Configurable Logic Block)들로 구성됨
- Xilinx ISE
 - Xilinx에서 만든 하드웨어설계, 합성 및 분석 용 소프트웨어
 - 회로도 및 HDL (Hardware Description Language) 등을 이용하여 시스템을 설계하고 시뮬레이션, 실제 디바이스 합성 등 하드웨어 설계 과정에 필요한 전반적인 기능 지원

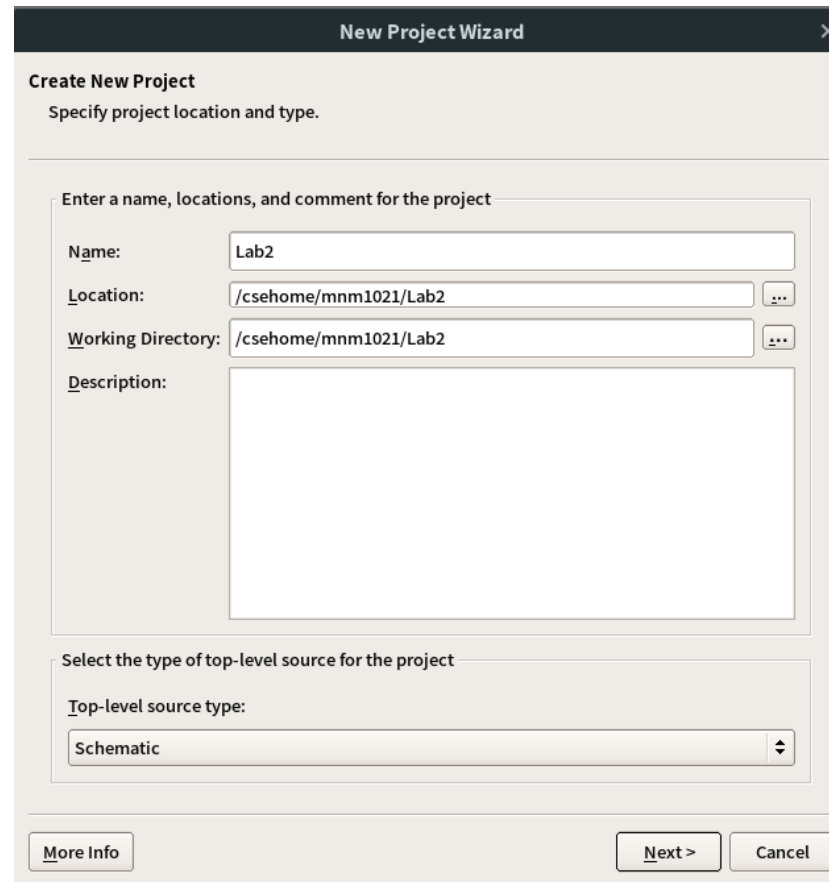
Xilinx ISE 를 이용한 회로도(Schematic) 설계 (1)

- Xilinx ISE 실행 후 File - New Project 클릭



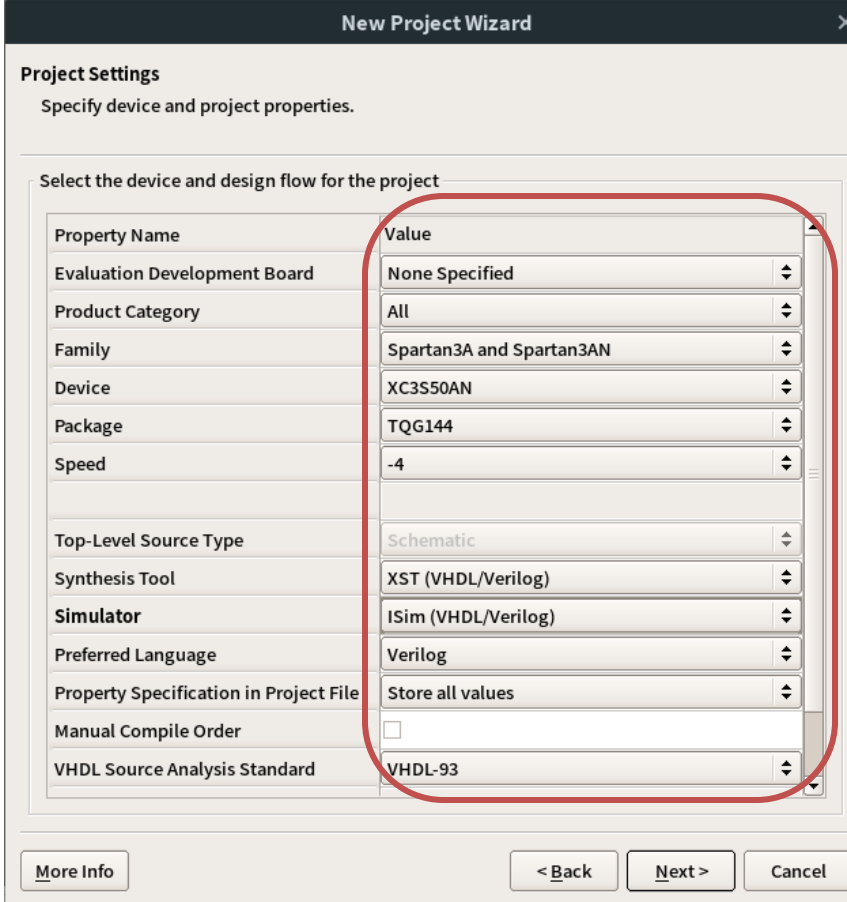
Xilinx ISE 를 이용한 회로도(Schematic) 설계 (2)

- Xilinx ISE 실행 후 File - New Project 클릭



프로젝트 설정

- 프로젝트 셋팅을 다음과 같이 설정



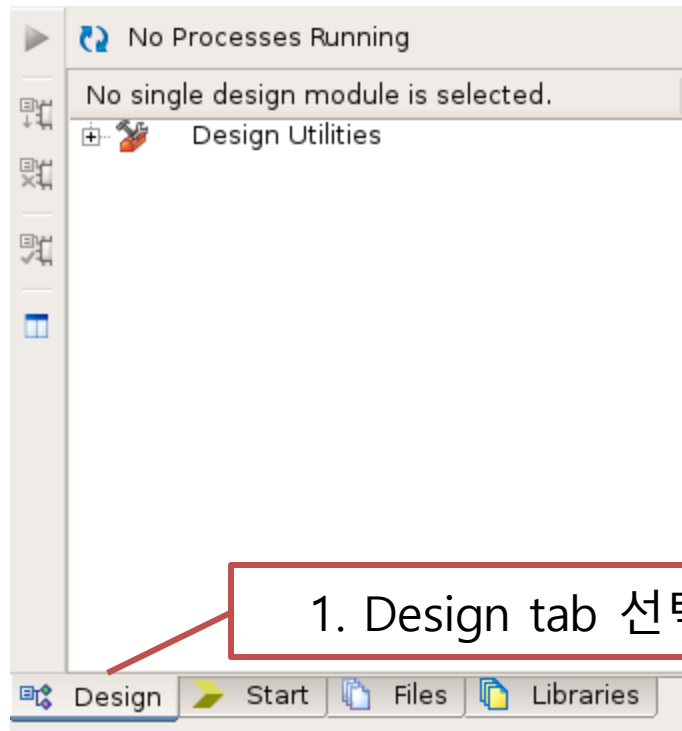
The image shows the 'New Project Wizard' dialog box, specifically the 'Project Settings' step. The title bar says 'New Project Wizard' with a close button. The main heading is 'Project Settings' with the instruction 'Specify device and project properties.' Below this is a section 'Select the device and design flow for the project'. A red oval highlights the 'Value' column of the settings table. The settings are as follows:

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3A and Spartan3AN
Device	XC3S50AN
Package	TQG144
Speed	-4
Top-Level Source Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93

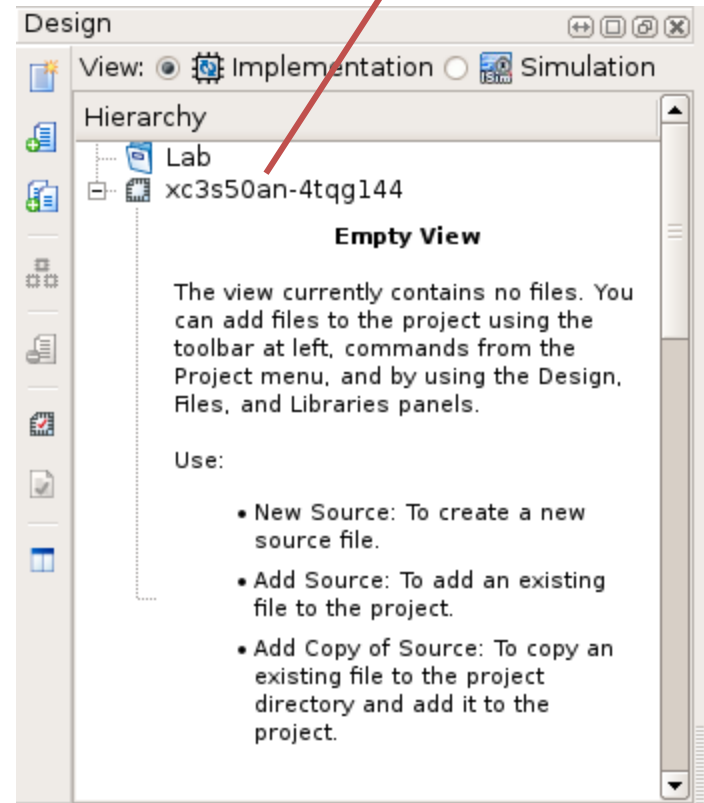
At the bottom, there are four buttons: 'More Info', '< Back', 'Next >', and 'Cancel'.

소스 파일 추가 (1)

2. Target device 마우스 우 클릭 후
"New Source..." 선택

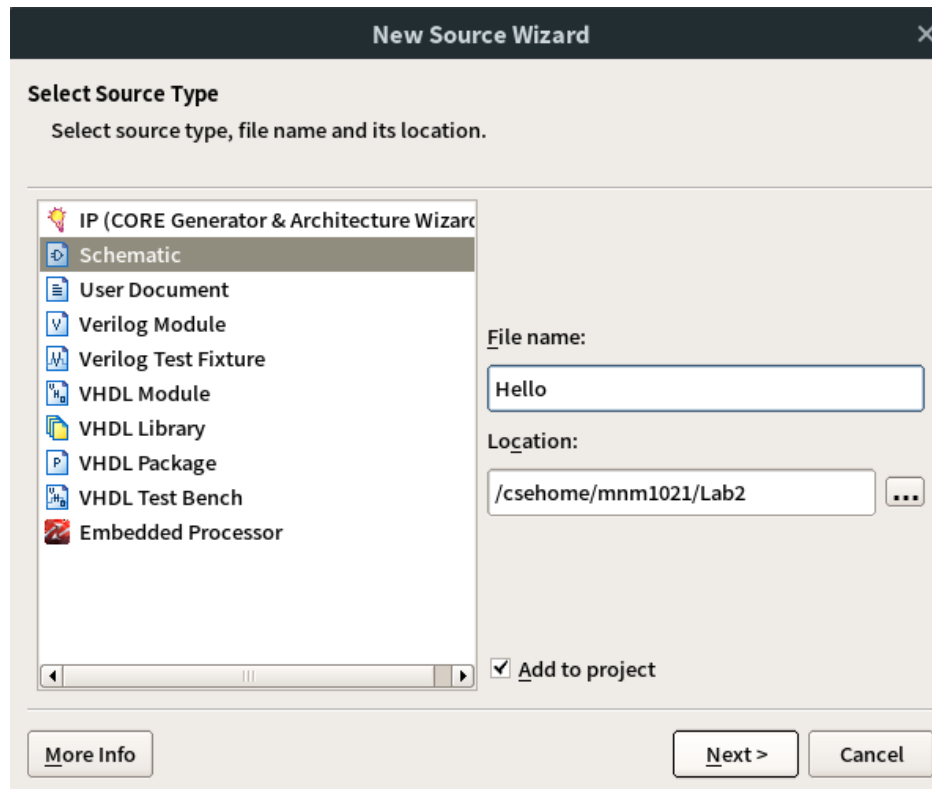


1. Design tab 선택



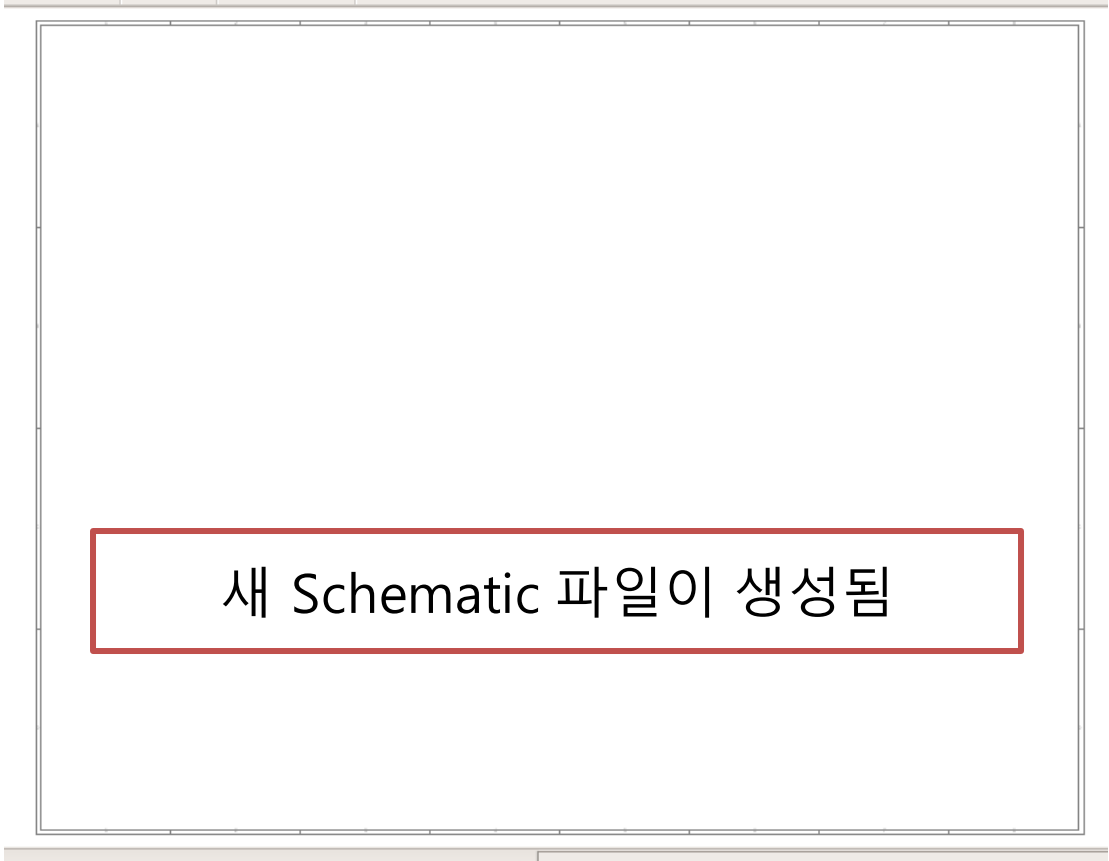
소스 파일 추가 (2)

- Source type 중 Schematic 선택 후 파일명 입력



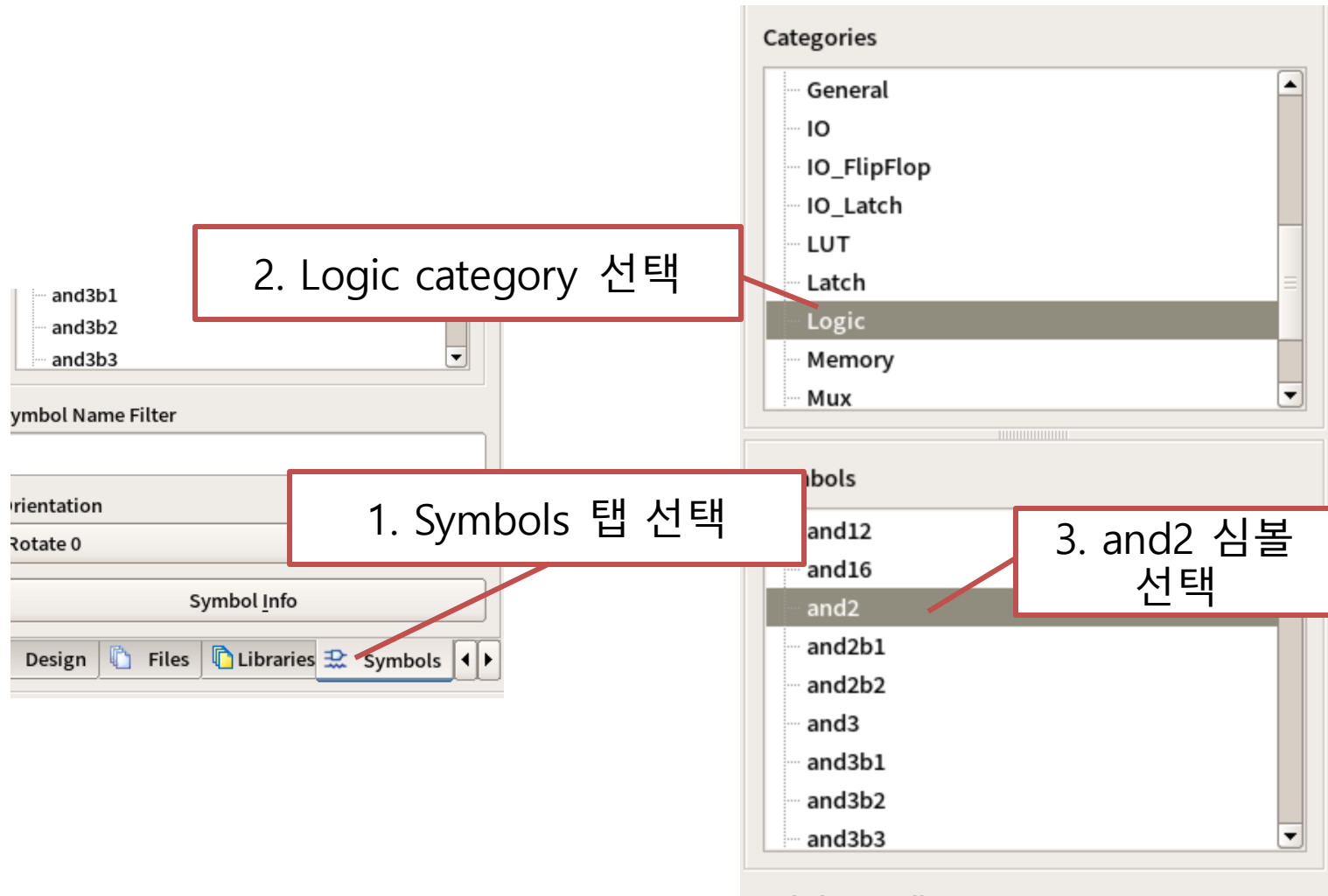
소스 파일 추가 (3)

- Source type 중 Schematic 선택 후 파일명 입력



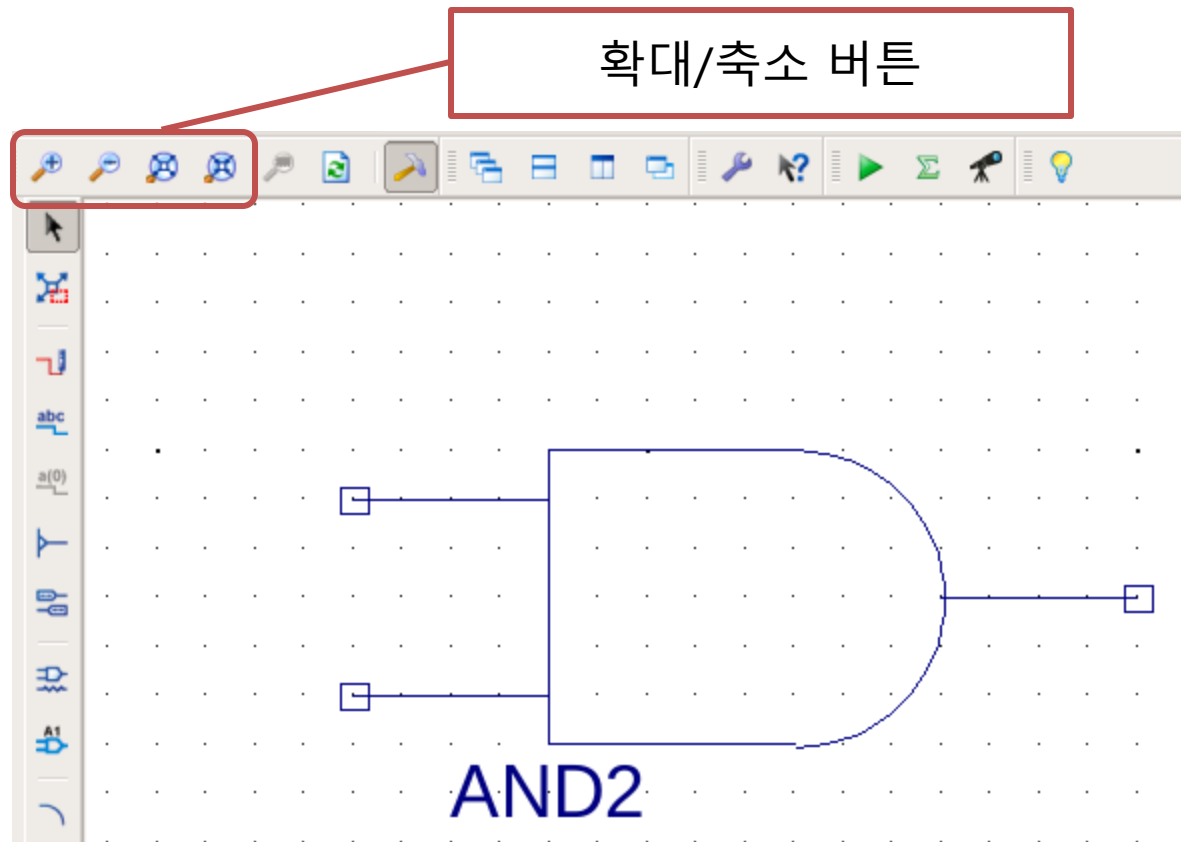
새 Schematic 파일이 생성됨

Schematic에 새로운 심볼 추가 (1)



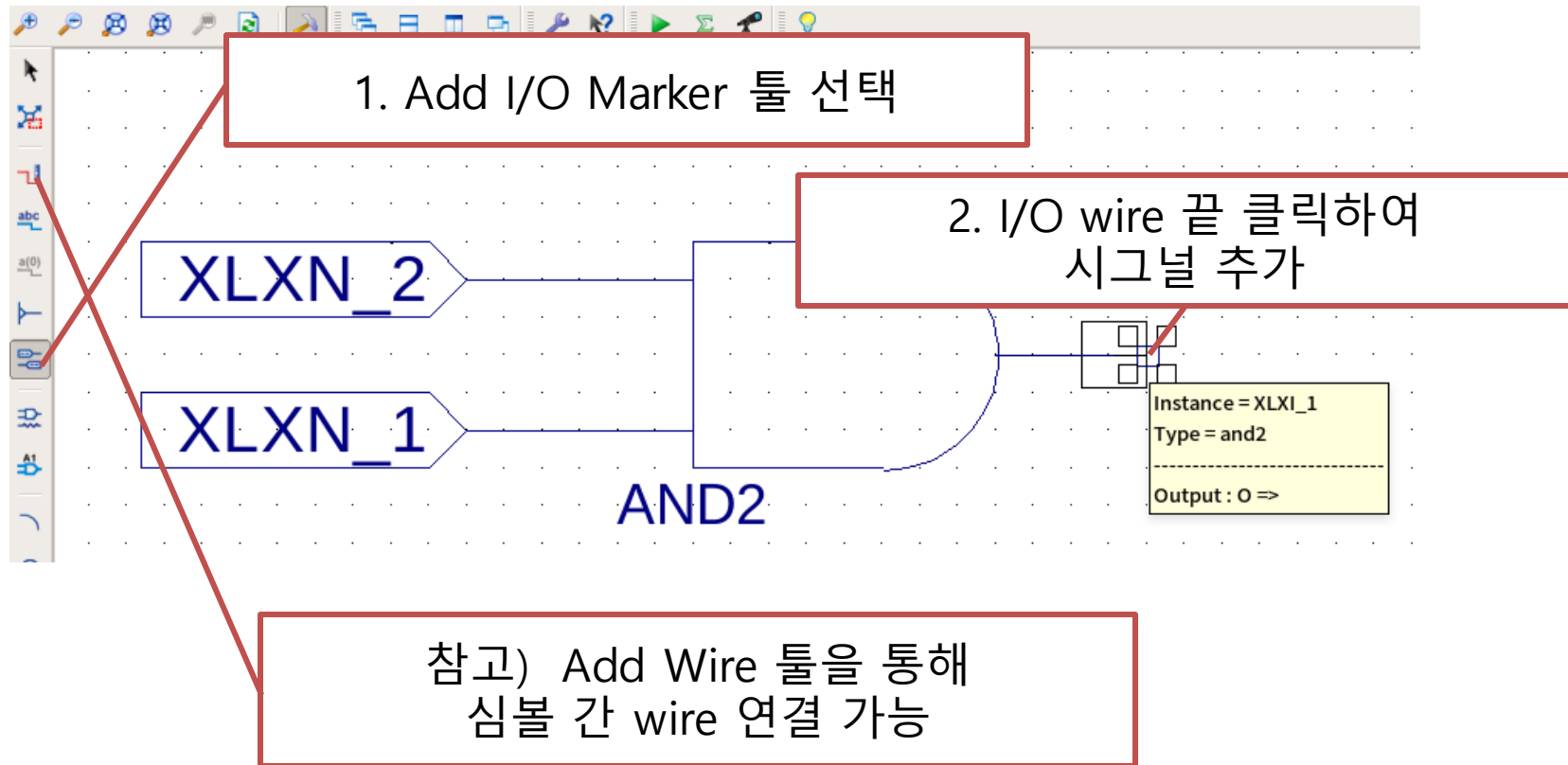
Schematic에 새로운 심볼 추가 (2)

- And2 게이트 추가

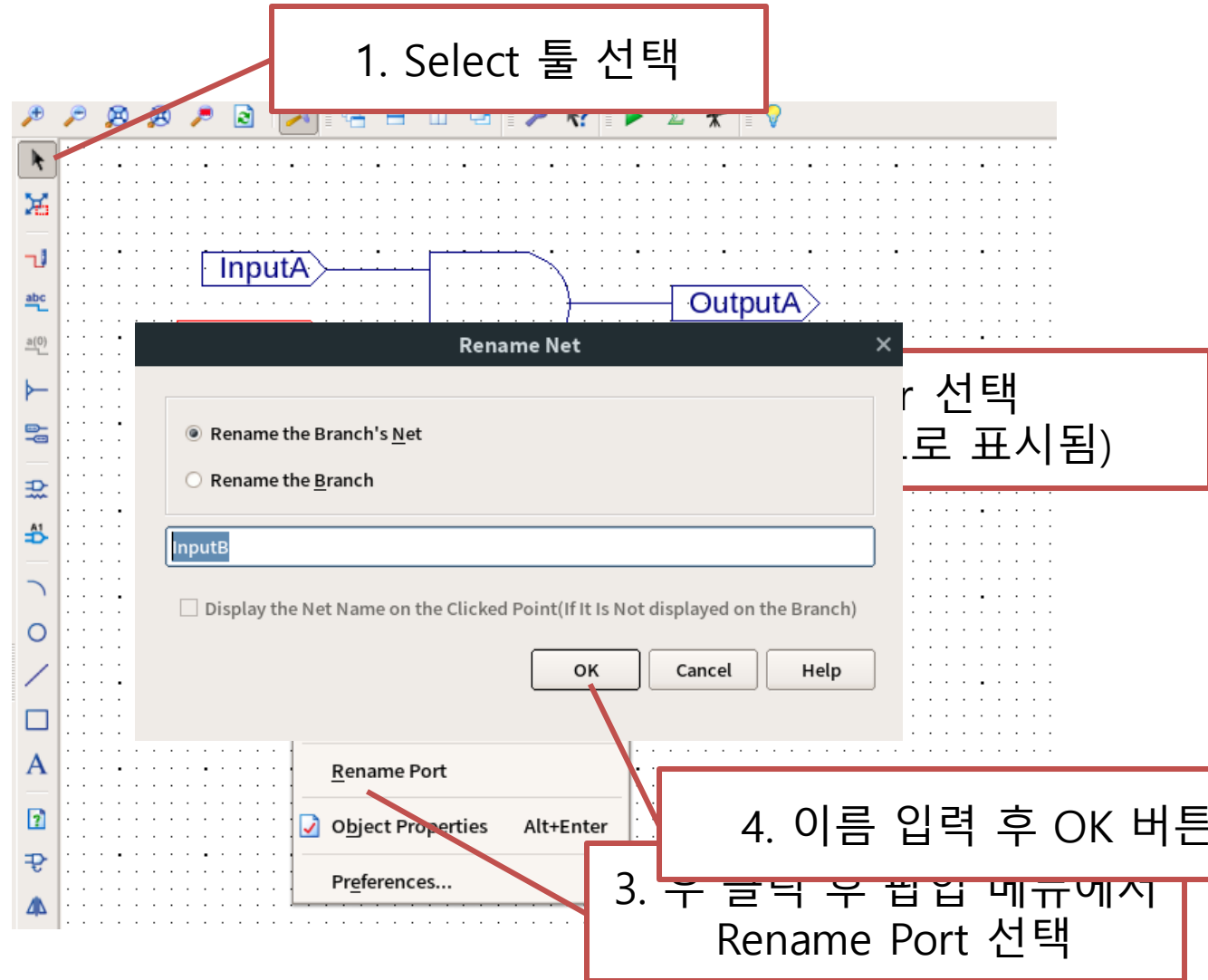


I/O 시그널 추가

- 게이트 Pin 모두 I/O Marker 추가

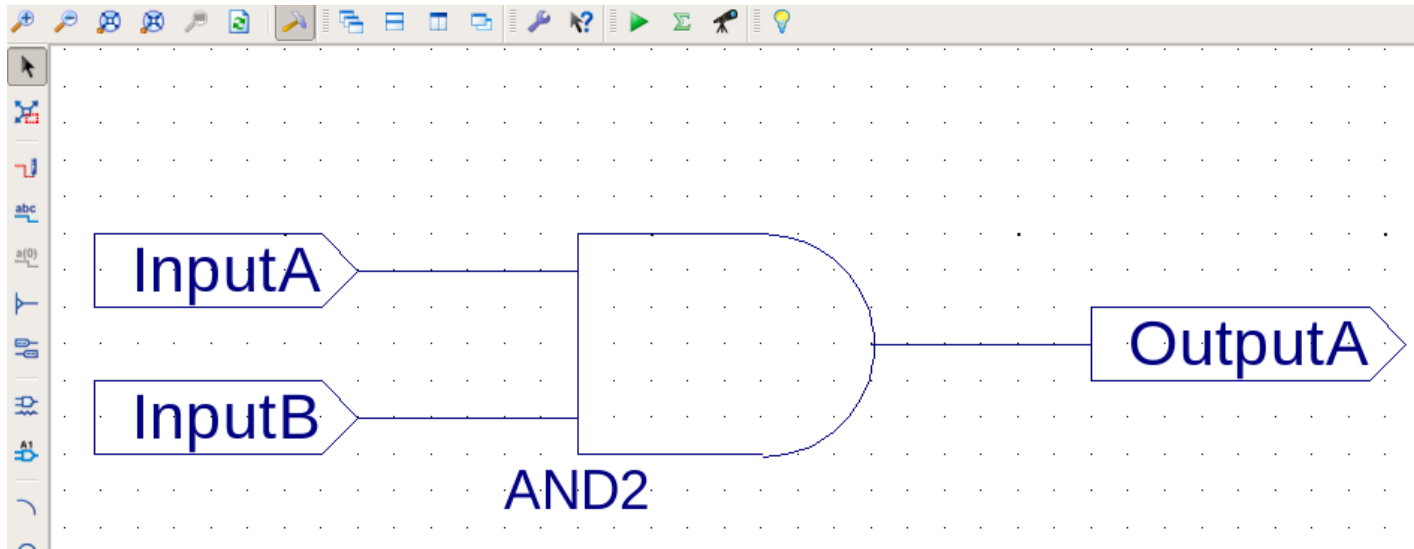


I/O 시그널 이름 설정 (1)



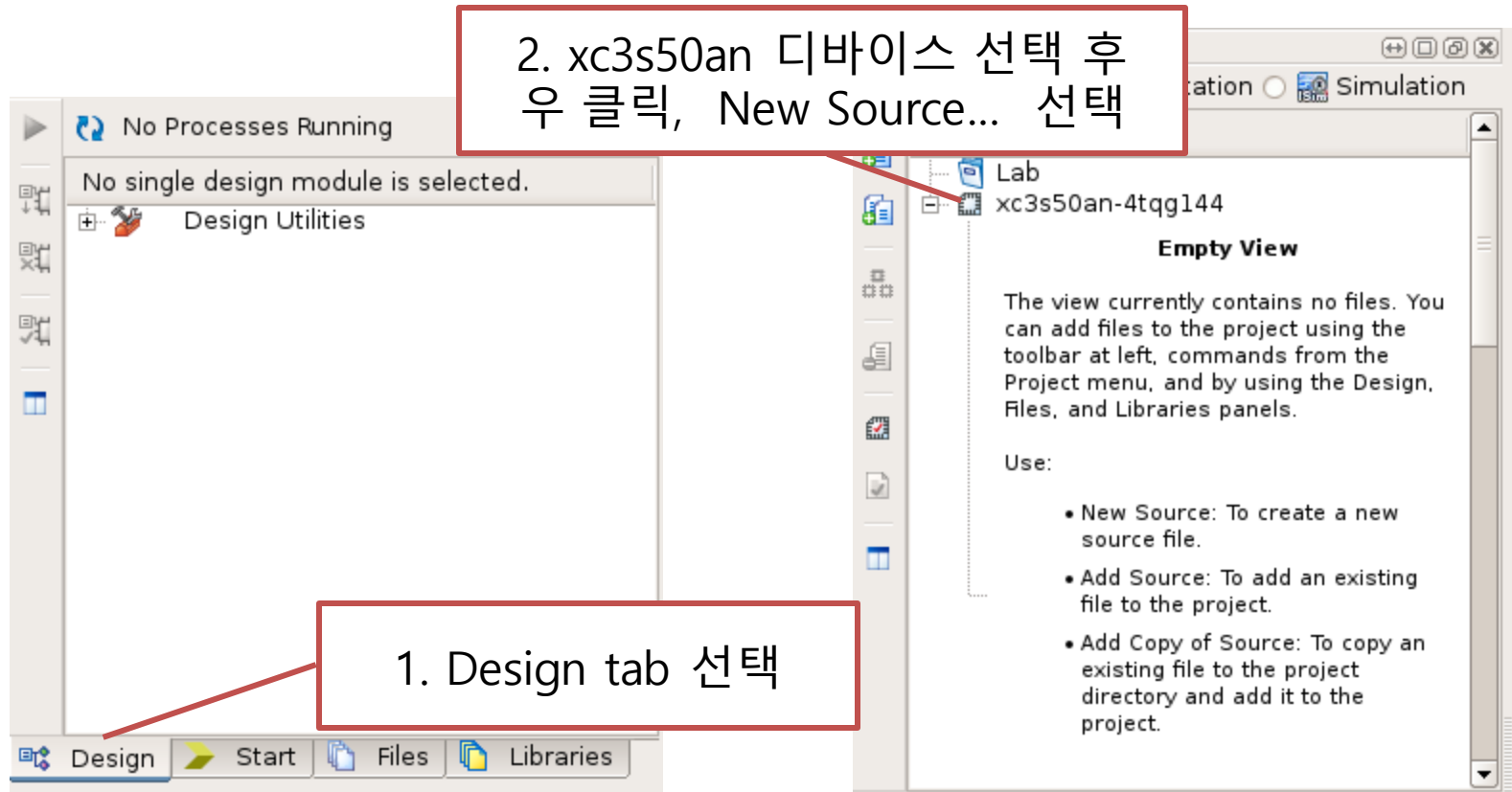
I/O 시그널 이름 설정 (2)

- 그림과 같이 InputA, InputB, OutputA로 이름 설정



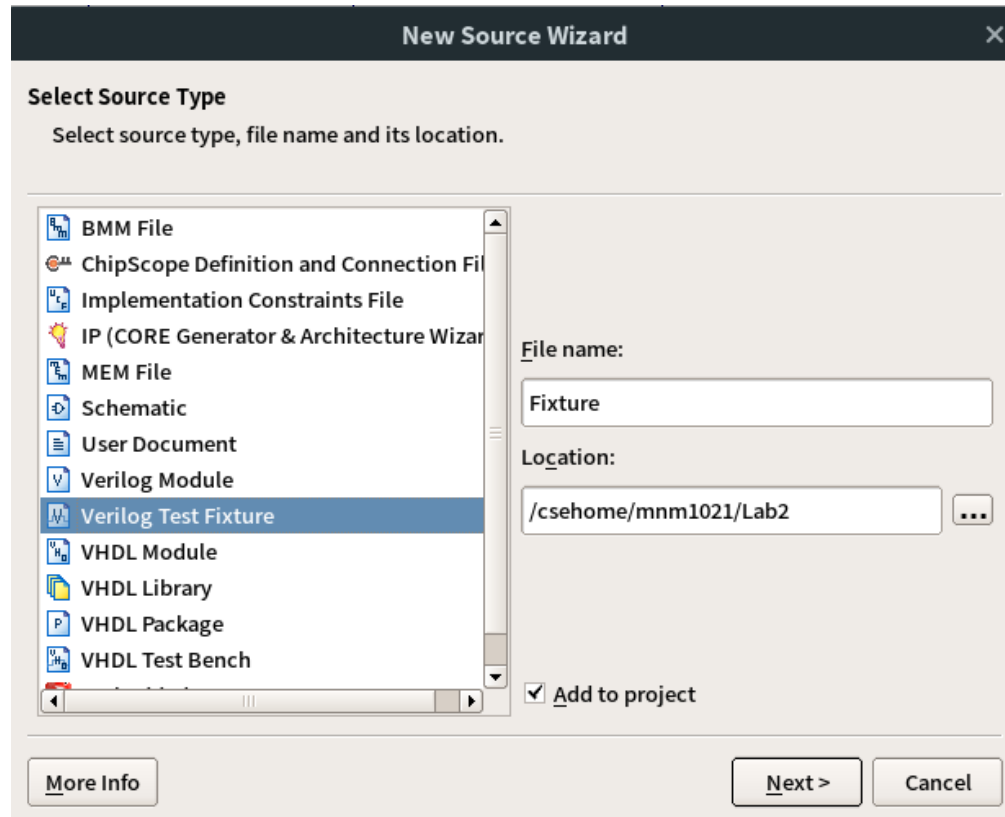
Xilinx ISE를 통한 시뮬레이션

- Test fixture는 HDL (Hardware Description Language)로 작성되는 파일로, UUT (Unit Under Test) 컴포넌트의 동작을 기술함



Test Fixture 파일 추가 (1)

- Verilog Test Fixture 선택 후 파일 이름 입력



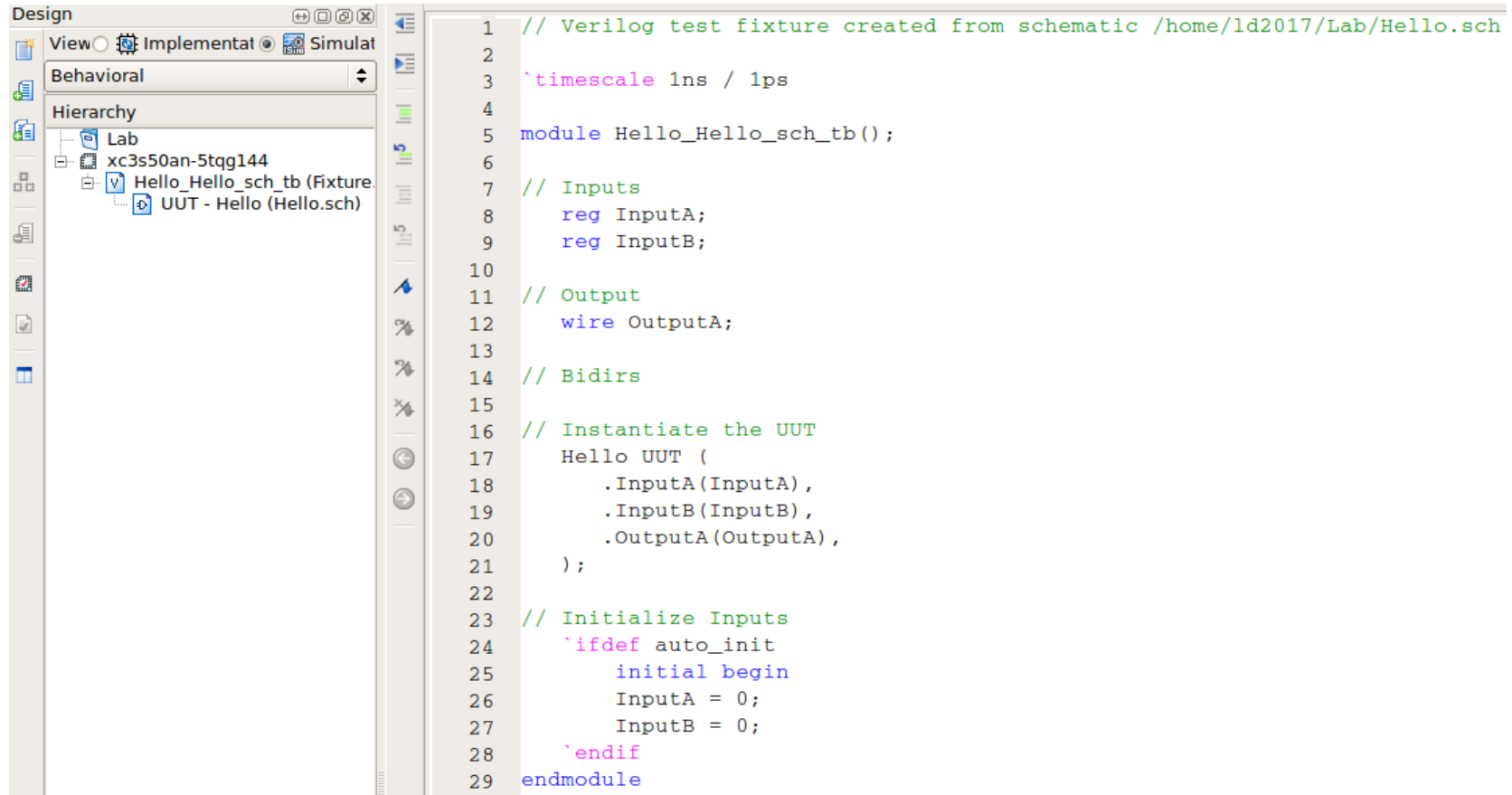
Test Fixture 파일 추가 (2)

- Test fixture과 연결할 schematic 파일 선택



Test Fixture 파일 추가 (3)

- Test fixture 파일이 생성되고 기본소스가 작성된 Verilog 창이 열림



Test Fixture 코드 작성

```
1 // Verilog test fixture created
2
3 `timescale 1ns / 1ps
4
5 module Hello_Hello_sch_tb();
6
7 // Inputs
8   reg InputA;
9   reg InputB;
10
11 // Output
12   wire OutputA;
13
14 // Bidirs
15
16 // Instantiate the UUT
17   Hello UUT (
18     .InputA(InputA),
19     .InputB(InputB),
20     .OutputA(OutputA)
21   );
22 // Initialize Inputs
23   initial begin
24     InputA = 0;
25     InputB = 0;
26     #100
27     InputA = 1;
28     #100
29     InputA = 0;
30     InputB = 1;
31     #100
32     InputA = 1;
33     InputB = 1;
34   end
35 endmodule
```

Timescale 정의(e.g., #1 = 1ns)

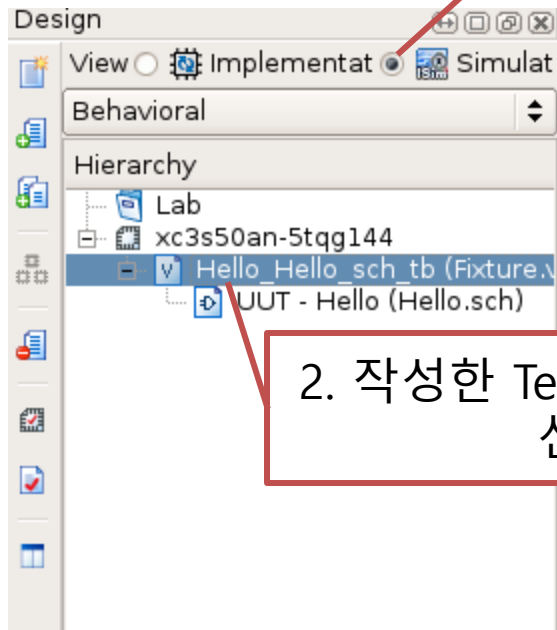
UUT 및 I/O 시그널 파라미터 정의

두 입력 모두 0으로 초기화

100 ns 대기

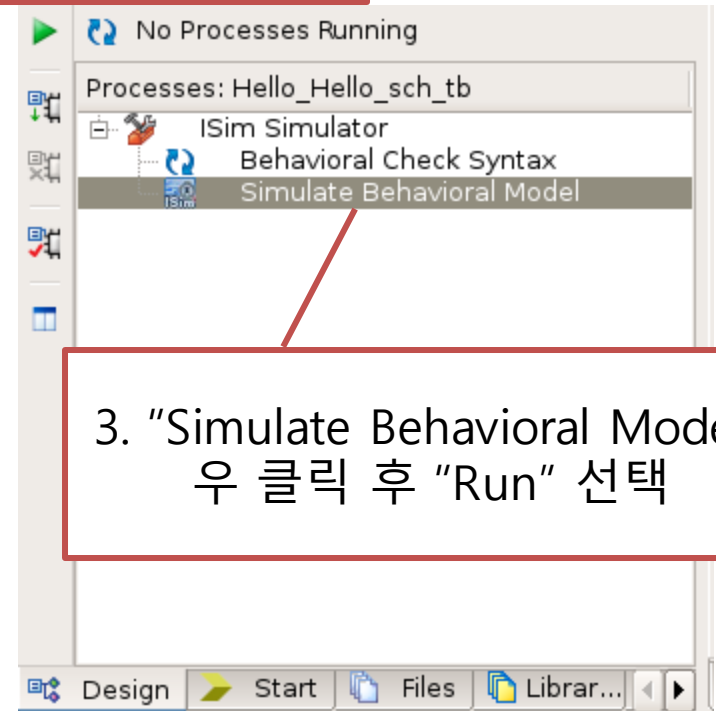
시뮬레이션을 통한 기능 검증

1. Simulation 라디오 버튼 선택

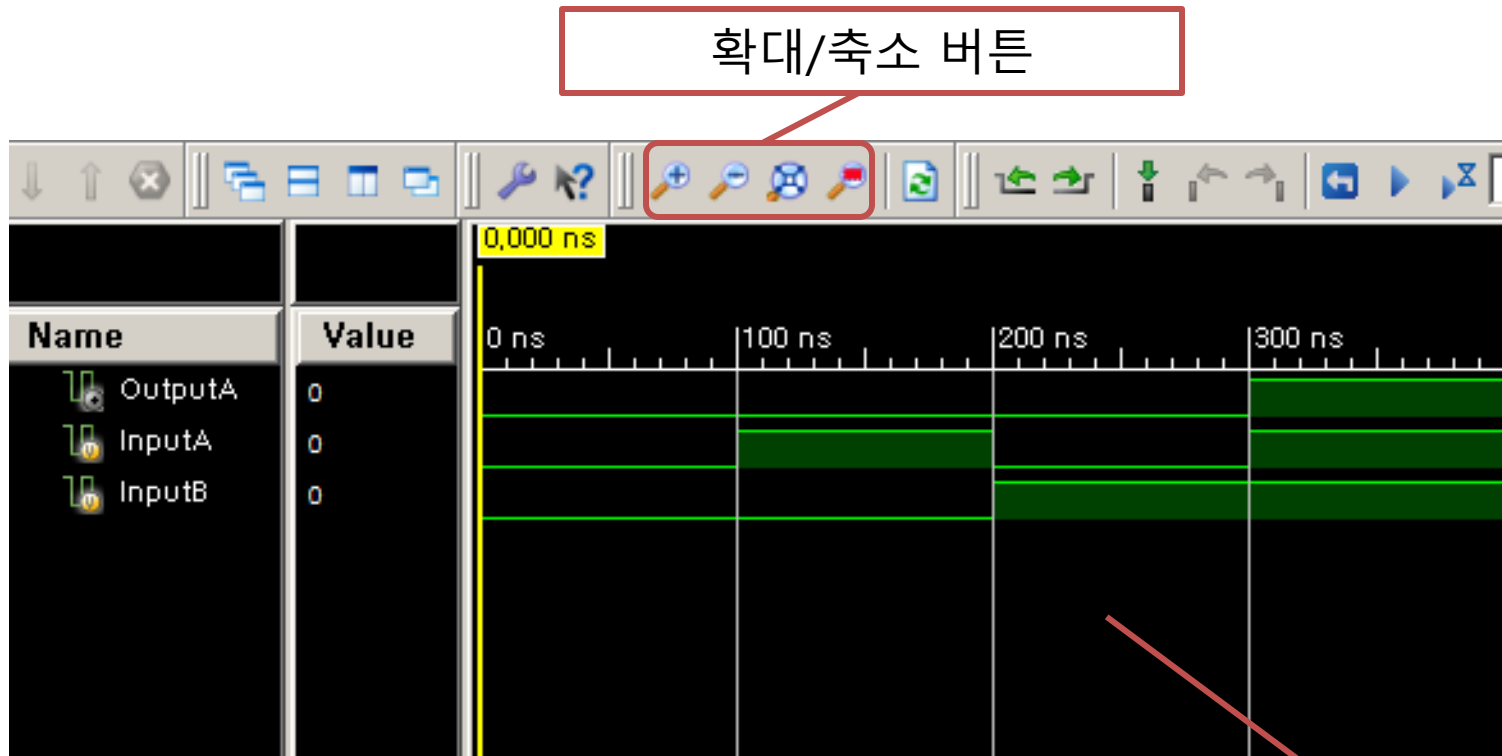


2. 작성한 Test Fixture 파일
선택

3. "Simulate Behavioral Model"
우 클릭 후 "Run" 선택



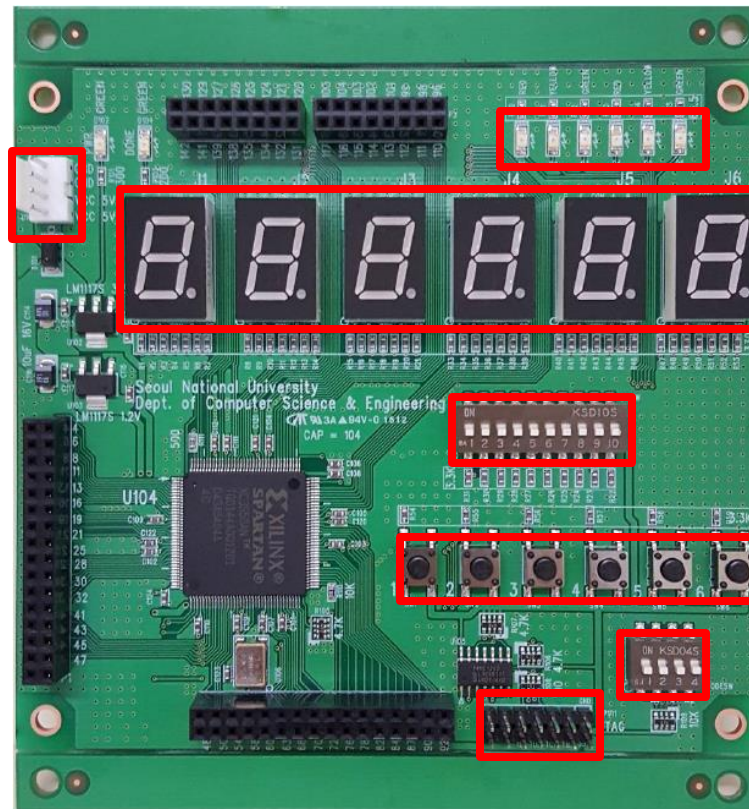
시뮬레이션 결과



SNU Logic Design 보드 실습

- SNU Logic Design Board
 - Spartan-3AN TQG144 FPGA 탑재

Power connector



6 output LEDs

6 7-segment LEDs

DIP switch

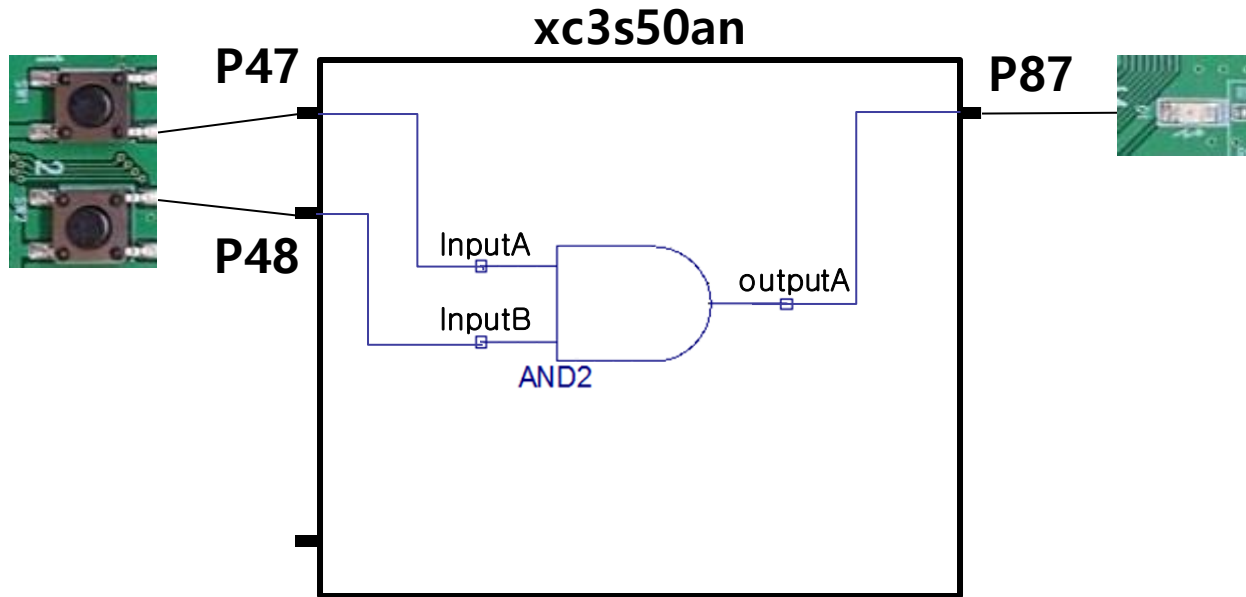
Push button

Mode select switch

JTAG Header

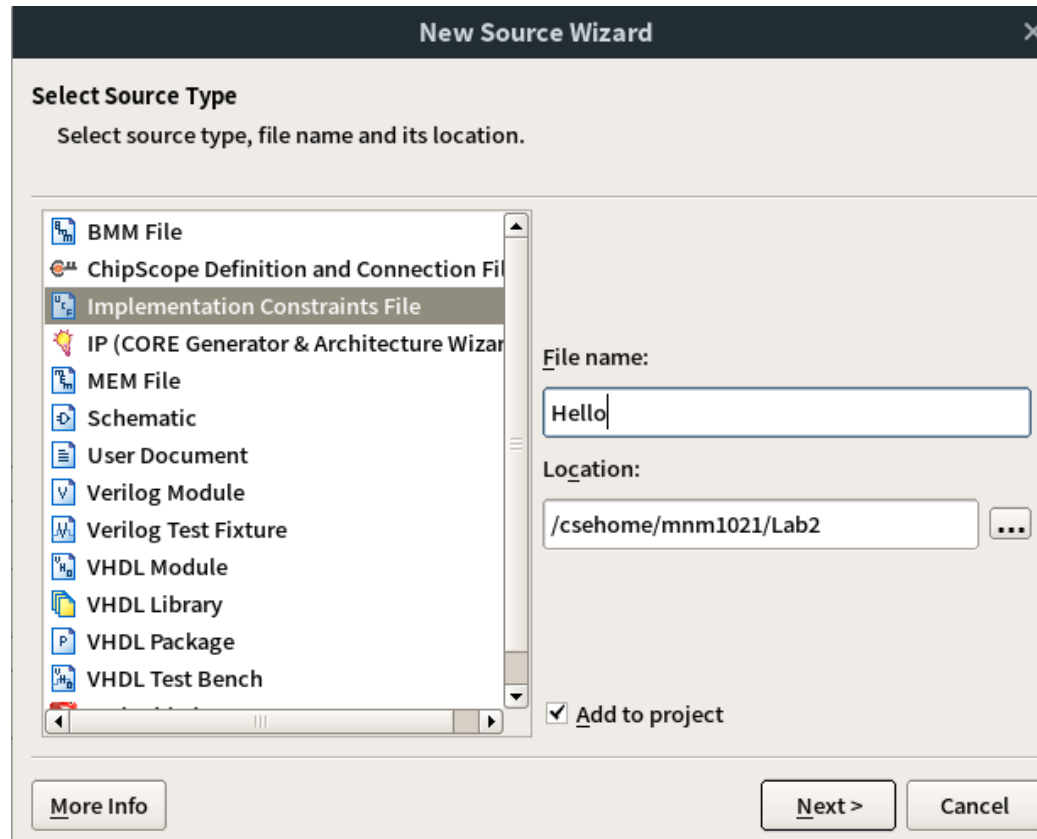
User Constraints File (UCF)

- 사용자 constraints를 기술하기 위한 ASCII text 파일
- 모듈 포트 신호들이 어떻게 물리적인 핀에 연결되어야 하는지 기술
 - 보드 전체 핀 연결은 SNU Logic Design Board User's Manual 참고



UCF 파일 추가

- Implementation Constraints File 선택 후 파일 이름 입력



UCF 파일 편집

- 선언한 I/O 포트를 Push button과 User LED로 연결

올바른 I/O 포트 이름 확인

핀 번호 확인

```
1 NET "InputA" LOC = P47;  
2 NET "InputB" LOC = P48;  
3 NET "OutputA" LOC = P87;  
4
```

I/O 핀 연결

핀 번호 확인

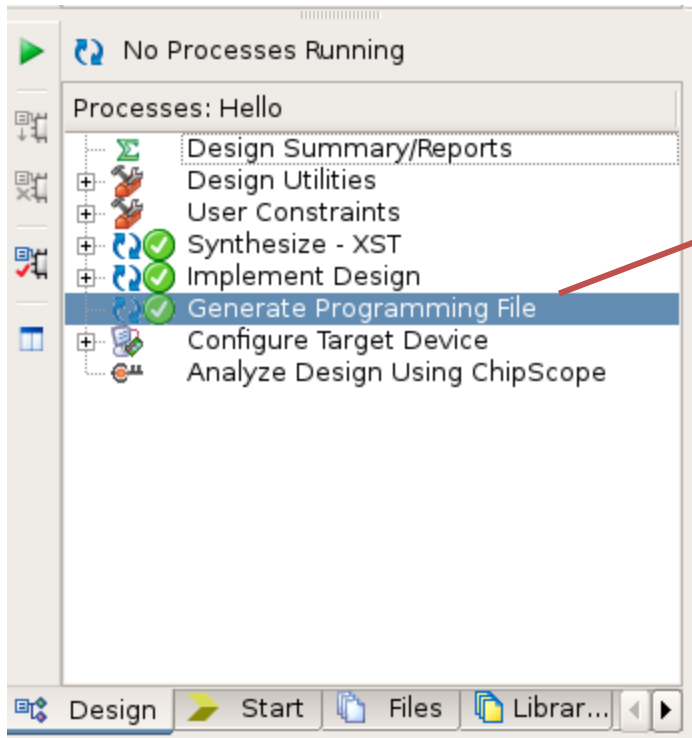
Pin Num	Component
3	A
4	B
5	C
6	D
7	E
8	F
10	G
11	A
12	B
13	C
15	D
16	E
18	F
19	G
20	A
21	B
24	C
25	D
27	E
28	F
29	G
30	1
31	2
32	3
33	4
41	5
42	6
43	7
44	8
45	9
46	10
47	Tactile Switch [SW1]

Pin Num	Component
48	Tactile Switch [SW2]
49	Tactile Switch [SW3]
50	Tactile Switch [SW4]
51	Tactile Switch [SW5]
54	Tactile Switch [SW6]
55	A
58	B
59	C
60	D
62	E
63	F
64	G
68	A
69	B
70	C
71	D
72	E
75	F
76	G
77	A
78	B
79	C
82	D
83	E
84	F
85	G
87	LED [D1] Red
88	LED [D2] Yellow
90	LED [D3] Green
91	LED [D4] Red
92	LED [D5] Yellow
93	LED [D6] Green

SNU Logic Design Board User's Manual

프로그래밍 파일 생성

- Design 탭 – Generate Programming File 실행

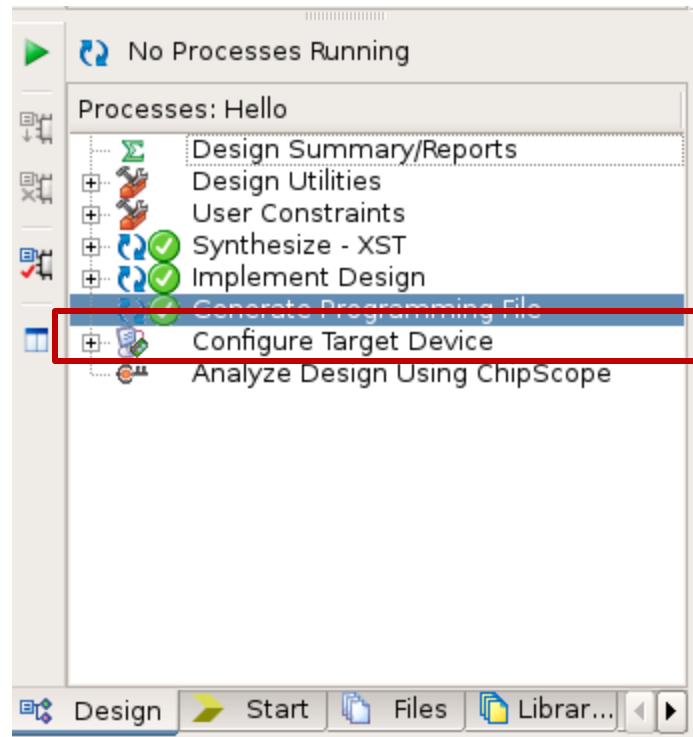


전 단계가 차례로 실행되며 최종적으로 장치에 프로그래밍될 bit파일이 생성됨

- 정상적으로 수행 시 콘솔 창에 Process "Generate Programming File" completed successfully 메시지가 출력됨

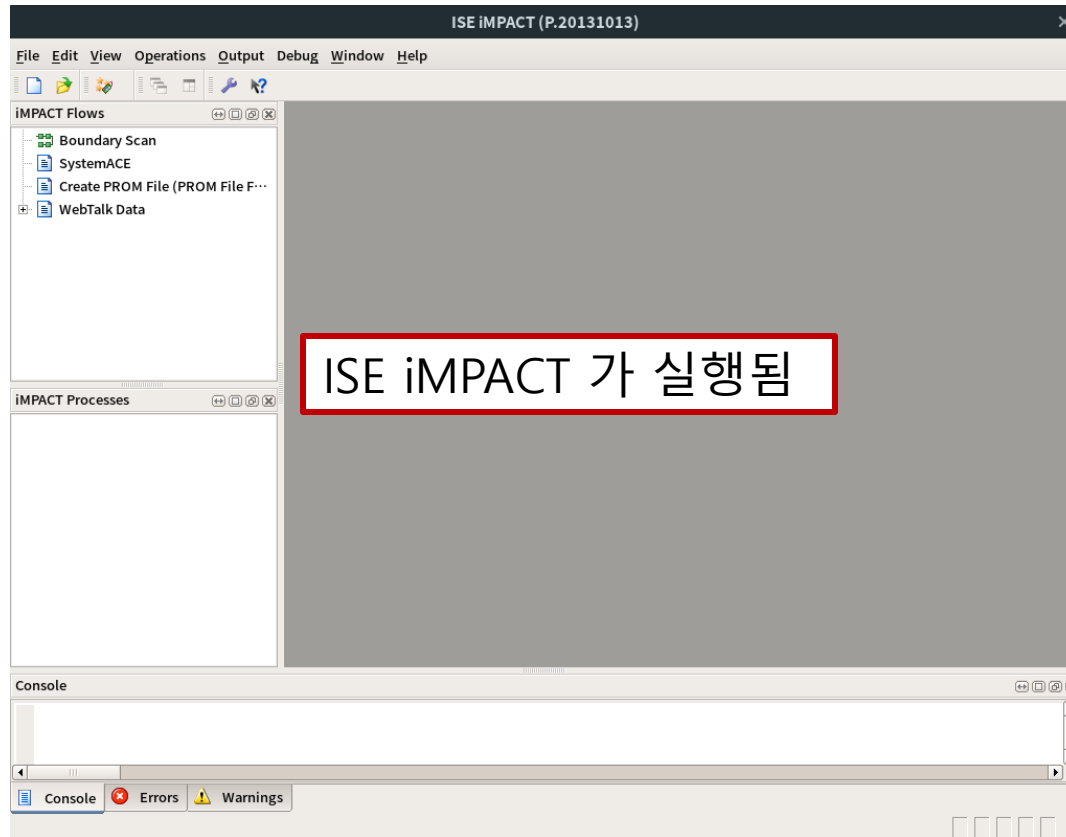
타겟 디바이스 설정 (1)

- Design 탭 – Configure Target Device 실행

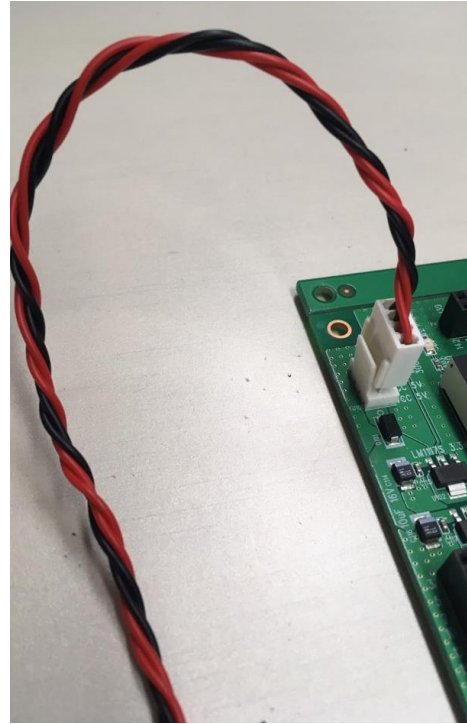


타겟 디바이스 설정 (2)

- Design 탭 – Configure Target Device 실행



전원 케이블 제작



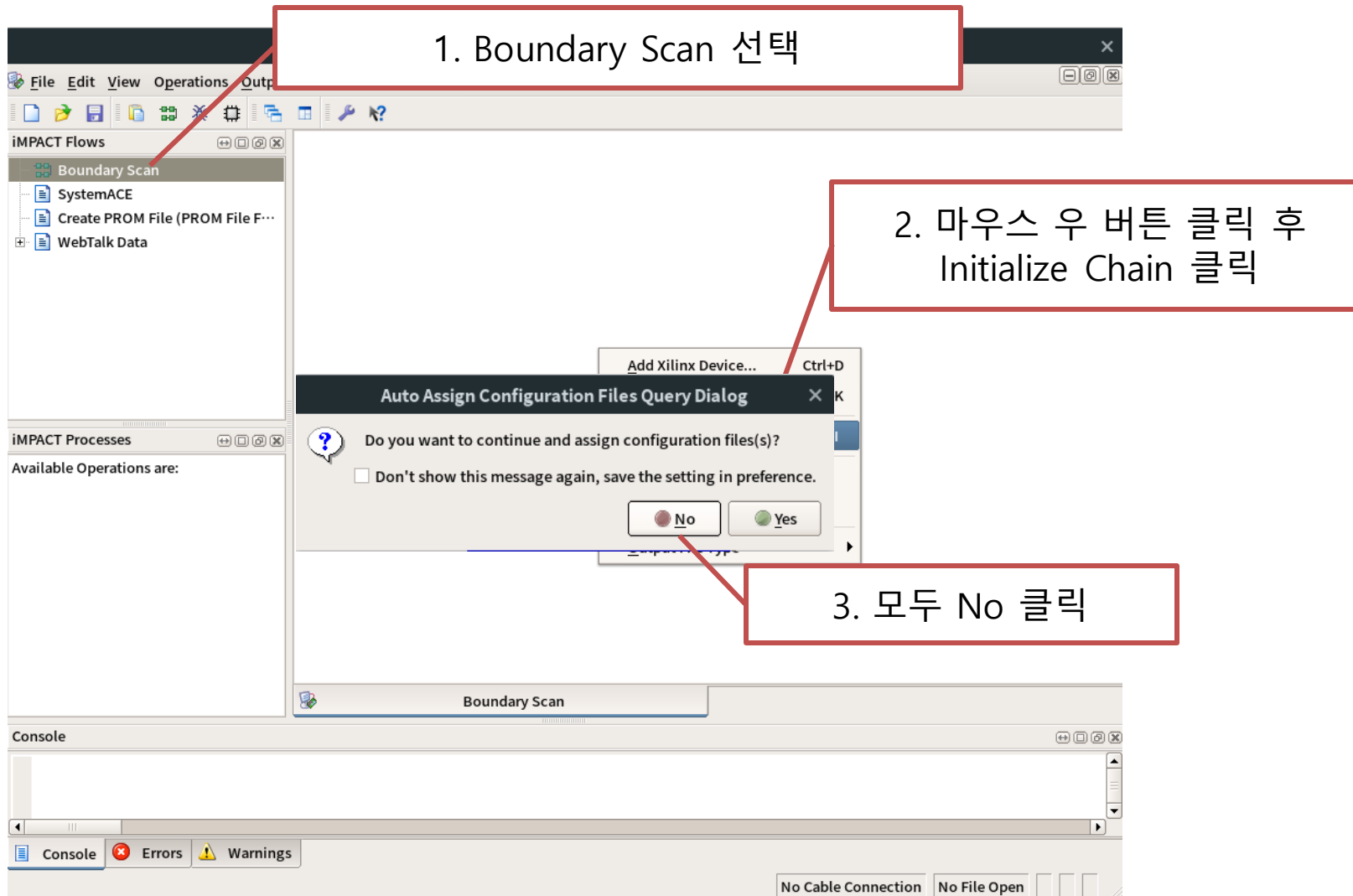
- 선을 같은 색끼리 꼬고, 꼬인 줄끼리 다시 곧 뒤 방향에 맞게 포트에 연결
- 전선 피복 제거

실습 보드와 컴퓨터 JTAG 연결

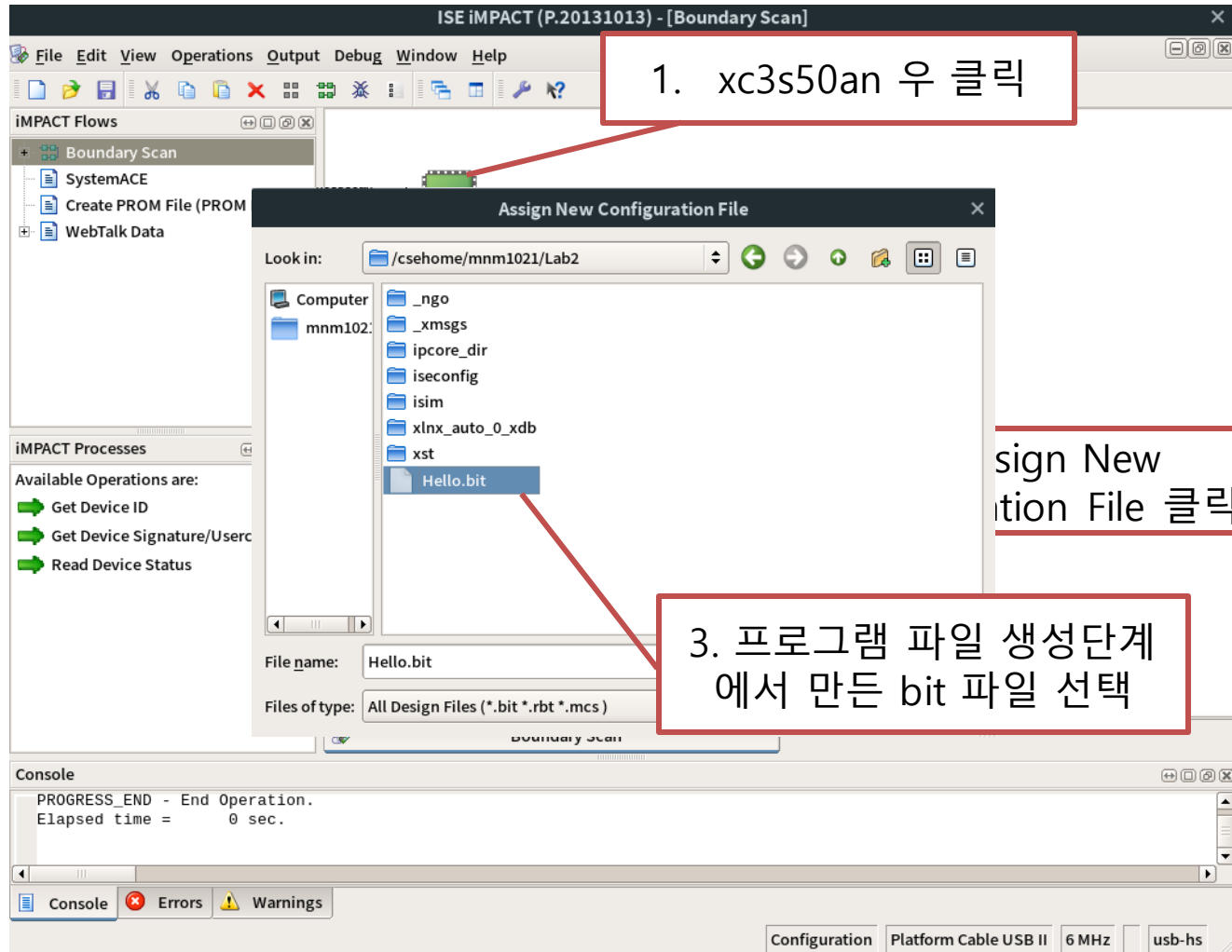


- TDI, TDO, TCK, TMS, VREF, GND 신호를 기판에 표시된 대로 연결

타겟 보드 Boundary Scan

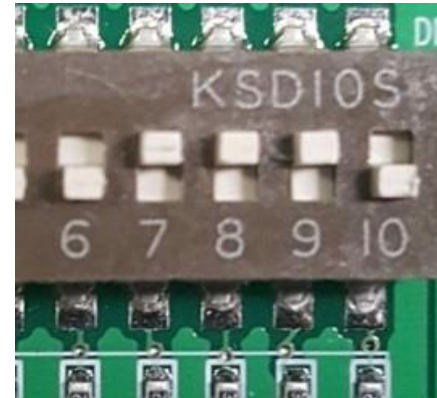
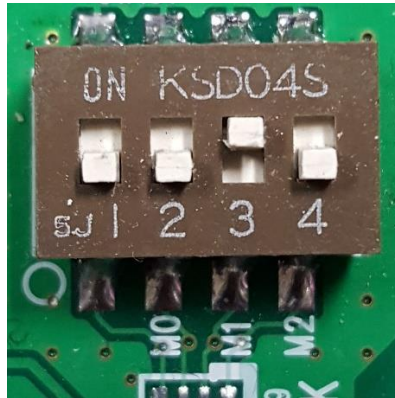


프로그램할 FPGA Bit File 선택



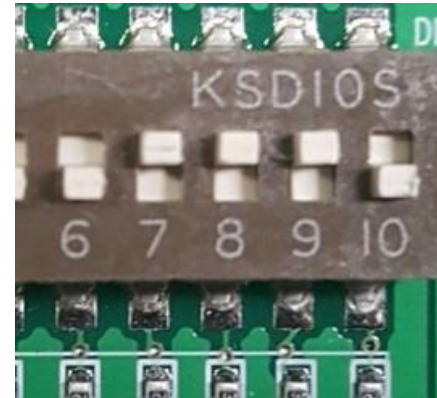
프로그램 모드 설정 (1)

- FPGA에만 프로그램 하는 경우
 - 프로그래밍 속도가 빠름
 - 전원 차단될 시 프로그램한 내용이 삭제됨
- 보드의 딥 스위치를 아래와 같이 설정

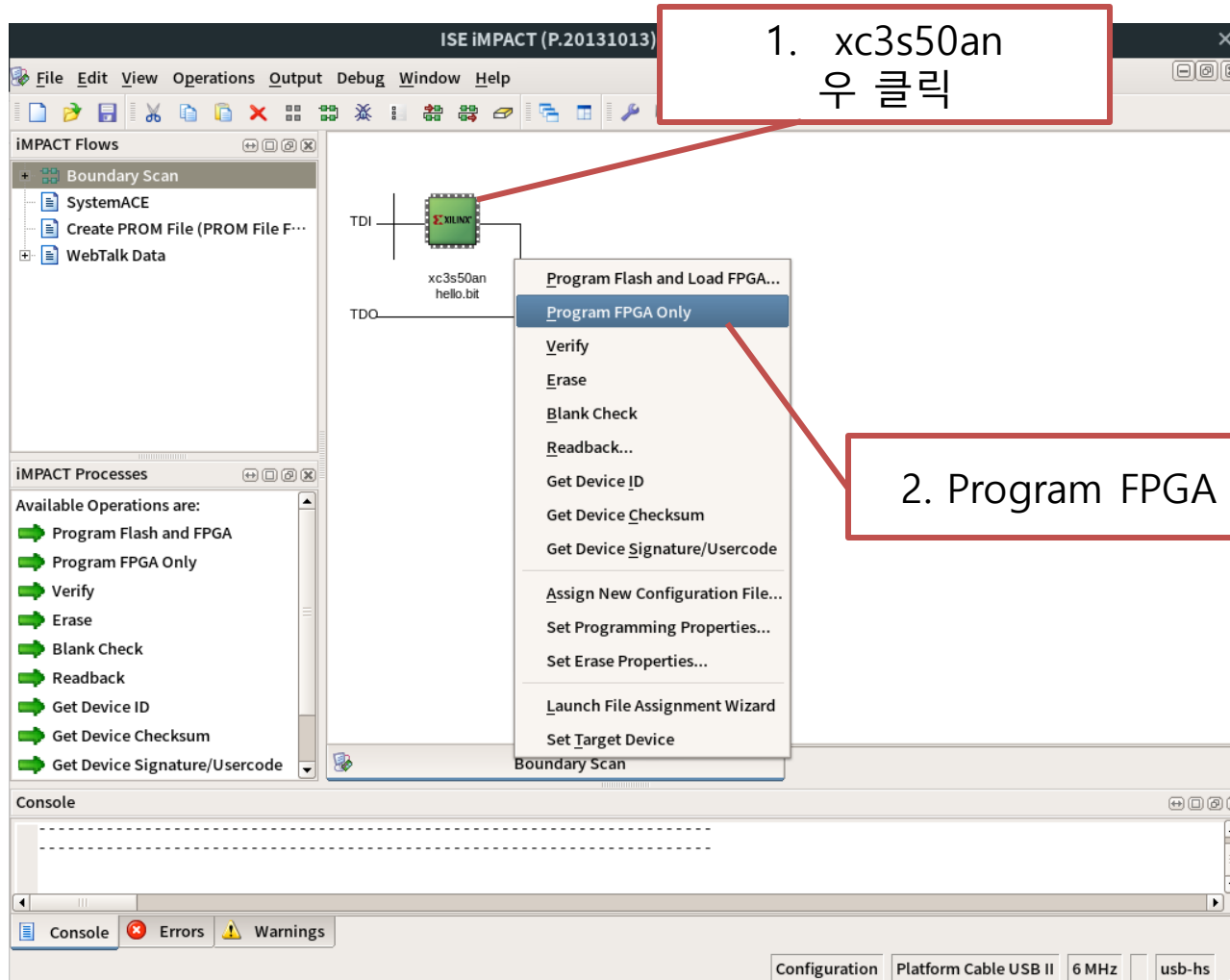


프로그램 모드 설정 (2)

- FPGA 와 Flash에 프로그램 하는 경우
 - 프로그래밍 속도가 느림
 - 전원이 차단돼도 프로그램한 내용이 삭제되지 않음
- 보드의 딥 스위치를 아래와 같이 설정

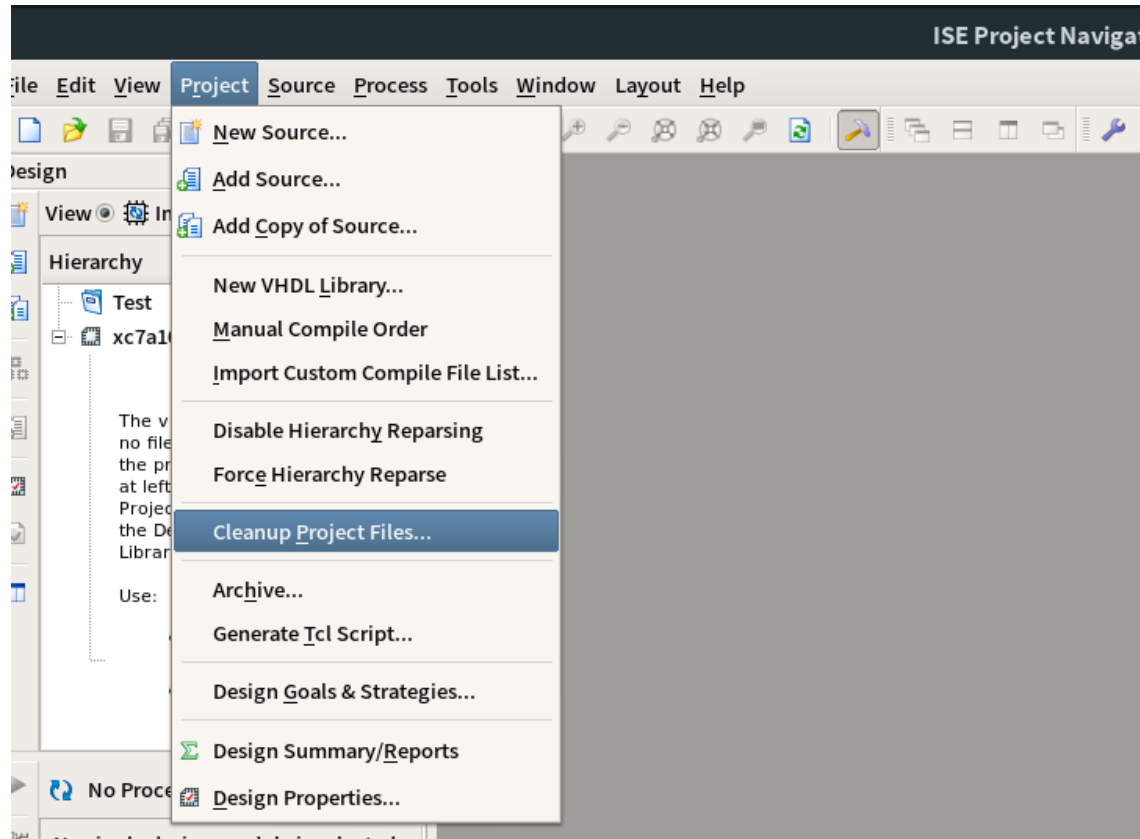


FPGA 프로그램



- Push button과 LED를 통해 결과 확인

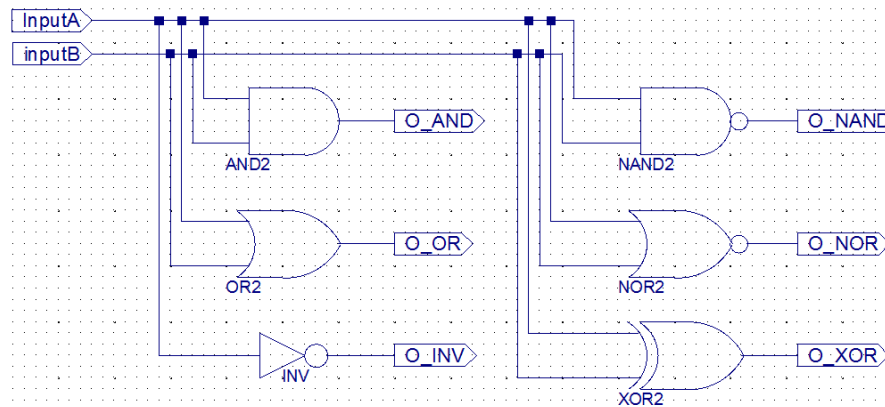
실험 후 유의사항



- 실습 완료 후, 불필요한 파일들을 제거

실험 1 – AND/OR/NOT/NAND/NOR/XOR 시뮬레이션

- 목표
 - AND/OR/NOT/NAND/NOR/XOR 게이트의 결과값 확인
- 실험 내용
 - Xilinx ISE 14.7를 이용하여 아래의 schematic 구현
 - 시뮬레이션 결과와 진리표 확인
- 제출 사항
 - 구현한 schematic과 시뮬레이션 결과 스크린샷
 - 시뮬레이션 결과에 따른 각 게이트들의 진리표
- Schematic



실험 2 – AND/OR/INV 게이트를 이용한 XOR 게이트 구현

- 목표
 - 2-input AND, 2-input OR, Inverter 게이트를 이용하여 2-input XOR 게이트 구현
- 실험 내용
 - Xilinx ISE 14.7를 이용하여 2-input XOR 게이트 schematic 구현
 - 시뮬레이션 결과 확인
- 제출 사항
 - 구현한 XOR 게이트 schematic과 시뮬레이션 결과 스크린샷
 - 시뮬레이션에 따른 2-input XOR 게이트의 진리표

실험 3 – 복잡한 논리식 구현

■ 목표

- De Morgan 법칙을 이용해서 식 완성

$$\overline{a \cdot b + c} = (\quad + \quad) \cdot (\quad)$$

① ②

- ①, ②번 식을 각각 2-input AND, 2-input OR, Inverter 게이트를 이용해 구현하고, 시뮬레이션을 통해 결과 확인

■ 실험 내용

- De Morgan 법칙을 이용하여 ②번 식 유도
- Xilinx ISE 14.7를 이용하여 ①, ②번 식 각각 schematic 구현
- 시뮬레이션 결과 확인

■ 제출 사항

- De Morgan 법칙을 이용한 ①, ②번 식 유도 과정
- ①, ②번 식 각각에 대해 schematic과 시뮬레이션 결과 스크린샷
- ①번 식에 대한 진리표

실험과제 제출 안내

■ 보고서 포함 사항

- 실험 1 schematic과 시뮬레이션 결과 스크린샷, 진리표
- 실험 2 schematic과 시뮬레이션 결과 스크린샷, 진리표
- 실험 3 De Morgan 법칙을 이용한 식 유도 과정, schematic과 시뮬레이션 결과 스크린샷, 진리표
- 추가적인 내용은 자유롭게 작성
- 하나의 문서로 정리

■ 제출 방법 및 기한

- ETL 과제 게시판에 팀별로 제출
- 일요일 오후 6시까지