

Chapter 1 소개 및 툴 설치

1-1 Verilog HDL 소개

책 참고

1-2 Xilinx 및 Vivado 툴 소개

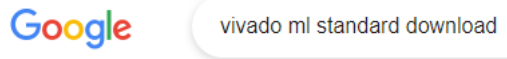
책 참고

1-3 CORA Z7 Kit 소개

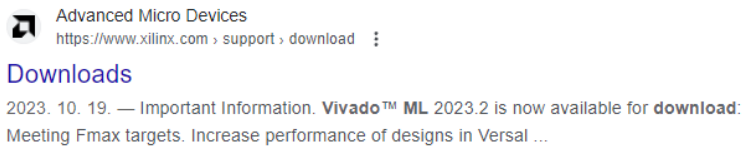
책 참고

1-4 Vivado 툴 설치

1. 다음과 같이 검색합니다.



2. 다음 사이트로 들어갑니다.



3. 다음과 같이 2022.2 버전을 선택합니다.

Version

2023.2

2023.1

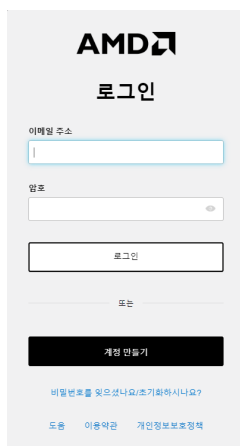
2022.2

2022.1

4. 페이지 아래로 조금 내려 다음 부분을 찾아 마우스 클릭합니다.



5. 다음과 같은 창이 나타납니다. 회원 가입을 위해 하단에 있는 [계정 만들기] 버튼을 마우스 클릭합니다.



6. 계정을 생성한 후, 로그인을 수행한 후, 다음과 같이 하단에 있는 [Download] 버튼을 누릅니다.

File Name:


Xilinx_Unified_2022.2_1014_8888_Win64.exe

If you are downloading the Vivado / Vitis unified installer, you will receive a follow-up confirmation email with a notice regarding our Developer Program.

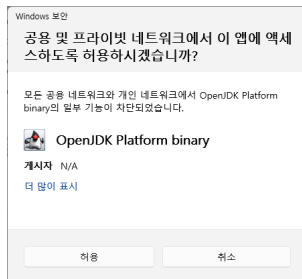
You can read about how we handle your personal data, your personal data rights, and how you can contact us in our [privacy policy](#).

Download

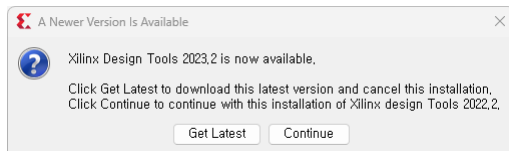
7. 그러면 다음과 같이 프로그램이 다운로드 됩니다. 마우스 클릭하여 실행시킵니다.

 Xilinx_Unified_2022.2_1014_8888_Win64.exe

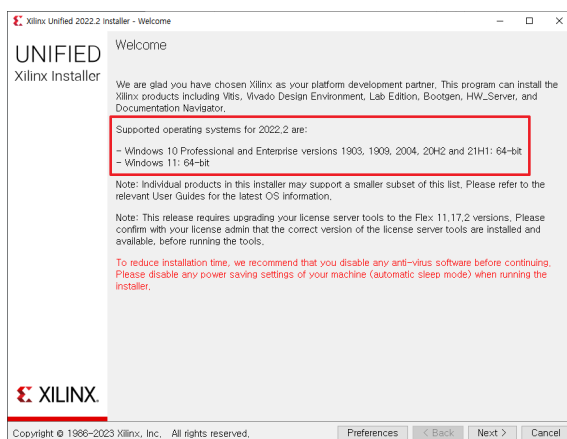
8. 다음과 같은 창이 뜨면 [허용] 버튼을 누릅니다.



9. 다음 창이 뜨면 [Continue] 버튼을 눌러줍니다.

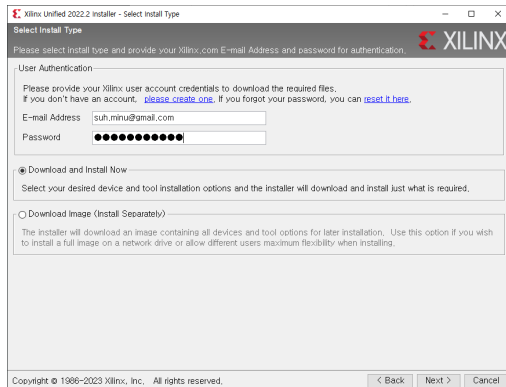


10. 다음과 같이 설치화면이 뜹니다. [Next] 버튼을 누릅니다.

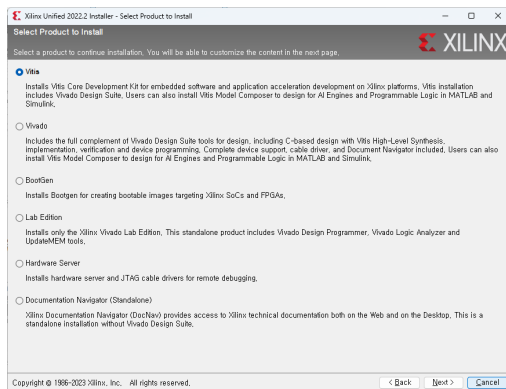


*** [Welcome] 창에서 현재 컴퓨터의 OS를 확인합니다.

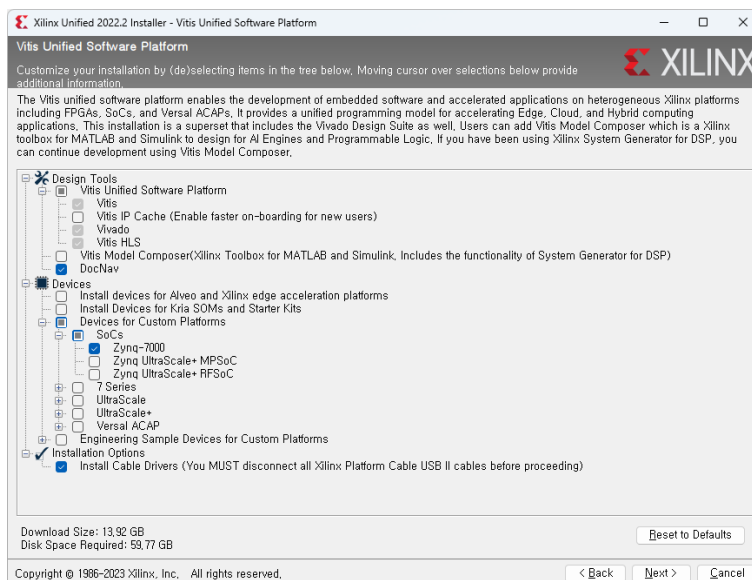
11. 다음과 같은 창이 나오면 앞에서 회원 가입 시 사용했던 아이디와 패스워드를 입력한 후, [Next] 버튼을 누릅니다.



12. 다음과 같은 창이 나오면 [Vitis]를 선택한 후, [Next] 버튼을 누릅니다.

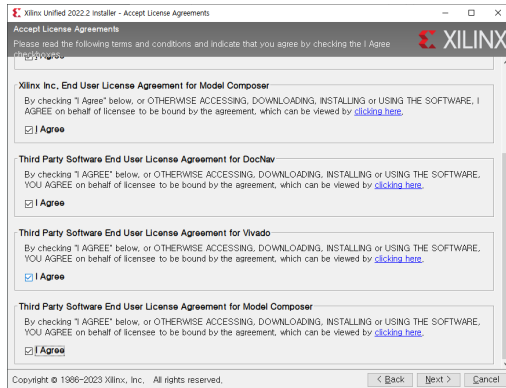


13. 다음과 같이 설치할 항목을 선택한 후 [Next] 버튼을 누릅니다.



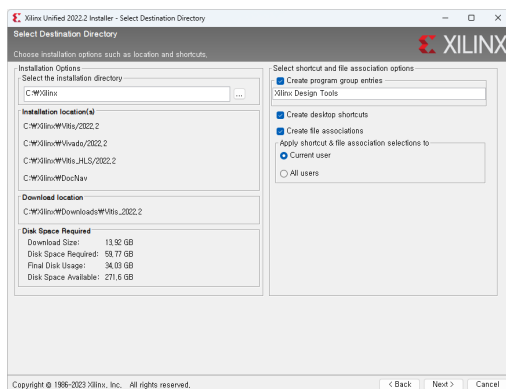
*** 이 책에서는 CORA Z7 보드를 교보재로 사용합니다.

14. 다음과 같은 창이 나오면 사용권 동의를 체크한 후, [Next] 버튼을 누릅니다.



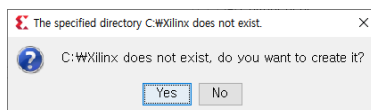
*** 우측 스크롤바를 하단으로 내려 동의 체크를 해 주도록 합니다.

15. 다음과 같이 설치 위치 창이 뜹니다. 기본 상태로 [Next] 버튼을 누릅니다.

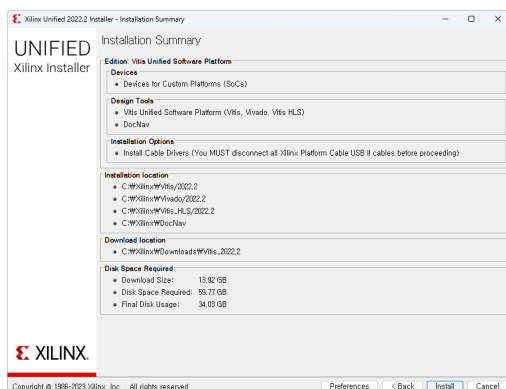


*** 사용자 필요에 따라 디렉터리를 변경할 수 있습니다.

16. 그러면 다음과 같은 창이 뜹니다. 디렉터리 생성창입니다. [Yes] 버튼을 누릅니다.



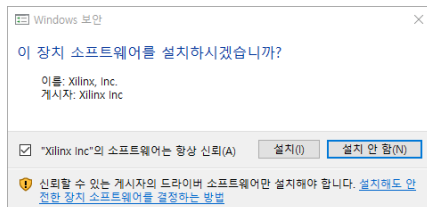
17. 그러면 다음과 같이 [설치 진행 요약] 창이 뜹니다. [Install] 버튼을 누릅니다.



18. 그러면 다음과 같이 설치가 진행됩니다.

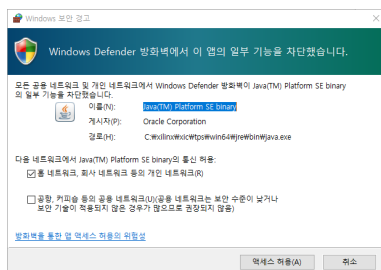


19. 중간에 다음과 같은 창이 뜨면 [설치(I)] 버튼을 누릅니다.

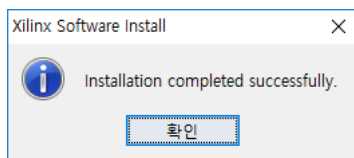


*** 보드 사용을 위한 드라이버 설치 창입니다.

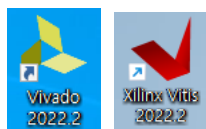
20. 다음과 같은 창이 뜨면 [액세스 허용(A)] 버튼을 누릅니다.



21. 다음과 같이 설치가 완료됩니다. [확인] 버튼을 누릅니다.



22. 바탕 화면에 다음과 같은 아이콘을 확인합니다.



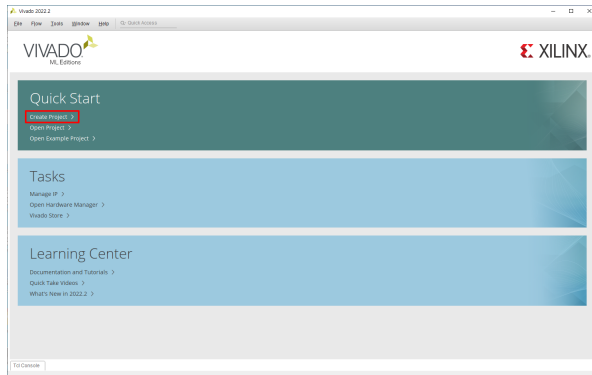
1-5 Vivado 프로젝트 만들기

Vivado를 이용하여 프로젝트를 생성해보도록 합니다.

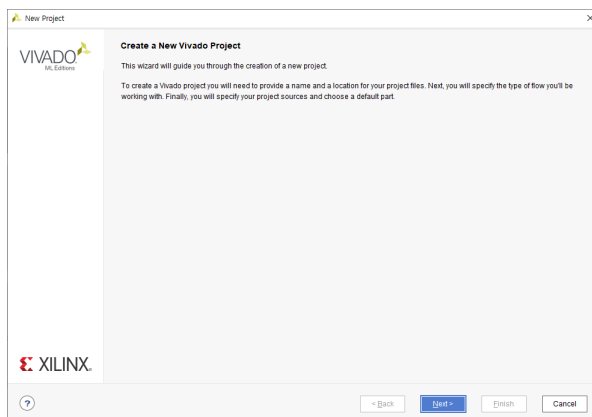
1. 다음 아이콘을 찾아 실행시킵니다.



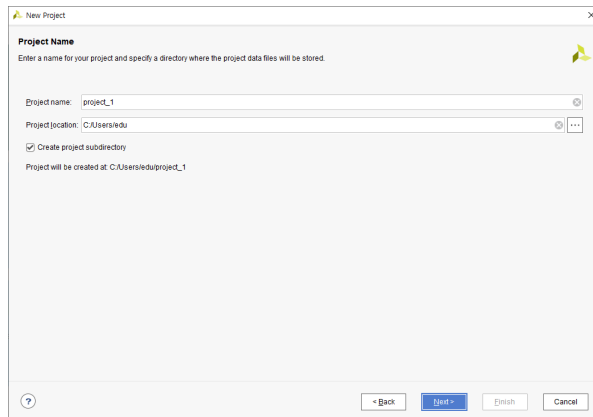
2. 다음과 같이 프로그램이 실행됩니다. [Quick Start] 하단에 있는 [Create Project]를 마우스 클릭합니다.



3. 다음과 같이 [새 프로젝트] 창이 뜹니다. [Next] 버튼을 누릅니다.

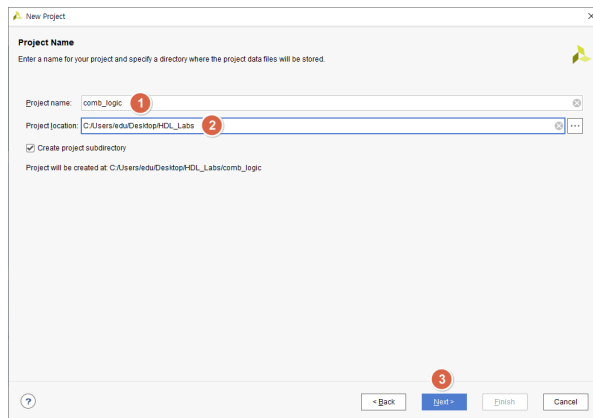


4. 다음과 같은 창이 뜹니다.

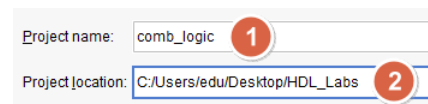


The 'New Project' dialog box is shown. It has a title bar with a yellow arrow icon and a close button. The main area is titled 'Project Name' and contains the instruction 'Enter a name for your project and specify a directory where the project data files will be stored.' There are two input fields: 'Project name:' with the value 'project_1' and 'Project location:' with the value 'C:/Users/edu'. Below these fields is a checked checkbox labeled 'Create project subdirectory'. At the bottom, there is a text line 'Project will be created at C:/Users/edu/project_1'. At the very bottom, there are four buttons: '?', '< Back', 'Next >', and 'Cancel'.

5. 다음과 같이 프로젝트 이름과 프로젝트 디렉터리를 정한 후, [Next] 버튼을 누릅니다.

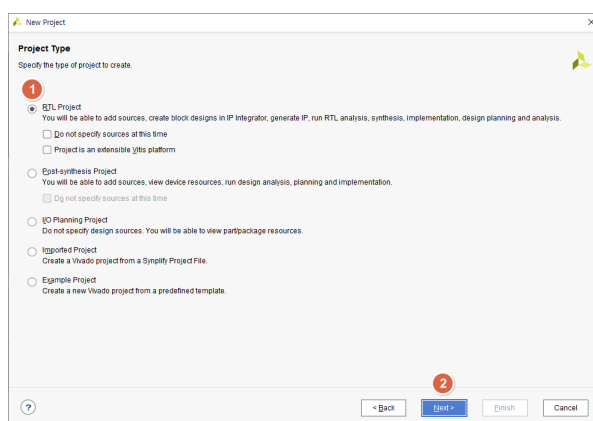


The 'New Project' dialog box is shown with red circles and numbers indicating the steps. Circle 1 is over the 'Project name:' field containing 'comb_logic'. Circle 2 is over the 'Project location:' field containing 'C:/Users/edu/Desktop/HDL_Labs'. Circle 3 is over the 'Next >' button.



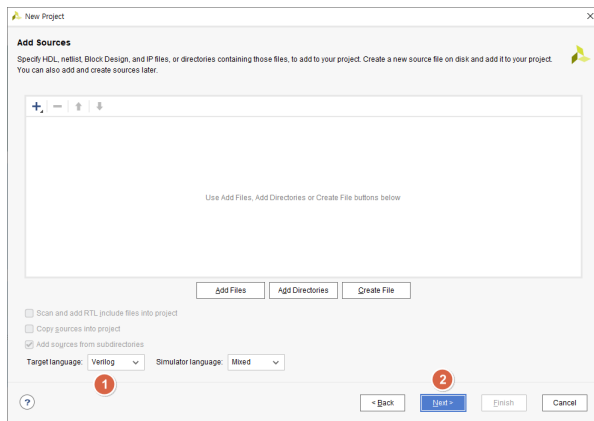
A close-up of the 'Project name:' and 'Project location:' fields. The 'Project name:' field contains 'comb_logic' and the 'Project location:' field contains 'C:/Users/edu/Desktop/HDL_Labs'. Red circles with numbers 1 and 2 are placed over the respective fields.

6. 다음은 [프로젝트 형태]를 선택하는 창입니다. [RTL Project]를 선택한 후, [Next] 버튼을 누릅니다.

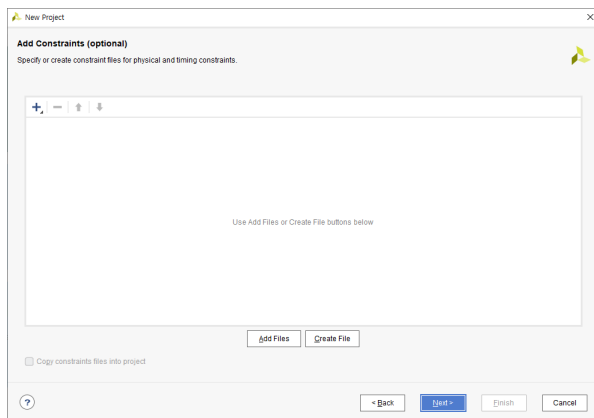


The 'Project Type' dialog box is shown. It has a title bar with a yellow arrow icon and a close button. The main area is titled 'Project Type' and contains the instruction 'Specify the type of project to create.' There are five radio button options: 'RTL Project' (selected), 'EPLD-synthesis Project', 'IO Planning Project', 'Imported Project', and 'Example Project'. Each option has a brief description. At the bottom, there are four buttons: '?', '< Back', 'Next >', and 'Cancel'.

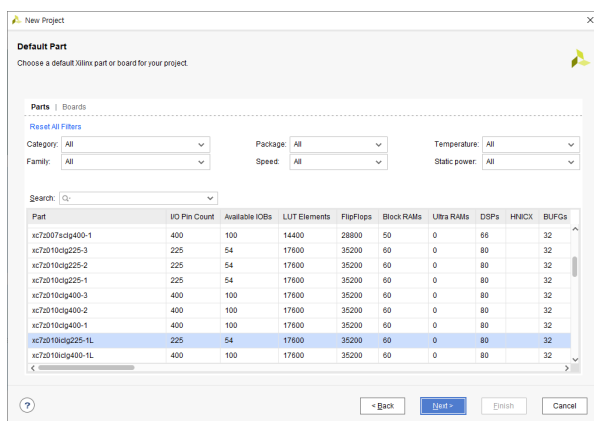
7. 다음은 [소스 추가] 창입니다. ❶과 같이 Verilog 언어를 선택한 후, ❷[Next] 버튼을 누릅니다.



8. 다음은 [제약 파일] 추가창입니다. 기본 상태에서 [Next] 버튼을 누릅니다.



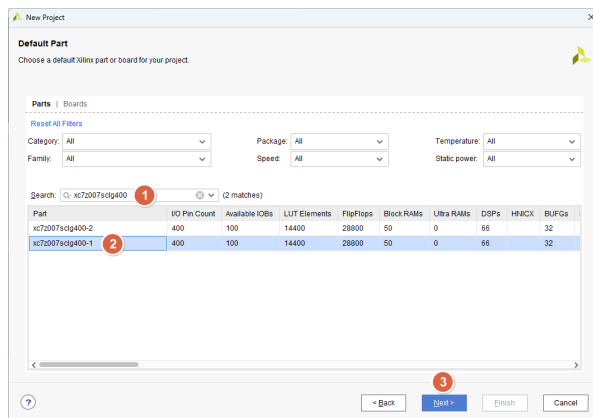
9. 다음은 [칩 선택] 창입니다.



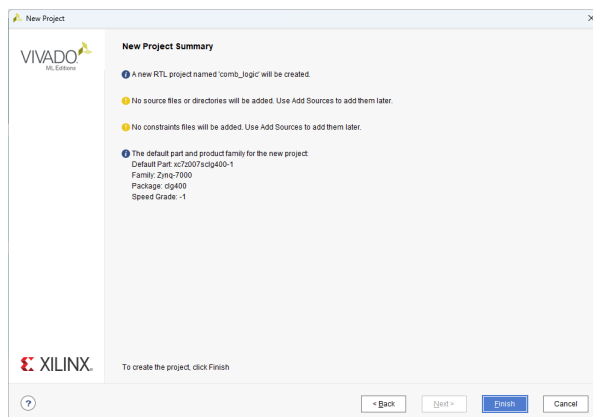
10. 보드 상에 있는 FPGA 칩의 이름을 확인합니다.



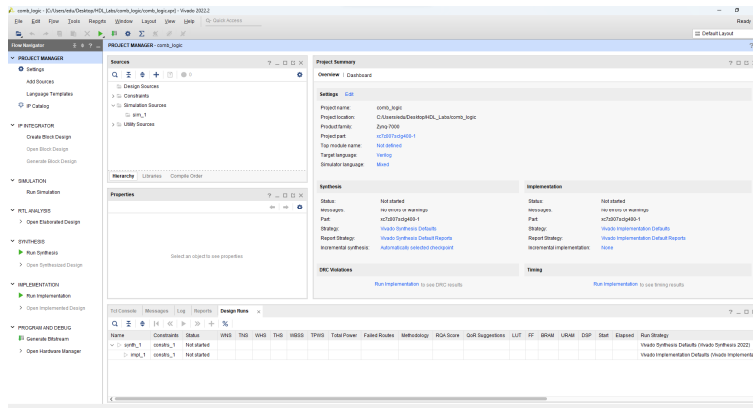
12. 칩 위에 있는 이름을 아래와 같이 [Search] 창에 입력합니다. [xc7z007sclg400-1]을 선택한 후, [Next] 버튼을 누릅니다.



13. 다음은 [새 프로젝트 요약] 창입니다. [Finish] 버튼을 누릅니다.

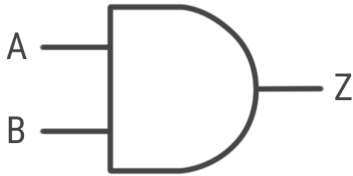


15. 다음과 같이 프로젝트가 생성됩니다.



1-6 Vivado Design Flow

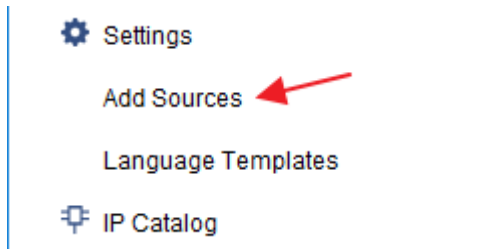
프로젝트에 Verilog 디자인 소스를 추가하여 CORA Z7 보드에 프로그래밍해 봅니다.



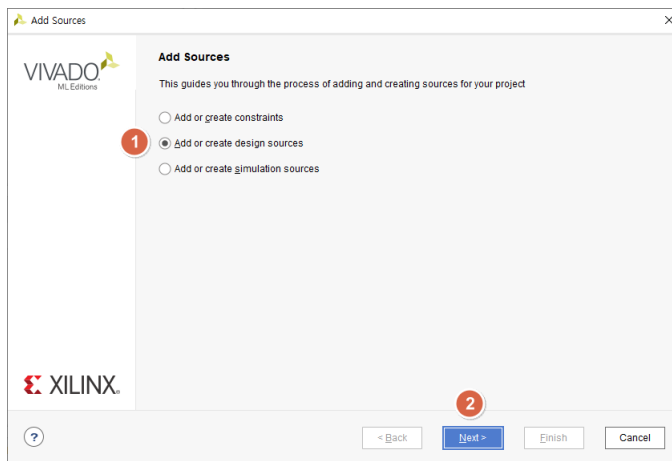
다음과 같이 좌측에 있는 [Flow Navigator]를 기준으로 설계를 진행합니다.



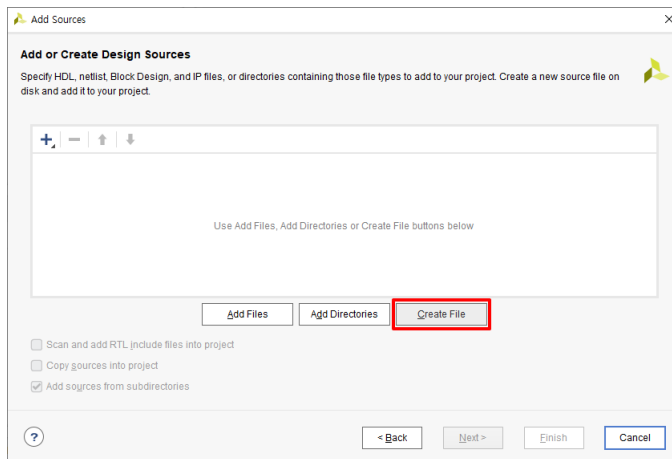
1. 다음과 같이 [Flow Navigator]의 상단에 있는 [Add Sources]를 마우스 클릭합니다.



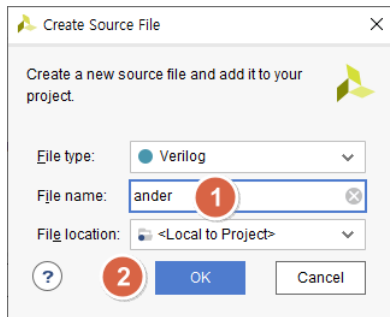
2. 그러면 다음과 같은 창이 뜹니다. [Add or create design sources]를 선택한 후, [Next] 버튼을 누릅니다.



3. 다음 창에서 [Create File] 버튼을 누릅니다.



4. 그러면 다음과 같은 창이 뜹니다. [ander]라고 입력해줍니다.



Create a new source file and add it to your project.

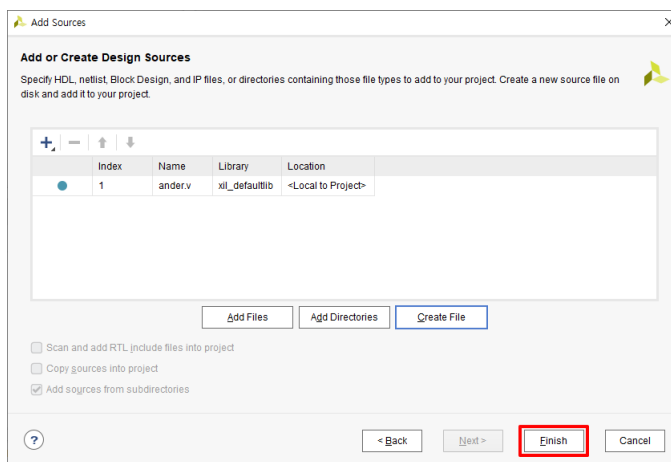
File type: Verilog

File name: ander **1**

File location: <Local to Project>

? **2** OK Cancel

5. 그러면 다음창으로 돌아옵니다. [Finish] 버튼을 누릅니다.



Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

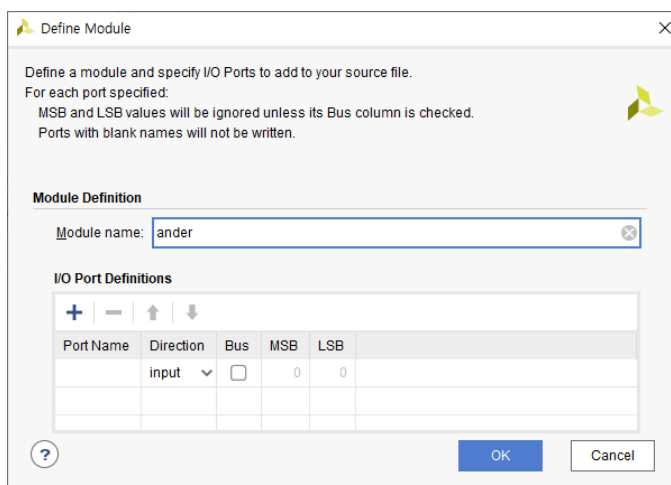
	Index	Name	Library	Location
	1	ander.v	xil_defaultlib	<Local to Project>

Add Files Add Directories Create File

☐ Scan and add RTL include files into project
☐ Copy sources into project
☒ Add sources from subdirectories

? < Back Next > **Finish** Cancel

6. 그러면 다음과 같이 [모듈 정의] 창이 뜹니다.



Define a module and specify I/O Ports to add to your source file.

For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

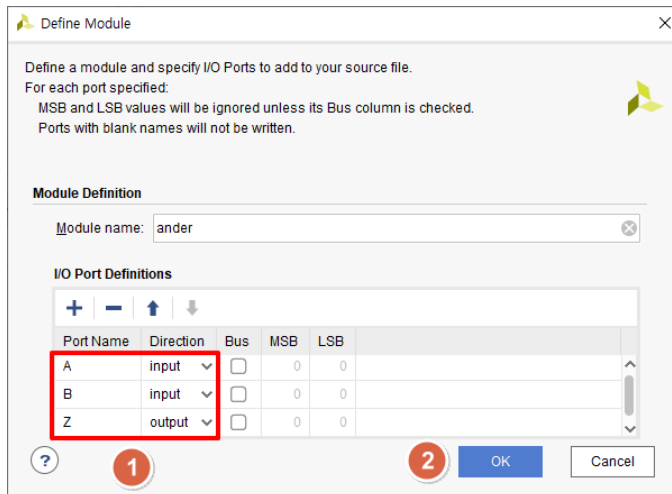
Module name: ander

I/O Port Definitions

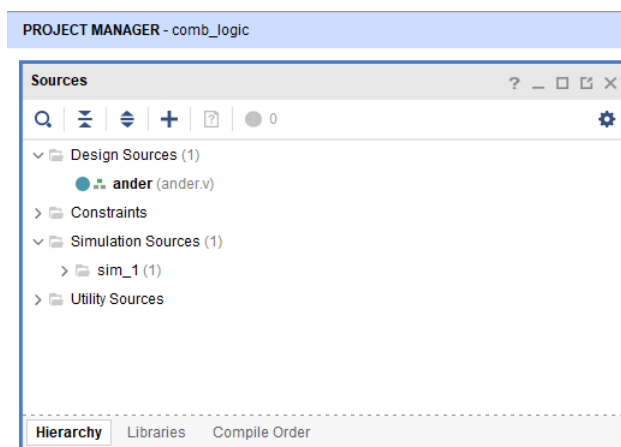
Port Name	Direction	Bus	MSB	LSB
	input	<input type="checkbox"/>	0	0

? OK Cancel

7. 다음과 같이 입출력 포트를 설정한 후, [OK] 버튼을 누릅니다.



8. 다음과 같이 소스가 추가된 것을 확인합니다.



9. 다음과 같이 소스의 내용을 확인합니다.

```
22 |
23 | module ander(
24 |     input A,
25 |     input B,
26 |     output Z
27 | );
28 | endmodule
29 |
```

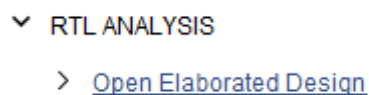
소스 작성하기

10. 다음과 같이 소스를 작성합니다.

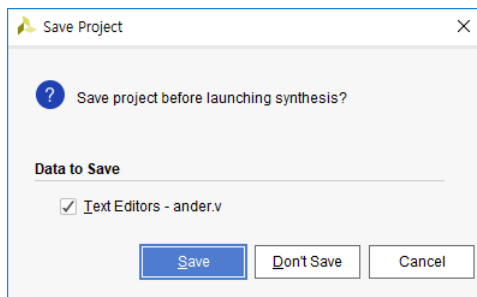
```
23 module ander(  
24     input A,  
25     input B,  
26     output Z  
27 );  
28  
29 assign Z = A & B;  
30  
31 endmodule
```

RTL 스키매틱 확인하기

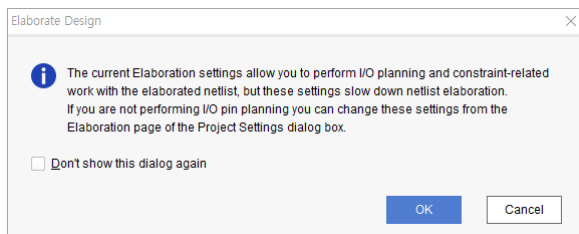
11. [RTL ANALYSIS]--[Open Elaborated Design]을 마우스 클릭합니다.



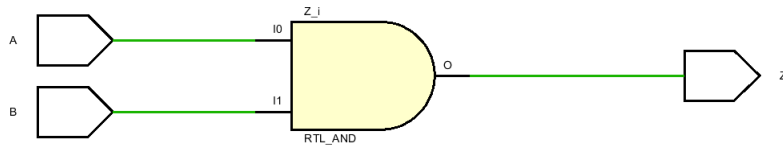
12. 다음과 같은 창이 뜨면 [Save] 버튼을 눌러 소스를 저장합니다.



13. 다음과 같은 창이 뜨면 [OK] 버튼을 누릅니다.

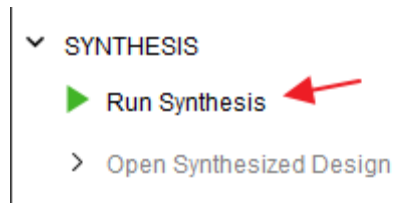


14. 다음과 같이 RTL 레벨의 [Schematic]을 확인합니다.

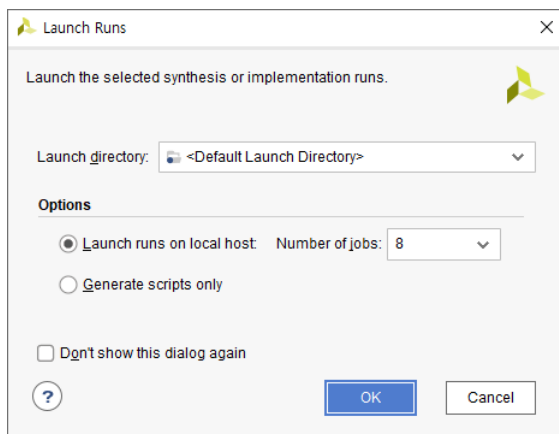


합성하기

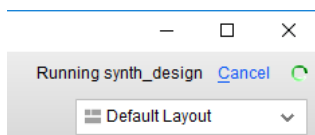
15. 다음과 같이 [SYNTHESIS]--[Run Synthesis]를 마우스 클릭합니다.



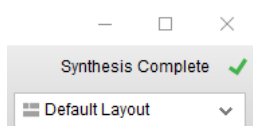
16. 다음과 같은 창이 뜹니다. [OK] 버튼을 누릅니다.



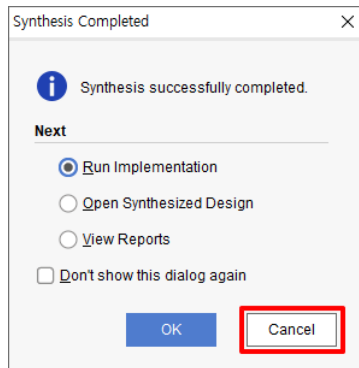
17. 우측 상단에 합성 진행 상태가 표시됩니다. 시간이 몇 분 정도 걸립니다.



18. 합성이 끝나면 다음과 같이 표시됩니다.

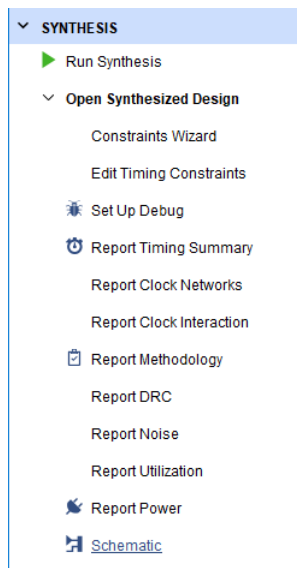


19. 다음과 같은 창이 뜹니다. [Cancel] 버튼을 누릅니다.

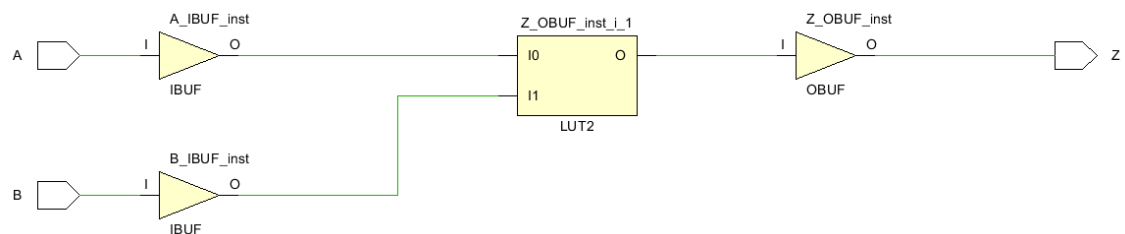


합성 스키매틱 보기

20. [SYNTHESIS]--[Open Synthesized Design]--[Schematic]을 눌러줍니다.



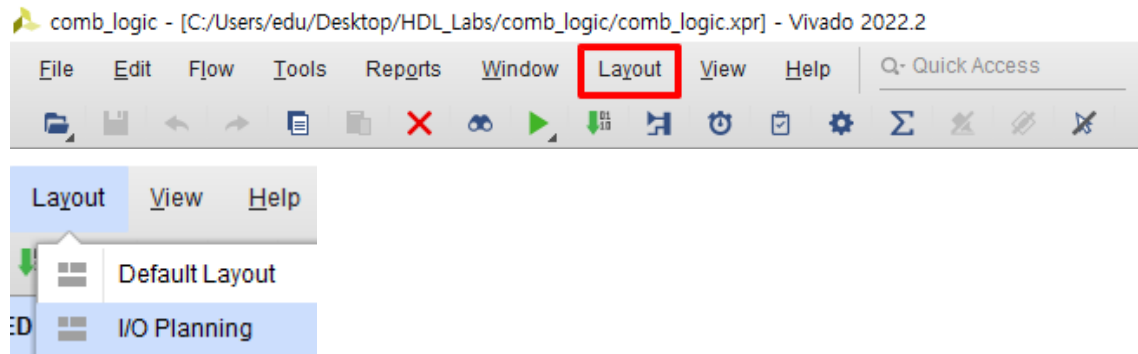
21. FPGA 블록 기반의 합성 스키매틱을 확인합니다.



핀 배치하기

여기서는 핀 배치를 해 봅니다.

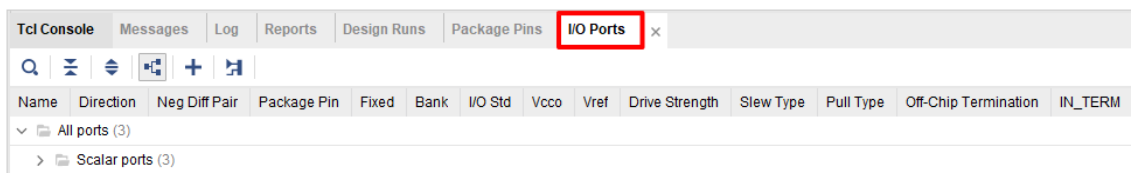
1. 다음과 같이 [Layout]--[I/O Planning]을 선택합니다.



2. 다음 [Package]창을 확인합니다.



3. 하단에 있는 [I/O Ports]를 선택합니다.



4. CORA Z7 보드의 핀을 확인합니다.

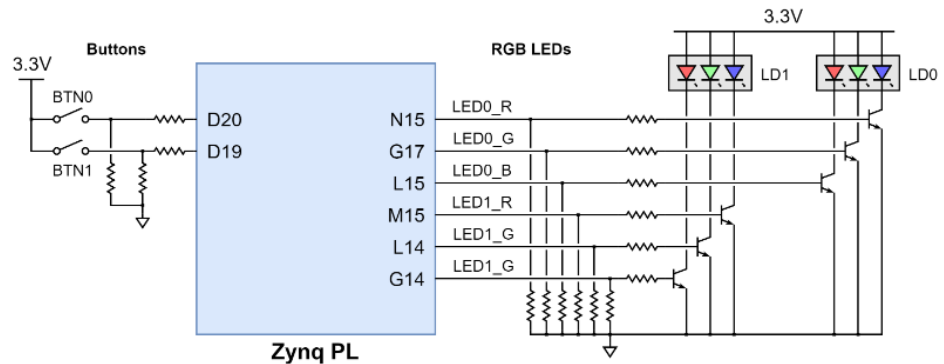


Figure 11.1. Cora Z7 Basic I/O

*** 다음 사이트를 참조합니다.

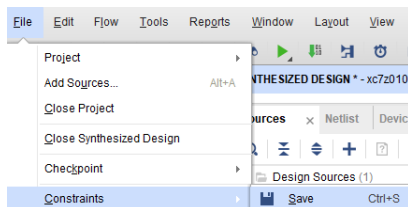
<https://digilent.com/reference/programmable-logic/cora-z7/reference-manual>

5. 다음과 같이 [Package Pin] 란에 핀을 설정해줍니다. [I/O Std]는 LVCMOS33*로 해줍니다. 버튼과 LED가 3.3V를 사용하도록 보드가 설계되어 있습니다.

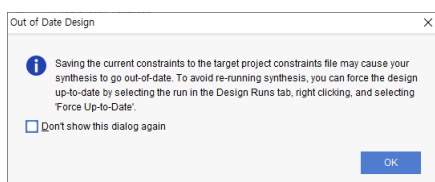
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM
All ports (3)													
Scalar ports (3)													
IN			D20	<input checked="" type="checkbox"/>	35	LVCMOS33*	3.300				NONE	NONE	
IN			D19	<input checked="" type="checkbox"/>	35	LVCMOS33*	3.300				NONE	NONE	
OUT			N15	<input checked="" type="checkbox"/>	35	LVCMOS33*	3.300		12		NONE	FP_VTT_50	



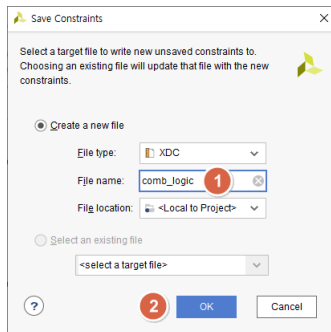
6. 다음과 같이 파일로 저장합니다.



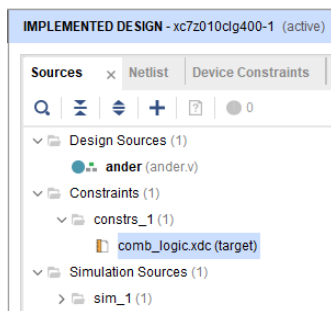
7. 다음과 같은 창이 뜨면 [OK] 버튼을 누릅니다.



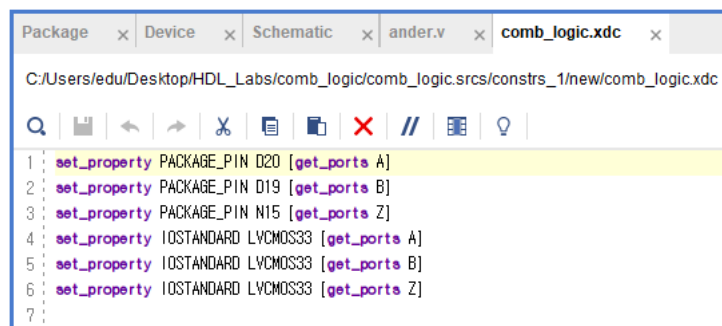
8. 다음과 같은 창이 뜨면 [OK] 버튼을 눌러줍니다.



9. 다음과 같이 파일이 생성된 것을 확인합니다.

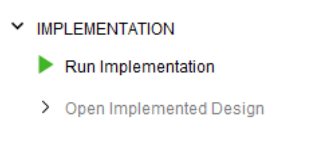


10. 다음은 파일의 내용입니다.

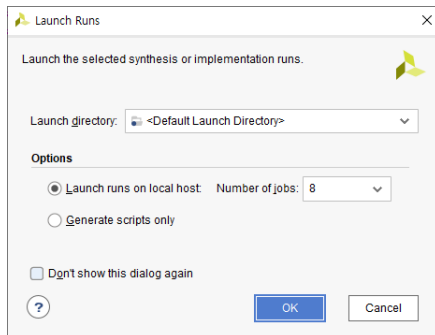


구현하기

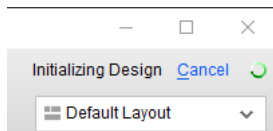
11. 다음과 같이 [IMPLEMENTATION]--[Run Implementation]을 선택합니다.



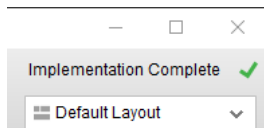
12. 다음과 같은 창이 뜹니다. [OK] 버튼을 누릅니다.



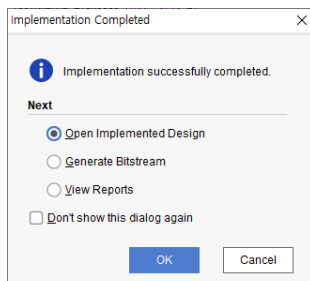
13. 우측 상단에서 구현이 진행되는 것을 확인할 수 있습니다. 시간이 걸립니다.



14. 다음과 같이 구현이 완료됩니다.

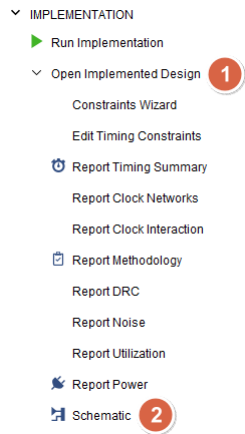


15. 다음과 같은 창이 뜹니다. [Cancel] 버튼을 누릅니다.

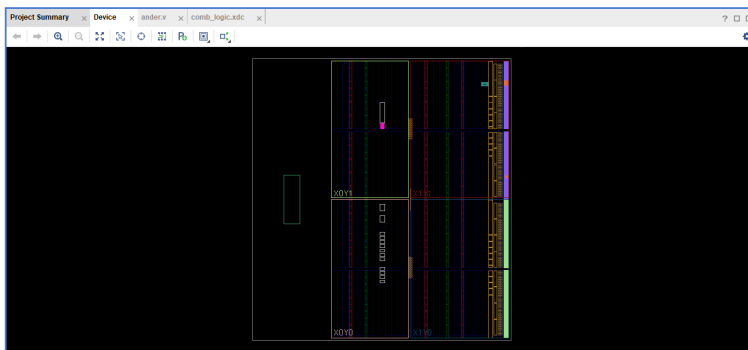


FPGA 내부 확인하기

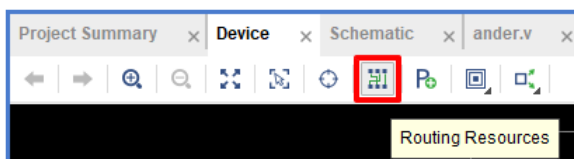
16. [IMPLEMENTATION]--[Open Implemented Design]--[Schematic]을 선택합니다.



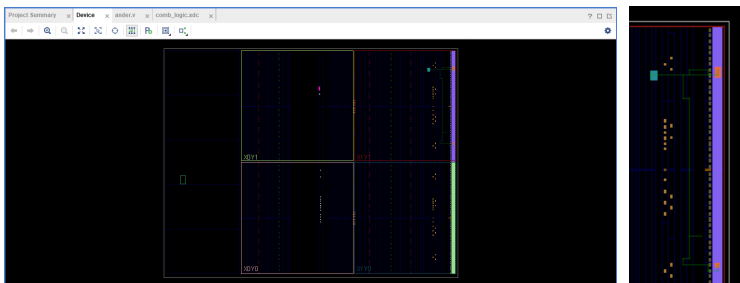
17. 다음 [Device] 창을 확인합니다.



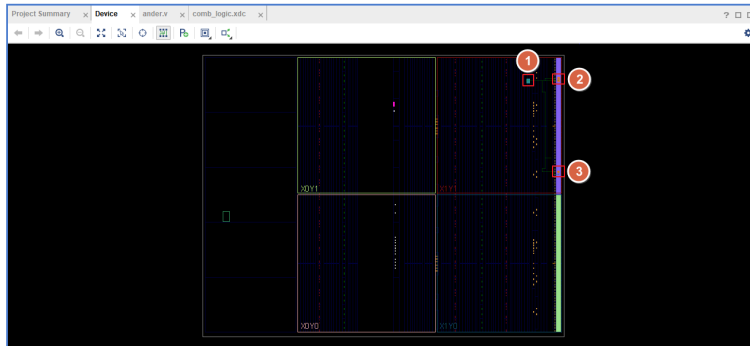
18. [Routing Resources] 아이콘을 눌러줍니다.



19. 그러면 다음과 같이 배선을 볼 수 있습니다.



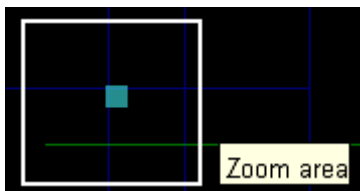
다음 부분을 확대해 봅니다.



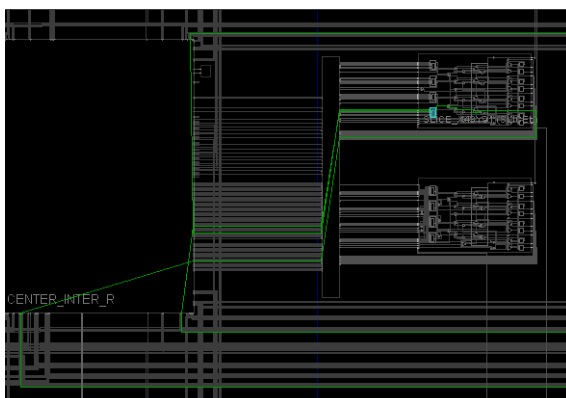
20. 파란 블록 주변으로 마우스 왼쪽 버튼을 눌러 사각형을 만든 후, 마우스 버튼을 놓습니다.




21. 그러면 사각형이 확대됩니다. 한 번 더 같은 동작을 반복합니다.



22. 그러면 다음과 같이 확대되는 것을 볼 수 있습니다.

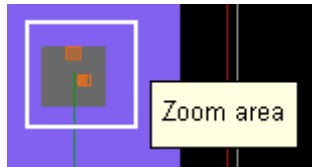


Zoom Fit (Ctrl+0)

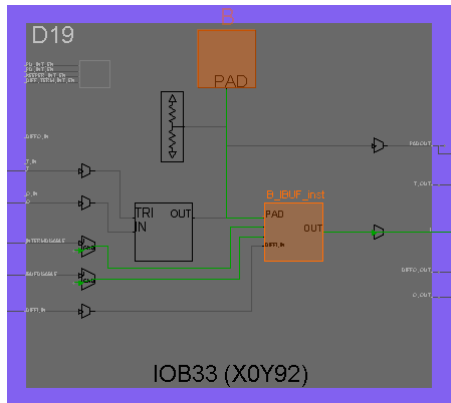


Zoom area

29. I/O 블록을 좀 더 확대해봅니다.

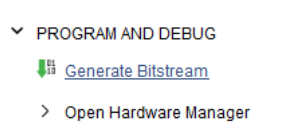


30. 다음과 같이 확대됩니다. I/O 블록 내부를 자세히 볼 수 있습니다.

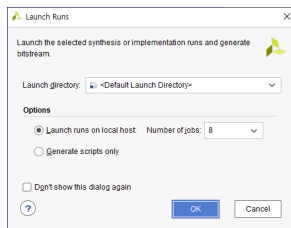


비트스트림 생성하기(FPGA 이미지)

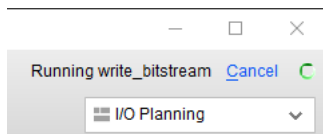
1. 다음과 같이 [PROGRAM AND DEBUG]--[Generate Bitstream]을 선택합니다.



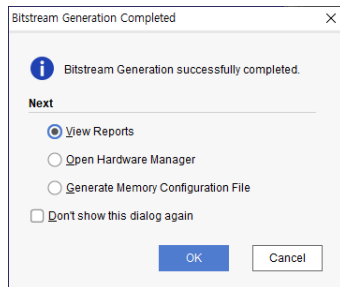
2. 다음과 같은 창이 뜹니다. [OK] 버튼을 눌러줍니다.



3. 우측 상단에서 작업 진행을 확인할 수 있습니다.

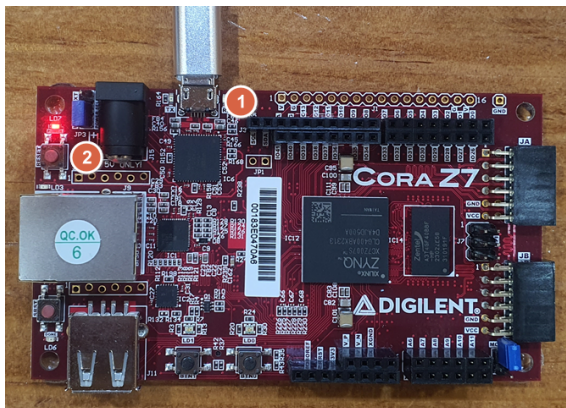


4. 다음과 같이 창이 뜹니다. 일단 [Cancel] 버튼을 눌러줍니다.



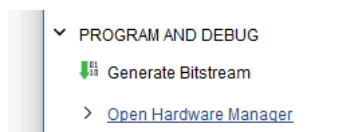
보드 연결하기

1. 다음과 같이 CORA Z7 보드를 PC에 연결한 후, 전원을 켭니다.



FPGA 이미지 다운로드하기

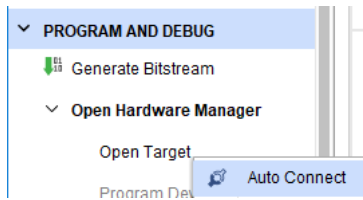
1. 다음과 같이 [PROGRAM AND DEBUG]--[Open Hardware Manager]를 선택합니다.



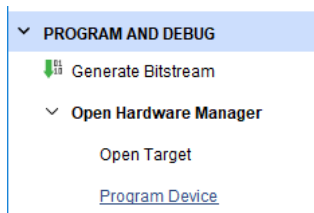
2. [Open Target]을 선택합니다.



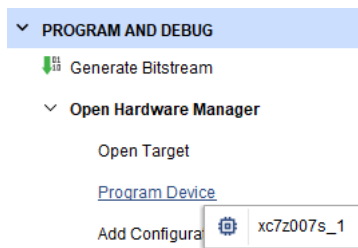
4. [Auto Connect]를 선택합니다.



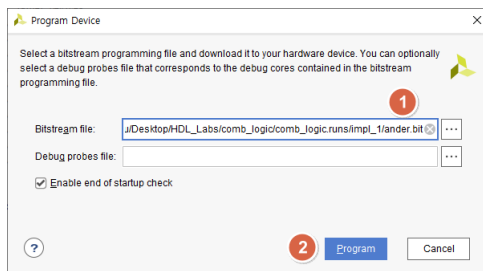
5. [Program Device]를 선택합니다.



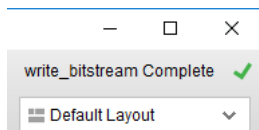
6. [xc7a35t_0]을 선택합니다.



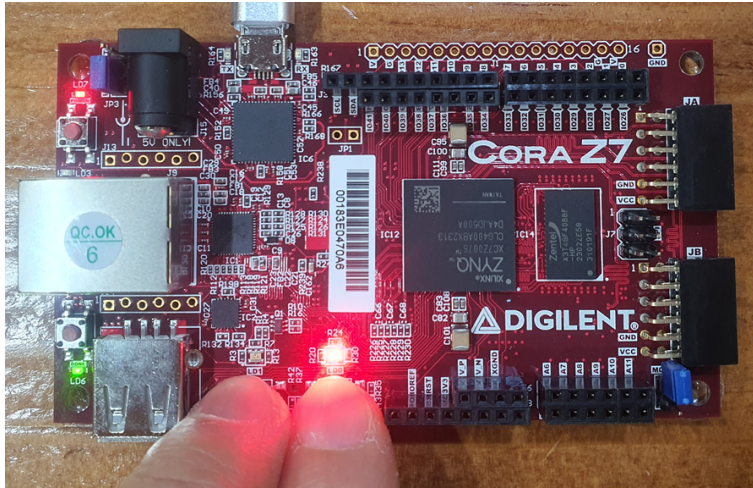
7. 다음과 같은 창이 뜹니다. ander.bit 파일을 확인 후, [Program] 버튼을 누릅니다.



9. 다음과 같이 완료된 것을 확인합니다.



10. 2 개의 버튼을 눌러 LED가 들어오는 것을 확인합니다.




bit 파일 확인하기

bit 파일은 FPGA 설정 파일로 다음과 같이 확인할 수 있습니다.

1, 다음과 같이 디렉터리를 찾아갑니다.

```
바탕 화면 > HDL_Labs > comb_logic > comb_logic.runs > impl_1 >
```

2. 다음 파일을 확인합니다.

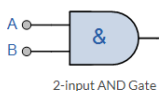
 ander.bit	2023-03-11 오전 9:19	BIT 파일	2,036KB
---	--------------------	--------	---------

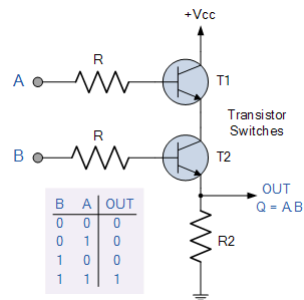
논리 로직의 트랜지스터 구조

AND

Digital Logic "AND" Gate Types

The 2-input Logic AND Gate

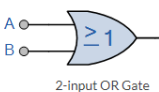
Symbol	Truth Table		
 <p>2-input AND Gate</p>	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression $Q = A \cdot B$		Read as A AND B gives Q	

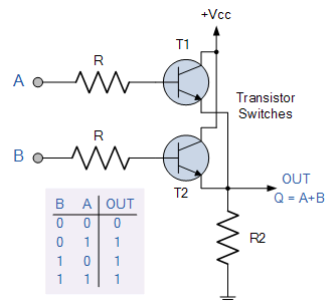


OR

Digital Logic "OR" Gate Types


The 2-input Logic OR Gate

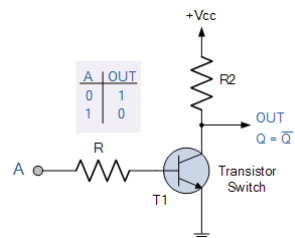
Symbol	Truth Table		
 <p>2-input OR Gate</p>	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression $Q = A + B$		Read as A OR B gives Q	



NOT

The Logic NOT Gate Truth Table

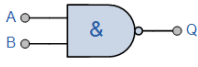
Symbol	Truth Table	
 <p>Inverter or NOT Gate</p>	A	Q
	0	1
	1	0
Boolean Expression $Q = \text{not } A \text{ or } \bar{A}$	Read as inverse of A gives Q	

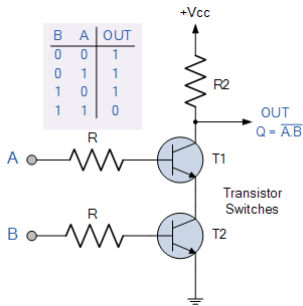


NAND

The Digital Logic “NAND” Gate

2-input Logic NAND Gate

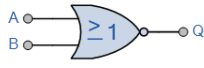
Symbol	Truth Table		
 2-input NAND Gate	B	A	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A \cdot B}$		Read as A AND B gives NOT Q	



NOR

The Digital Logic “NOR” Gate

2-input NOR Gate

Symbol	Truth Table		
 2-input NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression $Q = \overline{A + B}$		Read as A OR B gives NOT Q	

