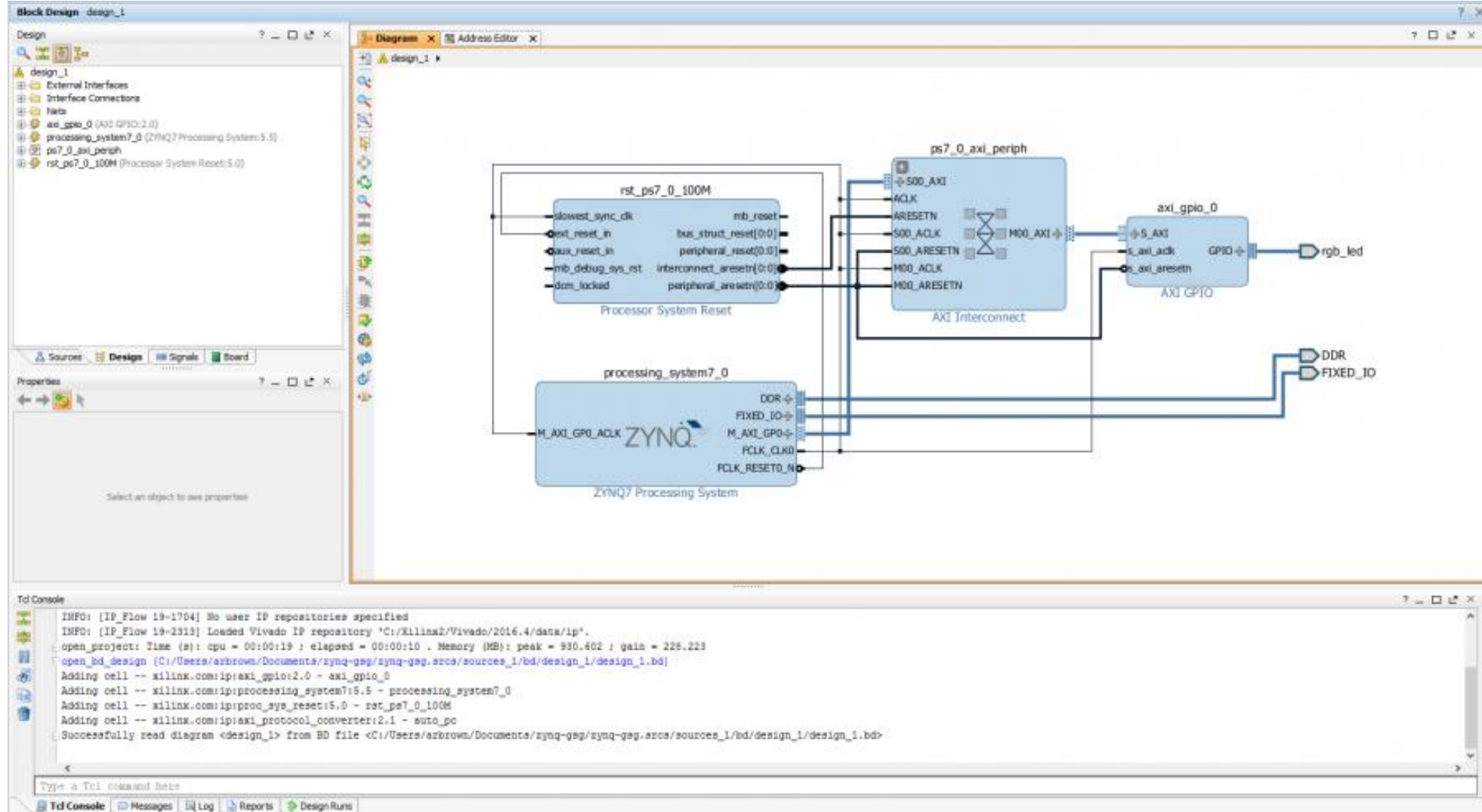


Getting Started with the Vivado IP Integrator

Innova Lee(이상훈)
gcccompil3r@gmail.com



Prerequisites

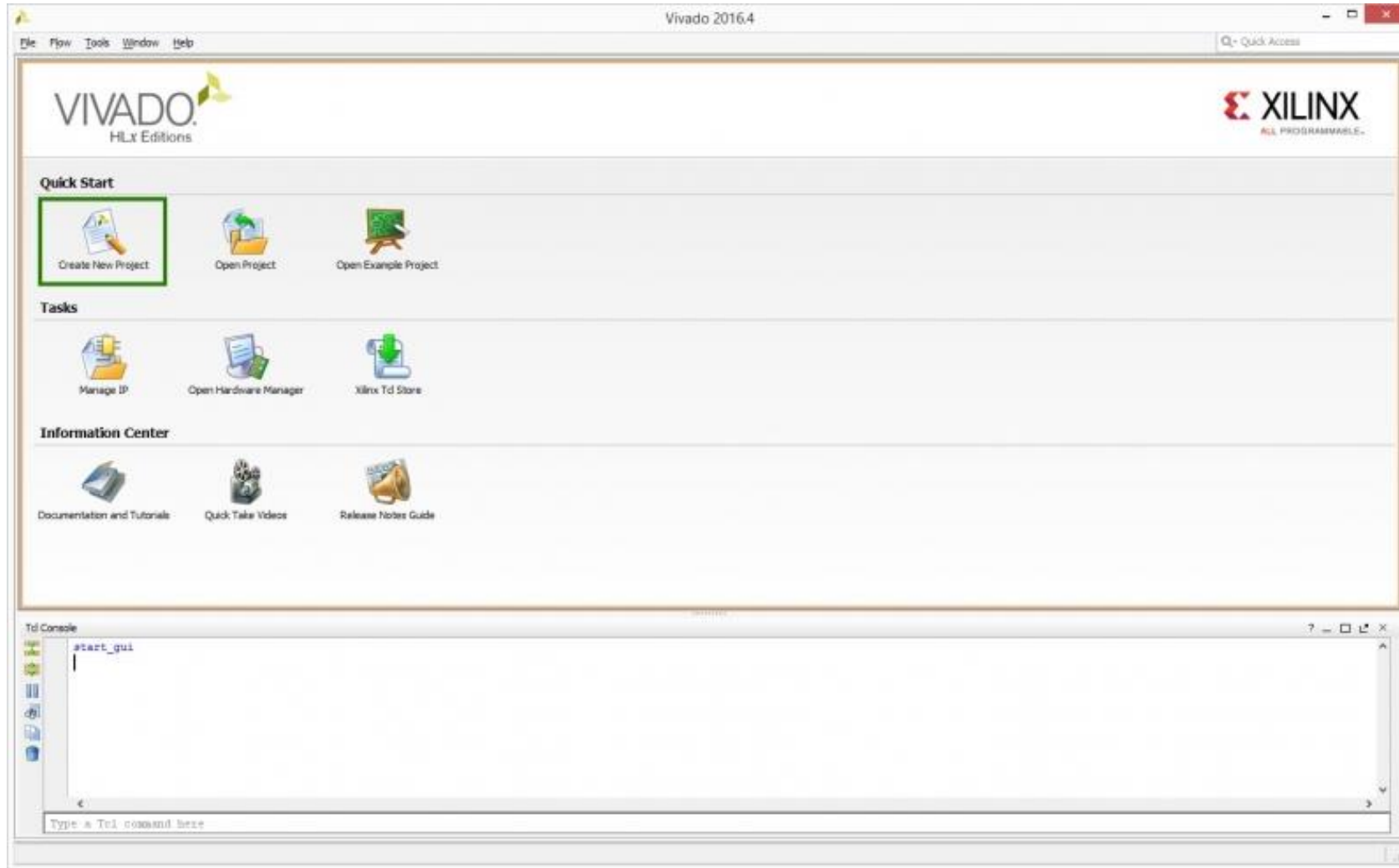
- Digilent FPGA 보드
- Vivado 가 설치되어 있고 Vivado 를 설치하고자 한다면 아래 튜토리얼을 참조하라.
<https://reference.digilentinc.com/learn/software/tutorials/vivado-install-guide/start>
 - 이 안내서는 버전 2016.4 를 사용하며 다른 버전도 사용할 수 있지만 차이가 있을 수 있다.
- Digilent Board Files 는 복잡한 구성 설정 선택을 도와준다.
<https://reference.digilentinc.com/reference/software/vivado/board-files#installation>
- Vivado 에 대한 기본 지식은 아래 튜토리얼을 참조하라.
https://reference.digilentinc.com/vivado/getting_started/start

Introduction

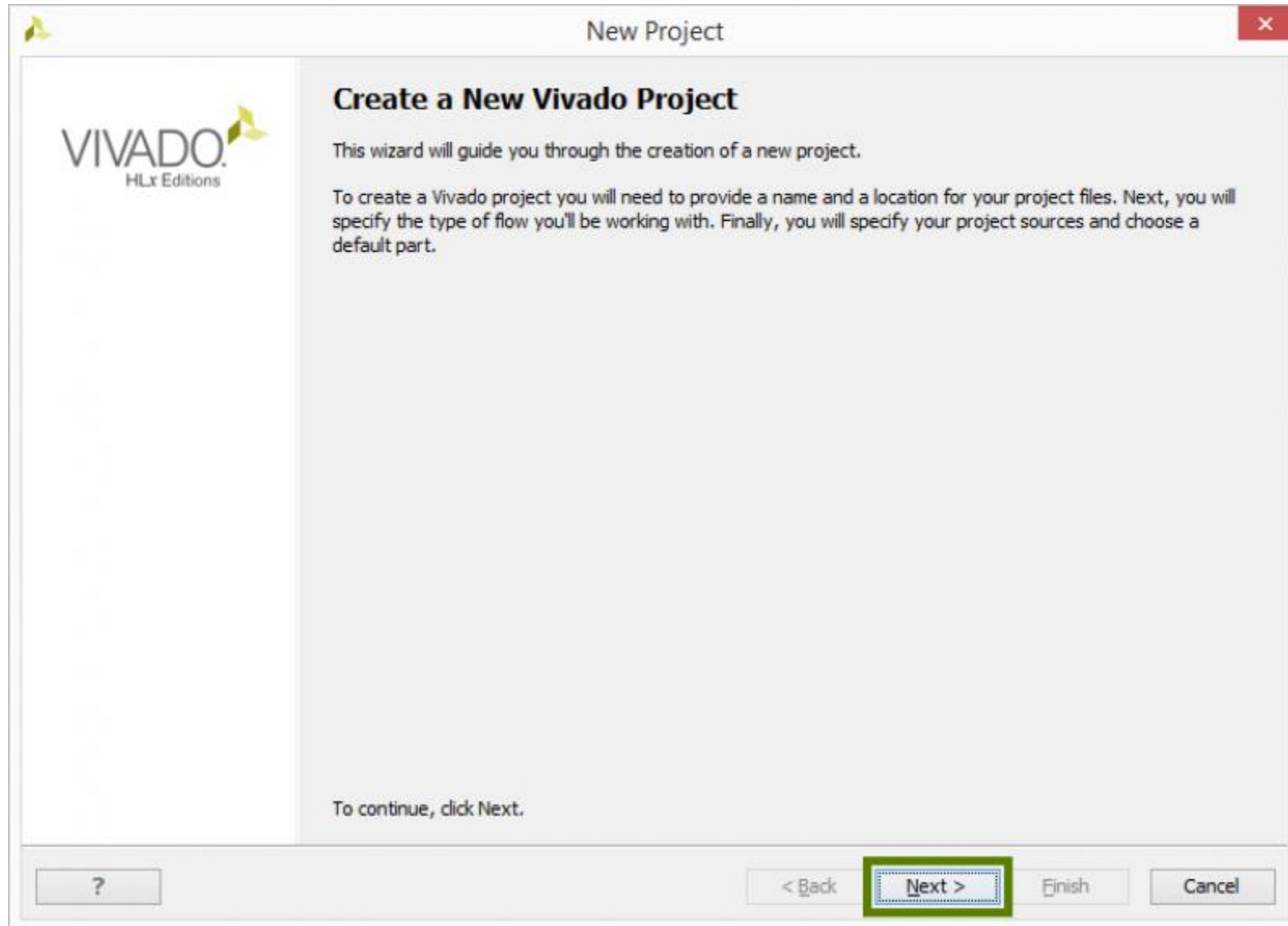
이 가이드는 Digilent FPGA 보드에 대한 IP Integrator Design Flow 의 주요 기능 대부분을 사용하는 방법을 설명한다. 데모가 끝나면 보드 상의 스위치 및 제어 LED 에서 데이터를 수신하는 간단한 Design 이 만들어진다.

1. Create a New IPI Project

1.1 Vivado 를 새로 연 다음 Create New Project 를 클릭한다.



1.2 이 마법사의 첫 번째 페이지에는 프로젝트를 만드는 단계가 요약되어 있다.
Next 를 클릭한다.



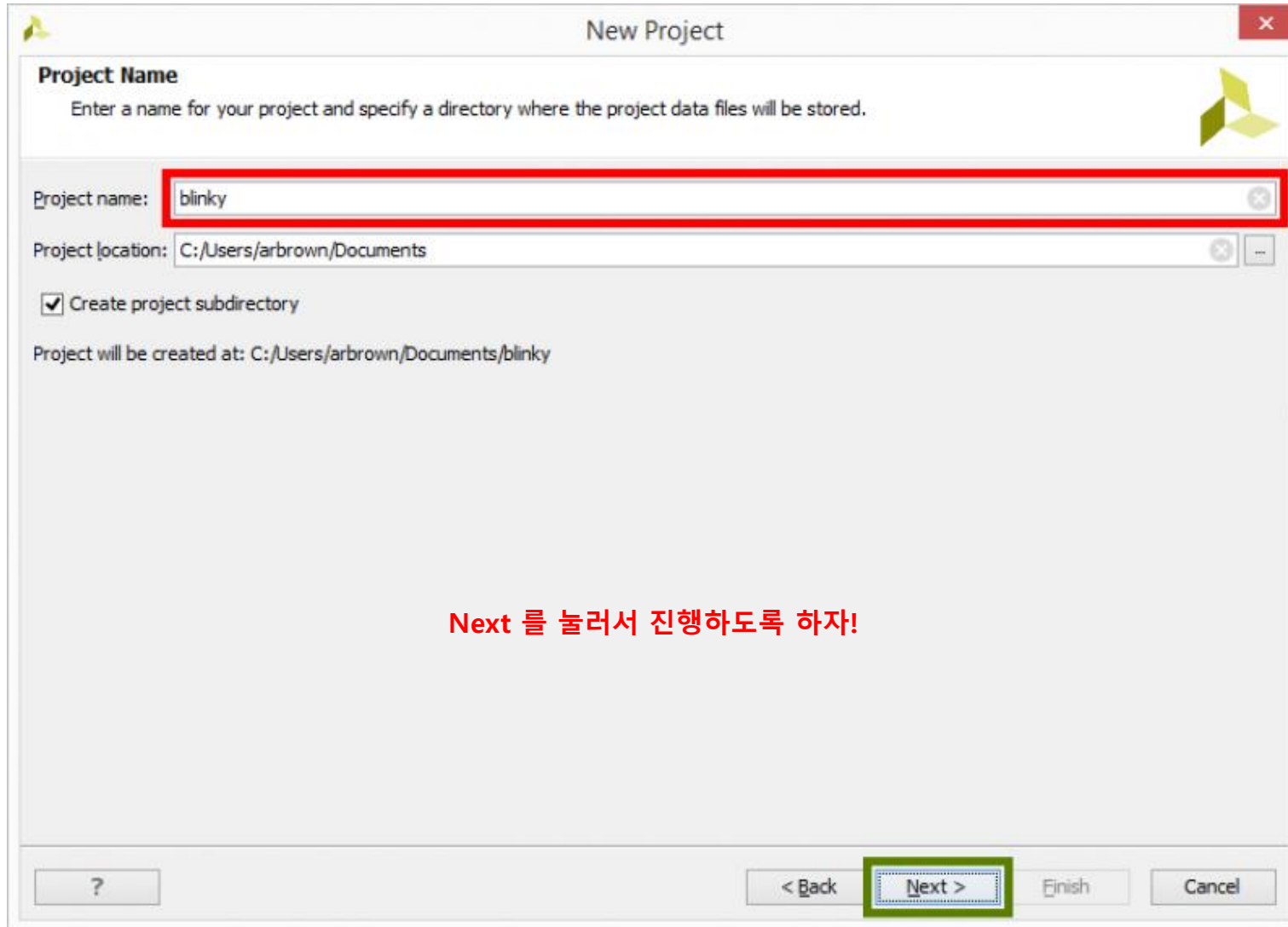
1.3 첫 번째 단계는 프로젝트의 이름을 설정하는 것이다.
Vivado 는 폴더 구조를 생성할 때 이 이름을 사용한다.

*** Important**

프로젝트 이름이나 위치 경로에 공백을 사용하지 말라.

이것은 Vivado 에 문제를 일으키므로 대신 밑줄, 대시 또는 CamelCase 를 사용하도록 하라.

https://en.wikipedia.org/wiki/Camel_case



The image shows the 'New Project' dialog box in Vivado. The 'Project Name' field is highlighted with a red rectangle and contains the text 'blinky'. The 'Project location' field contains 'C:/Users/arbrown/Documents'. The 'Create project subdirectory' checkbox is checked. The text 'Project will be created at: C:/Users/arbrown/Documents/blinky' is displayed. At the bottom, the 'Next >' button is highlighted with a green rectangle.

Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name:

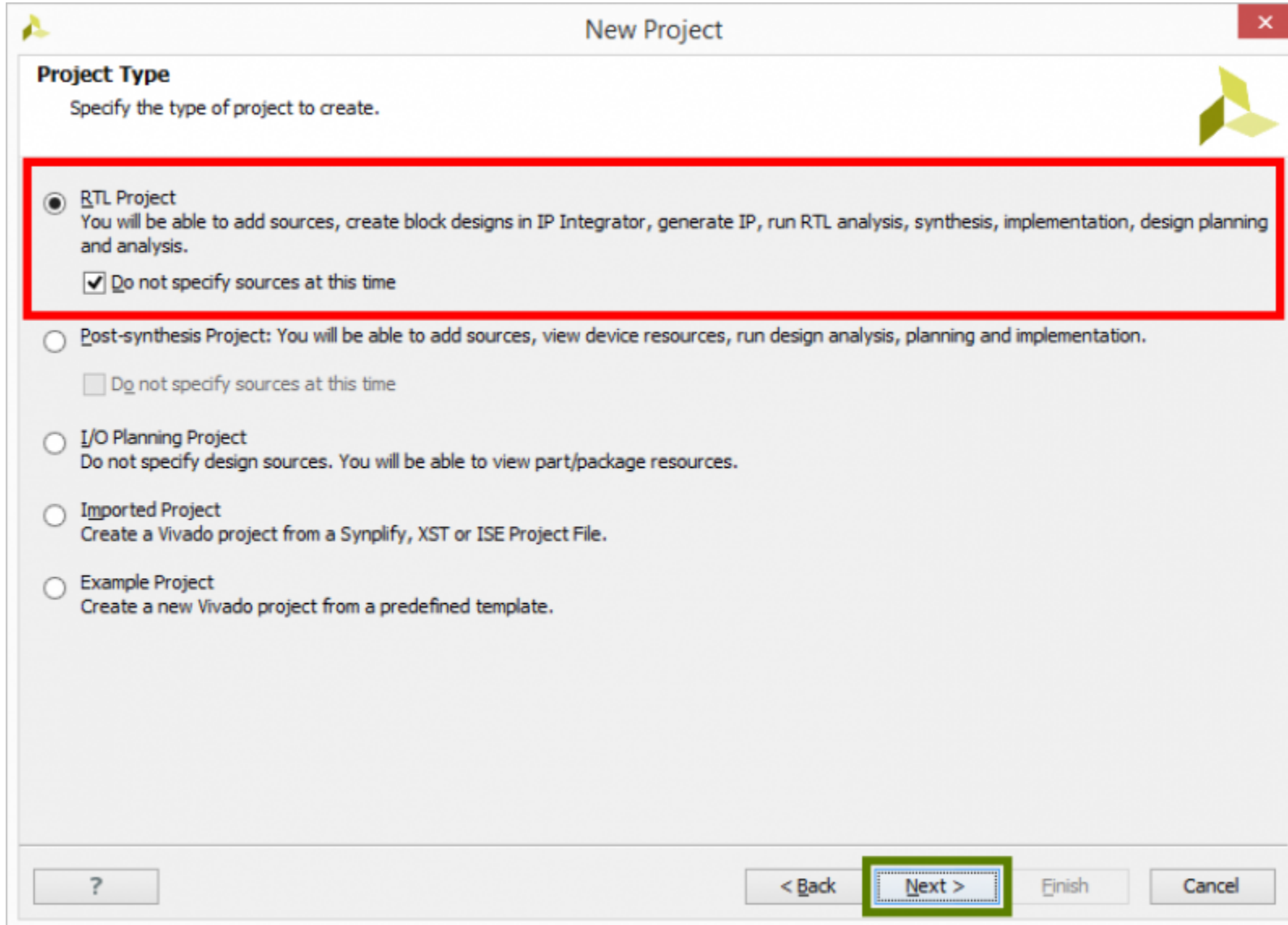
Project location:

☒ Create project subdirectory

Project will be created at: C:/Users/arbrown/Documents/blinky

Next 를 눌러서 진행하도록 하자!

- 1.4 프로젝트 유형 선택 화면에서 RTL 프로젝트를 선택하고 Do not specify sources at this time 상자를 체크한다.
고급 사용자는 이 화면에서 다른 옵션을 사용하려고 할 수 있지만 이 가이드에서는 다루지 않는다.
Next 를 클릭하여 계속한다.



New Project

Project Type
Specify the type of project to create.

☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
☒ Do not specify sources at this time

☐ **Post-synthesis Project:** You will be able to add sources, view device resources, run design analysis, planning and implementation.
☐ Do not specify sources at this time

☐ **I/O Planning Project**
Do not specify design sources. You will be able to view part/package resources.

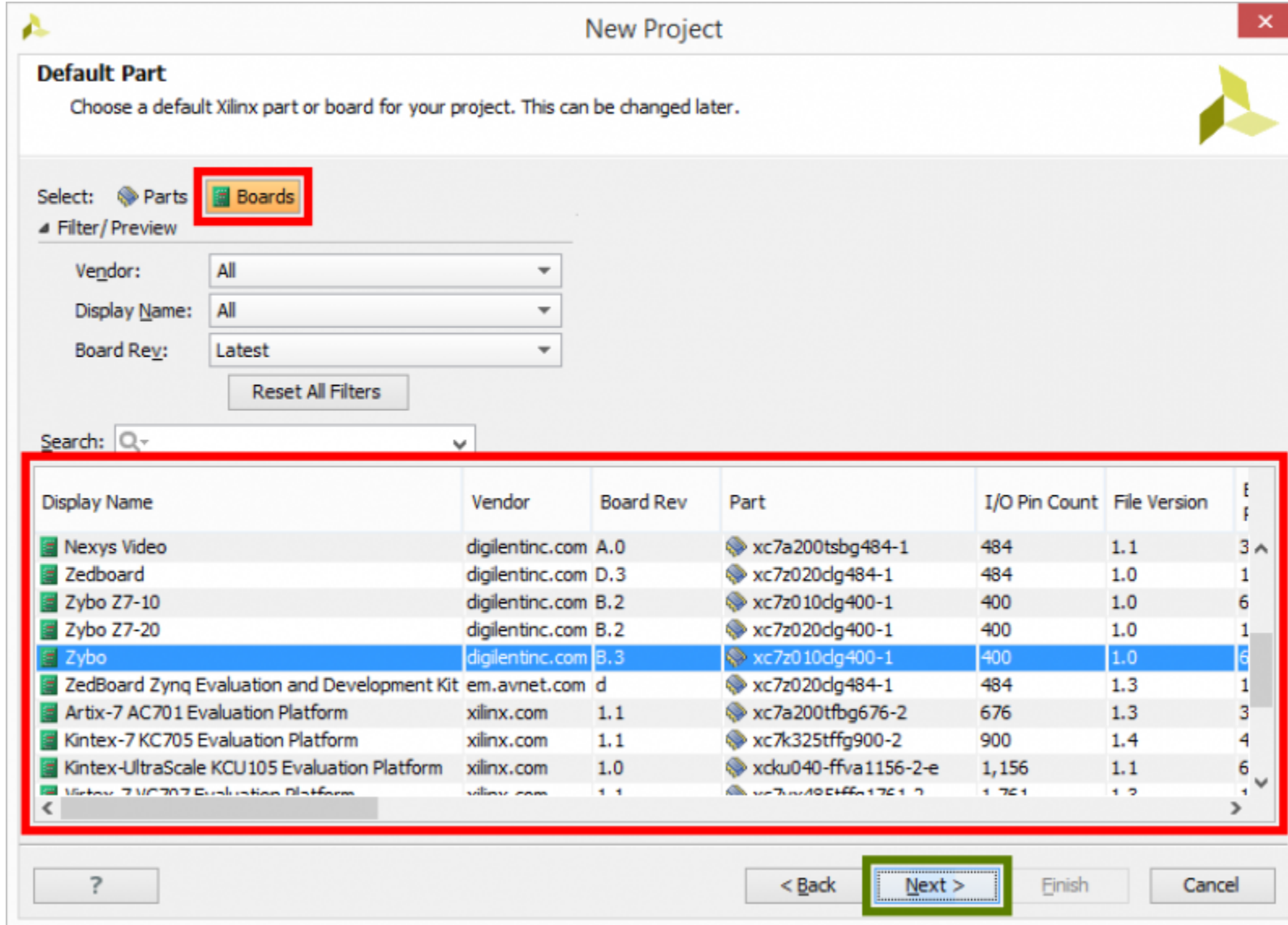
☐ **Imported Project**
Create a Vivado project from a Synplify, XST or ISE Project File.

☐ **Example Project**
Create a new Vivado project from a predefined template.

? < Back **Next >** Finish Cancel

1.5 이 화면은 프로젝트가 Target 으로 삼는 부분을 선택한다.

Digilent 의 보드 파일이 아직 설치되지 않은 경우 이 가이드의 prerequisites 섹션으로 돌아가서 해당 튜토리얼에 대한 링크를 따라 간 이후 Vivado 를 다시 시작하여 파일을 올바르게 로드해야 한다. Boards(보드) 탭을 선택한 다음 목록에서 Target Board 를 찾아 선택한다. Next 를 클릭한다.



New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: ☐ Parts ☒ **Boards**

Filter / Preview

Vendor:

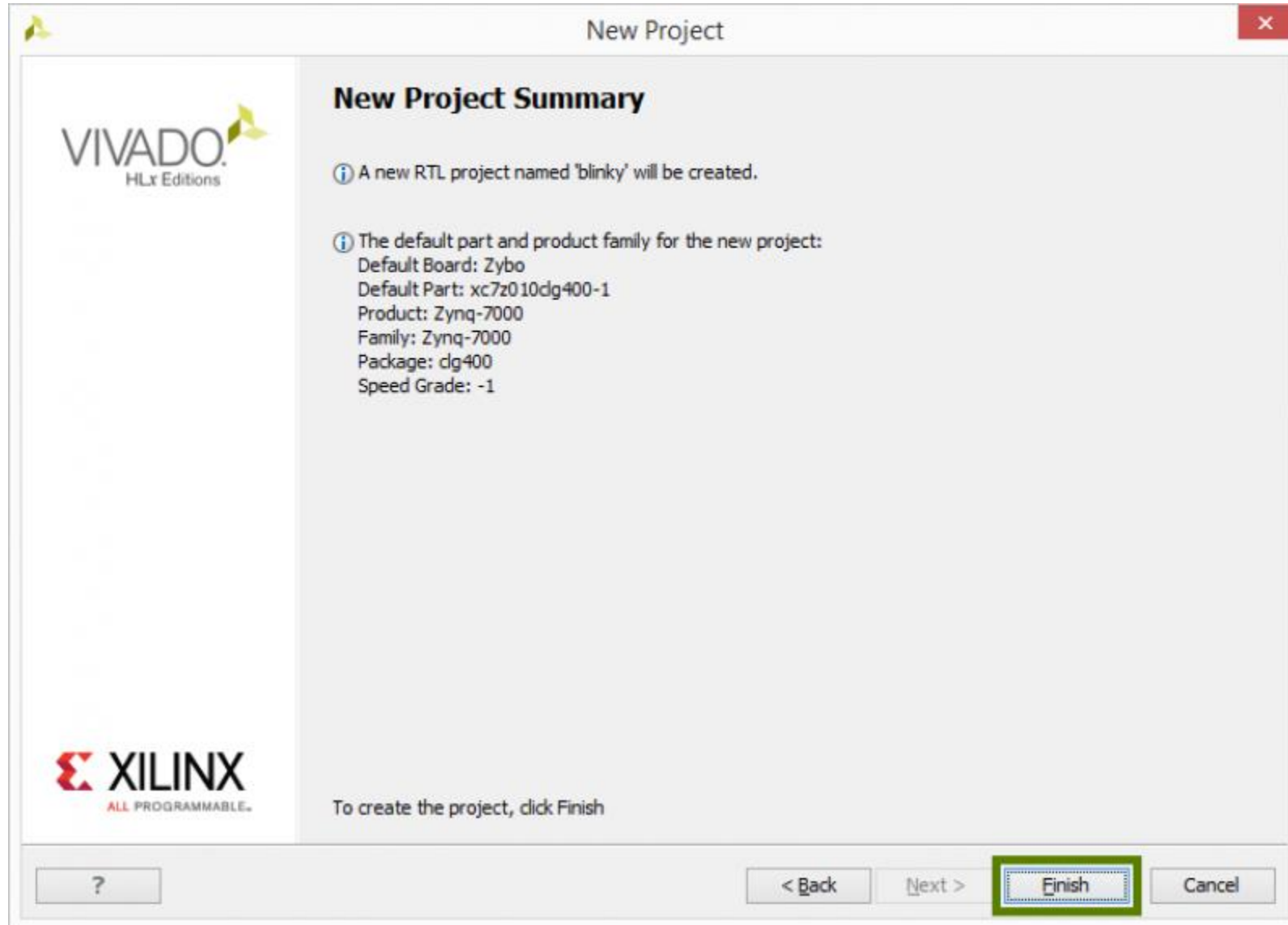
Display Name:

Board Rev:

Search:

Display Name	Vendor	Board Rev	Part	I/O Pin Count	File Version	File
Nexys Video	digilentinc.com	A.0	xc7a200tsbg484-1	484	1.1	3 ^
Zedboard	digilentinc.com	D.3	xc7z020dg484-1	484	1.0	1
Zybo Z7-10	digilentinc.com	B.2	xc7z010dg400-1	400	1.0	6
Zybo Z7-20	digilentinc.com	B.2	xc7z020dg400-1	400	1.0	1
Zybo	digilentinc.com	B.3	xc7z010dg400-1	400	1.0	6
ZedBoard Zynq Evaluation and Development Kit	em.avnet.com	d	xc7z020dg484-1	484	1.3	1
Artix-7 AC701 Evaluation Platform	xilinx.com	1.1	xc7a200tfbg676-2	676	1.3	3
Kintex-7 KC705 Evaluation Platform	xilinx.com	1.1	xc7k325tffg900-2	900	1.4	4
Kintex-UltraScale KCU105 Evaluation Platform	xilinx.com	1.0	xdku040-ffva1156-2-e	1,156	1.1	6
Virtex-7 VC707 Evaluation Platform	xilinx.com	1.1	xc7v485tffg1761-2	1,761	1.2	1

- 1.6 이 화면은 이전 화면에서 선택한 선택 사항을 요약한다.
Next 를 클릭하여 New Project 열기를 마무리한다.

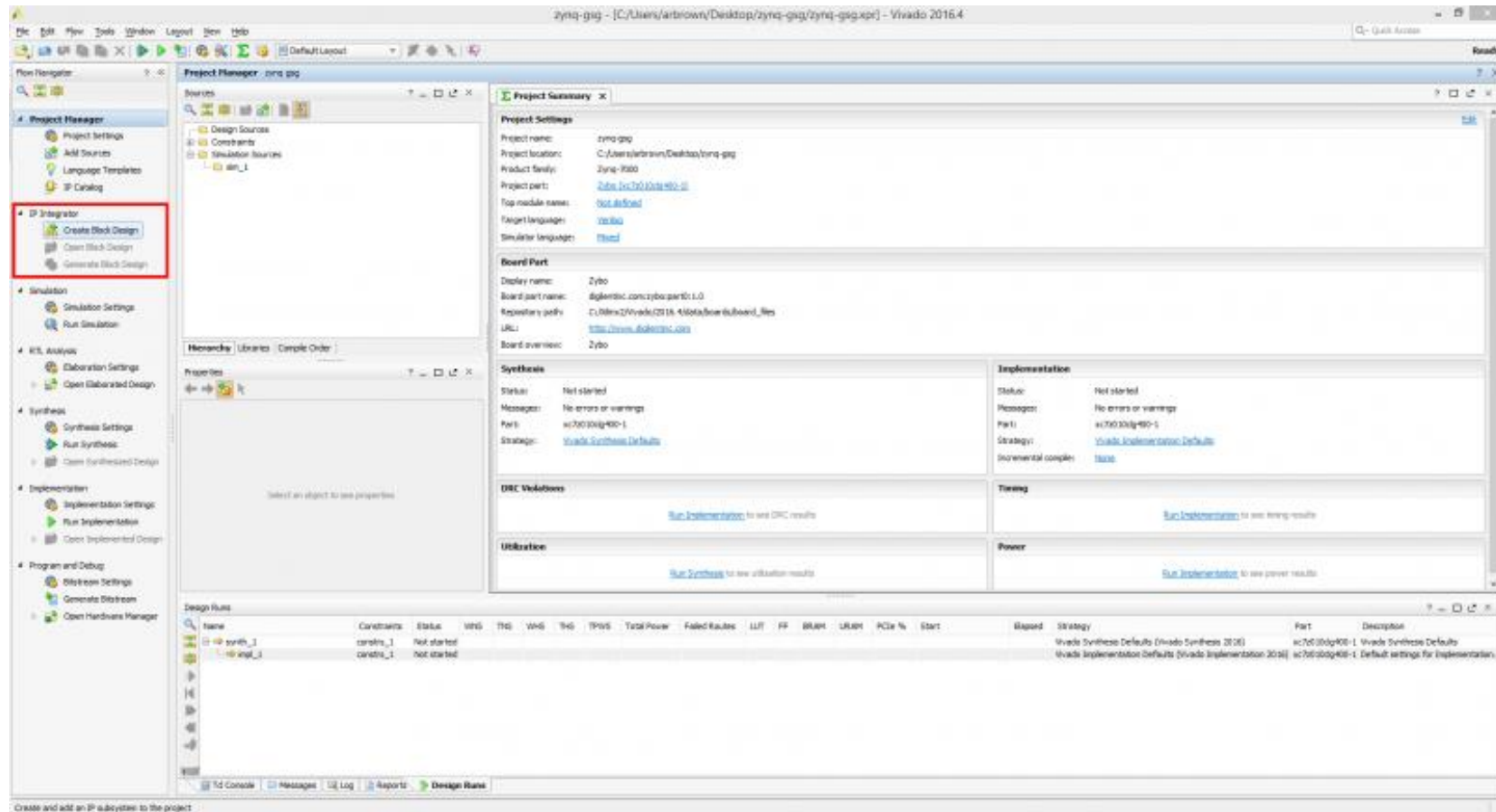


2. Open the IP Integrator

2.1 표준 Vivado 작업 흐름에 대한 자세한 내용은 Digilent 의 Getting Started with Vivado 튜토리얼에서 확인할 수 있다.

https://reference.digilentinc.com/vivado/getting_started/start

이 가이드에서는 Windows 의 오른쪽에 있는 Flow Navigator 에서 열 수 있는 IP Integrator 도구를 독점적으로 사용한다.
IP Integrator 탭을 확장하고 Create Block Design 을 선택한다.



2.2 Dialog Box 에서 Block Design 이름을 지정한다.

디렉토리 위치는 Block Design 이 저장되는 위치이며 변경할 수 있지만 <Local to Project> 로 두는 것이 좋다.
Specify source set 이 Design Sources 로 설정되어 있는지 확인한다.

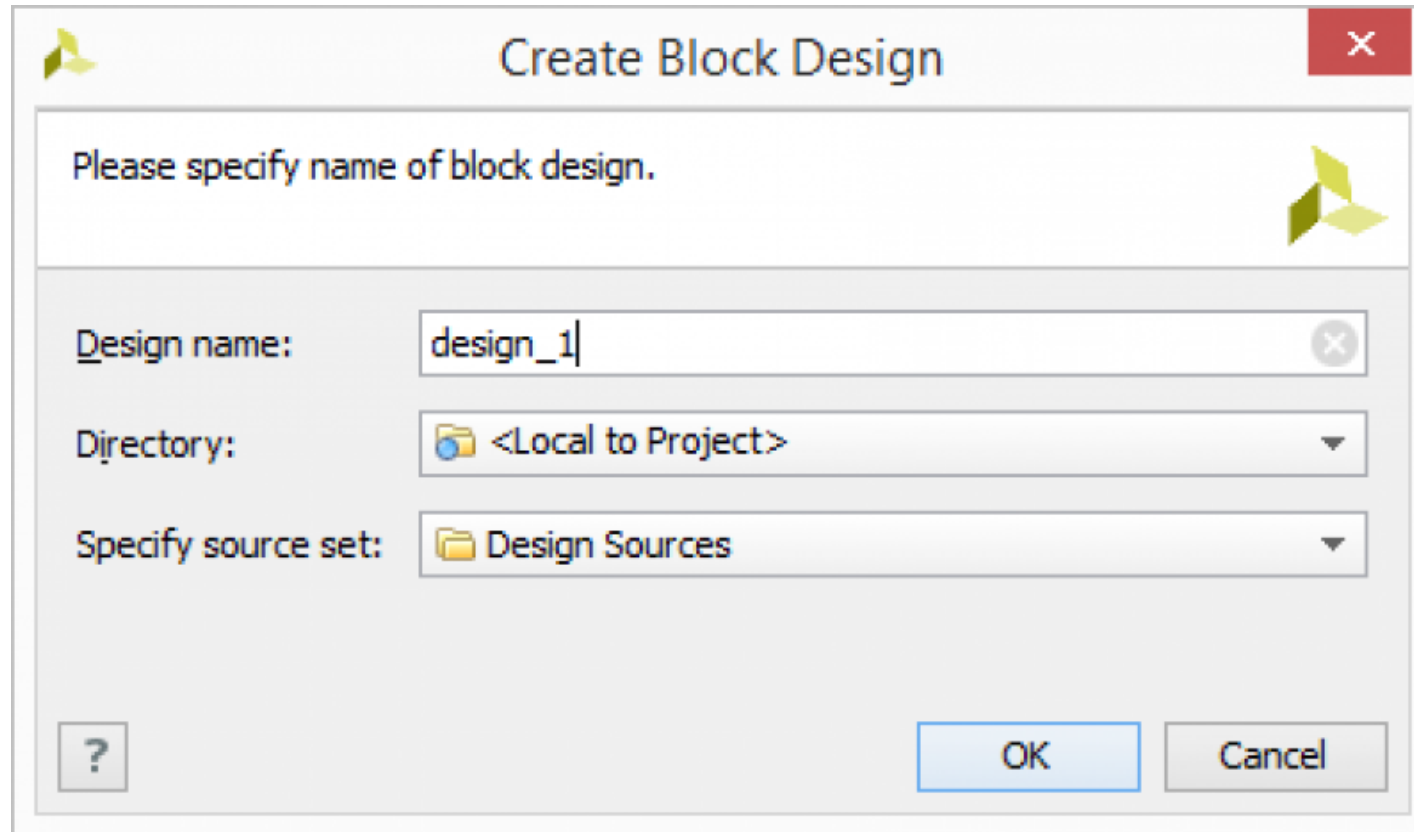
* Important

Block Design 이름이나 디렉토리 경로에 공백을 사용하지 말라.

이것은 Vivado 에 문제를 일으킬 것이다.

대신 밑줄, 대시 또는 CamelCase 를 사용하라.

OK 를 클릭하여 Block Design 을 작성하고 IPI 를 연다.



The image shows a 'Create Block Design' dialog box from the Vivado IDE. The title bar is light gray with the Vivado logo on the left and a red close button on the right. The main area has a light gray background. At the top, there is a text prompt 'Please specify name of block design.' followed by a small Vivado logo. Below this, there are three input fields: 'Design name:' with a text box containing 'design_1', 'Directory:' with a dropdown menu showing '<Local to Project>', and 'Specify source set:' with a dropdown menu showing 'Design Sources'. At the bottom, there is a help button (a square with a question mark), an 'OK' button, and a 'Cancel' button.

Create Block Design

Please specify name of block design.

Design name: design_1

Directory: <Local to Project>

Specify source set: Design Sources

? OK Cancel

3. IP Integrator Tools

이 가이드에서 설명하는 프로젝트는 크게 FPGA 와 프로세서로 구현되는 두 가지 주요 세그먼트로 구성된다.
타겟 보드에 따라 프로세서는 별개의 HW 구성 요소가 아닌 FPGA Fabric 내에서 구현 될 수 있지만 대부분의 경우 관련이 없다.

FPGA 와 프로세서에 구현 된 주변 장치간에 데이터를 이동하는데 사용되는 통신 프로토콜을 AXI(Advanced eXtensible Interface)라고 한다.
FPGA 상에 구현 된 각 AXI 주변 장치에는 Zynq 의 메모리 공간 영역이 할당되며 이 영역은 각 제어 레지스터를 주소 지정하는데 사용된다.
Design 에 간단한 AXI 인터페이스를 추가하려면 AXI 신호가 라우팅되는 방법을 정의하는
AXI 인터커넥트와 각 AXI 블록 및 인터페이스의 리셋을 생성하는 리셋 컨트롤러라는 두 개의 새로운 블록을 추가해야 한다.

대부분의 기본 블록 설계는 서로 다른 AXI 주변 장치를 프로세서에 연결하고 이를 사용하여 입력 및 출력 포트에서 읽고 쓰는 것으로 구성된다.

이러한 맥락에서 벗어나 Vivado IPI 에서 디자인을 생성하는데 사용할 수 있는 많은 도구에 대해 설명한다.

3.1 다음 페이지의 Image 에서 주황색으로 강조 표시된 Sources 탭에는 몇 개의 하위 탭이 있으며 이들 중 Hierarchy 및 IP Sources 가 가장 유용하다.

Hierarchy 하위 탭에는 프로젝트에 있는 소스 집합이 표시된다.

이들은 3 개의 그룹으로 나뉘며 Design Sources 는 Block Design 과 그 아래 Block Design 에 포함 된 모든 IP 코어 또는 다른 파일의 소스를 포함한다.
제약 조건 하위 탭에 프로젝트에 추가 된 XDC(Xilinx Design Constraint) 파일이 포함되어 있다.
XDC 파일은 보드 파일에서 생략 된 포트를 제한하는데 사용할 수 있다.

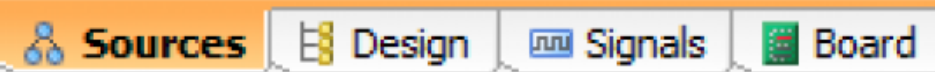
IP Sources 하위 탭에는 새로운 IP 코어가 블록 설계에 추가되었을 때 생성 된 파일이 표시된다.
이 화면에서 Block Design 을 마우스 우클릭하고 Reset Output Products 를 선택한 다음
Generate Output Products 를 선택하면 몇 가지 성가신 오류를 해결할 수 있다.

Sources

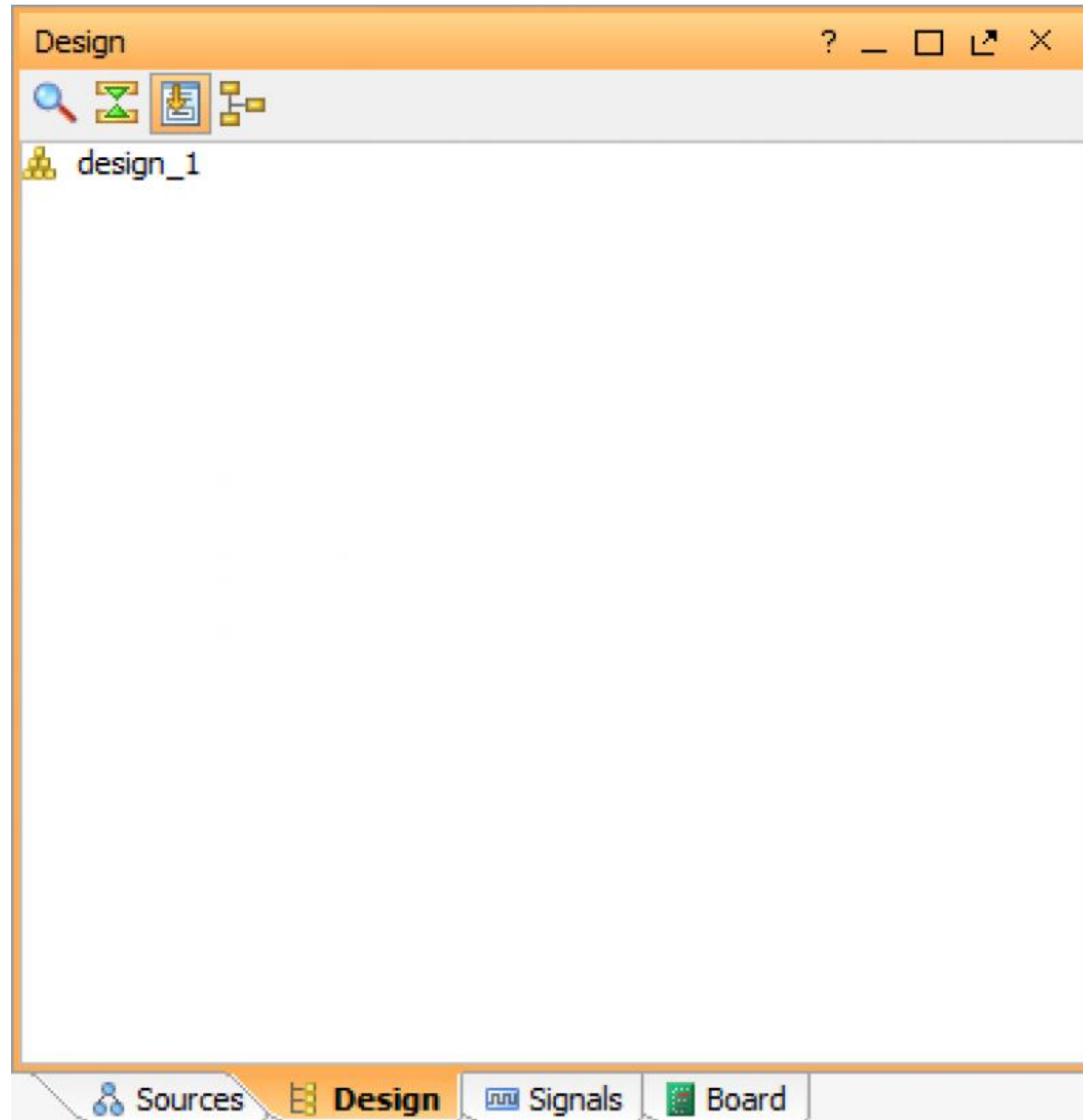


- Design Sources (1)
 - design_1 (design_1.bd)
- + Constraints
- Simulation Sources (1)
 - + sim_1 (1)

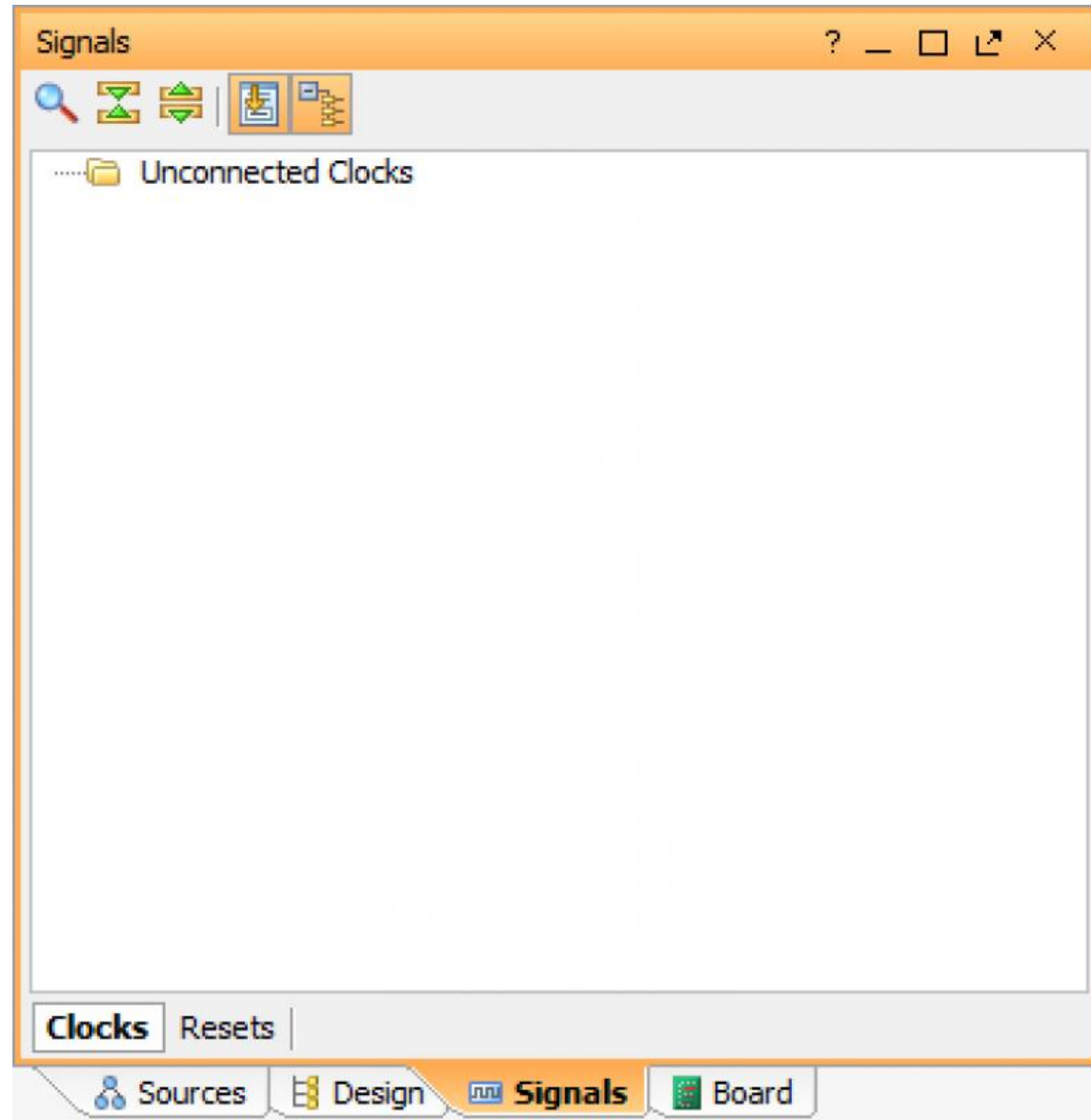
Hierarchy | IP Sources | Libraries | Compile Order



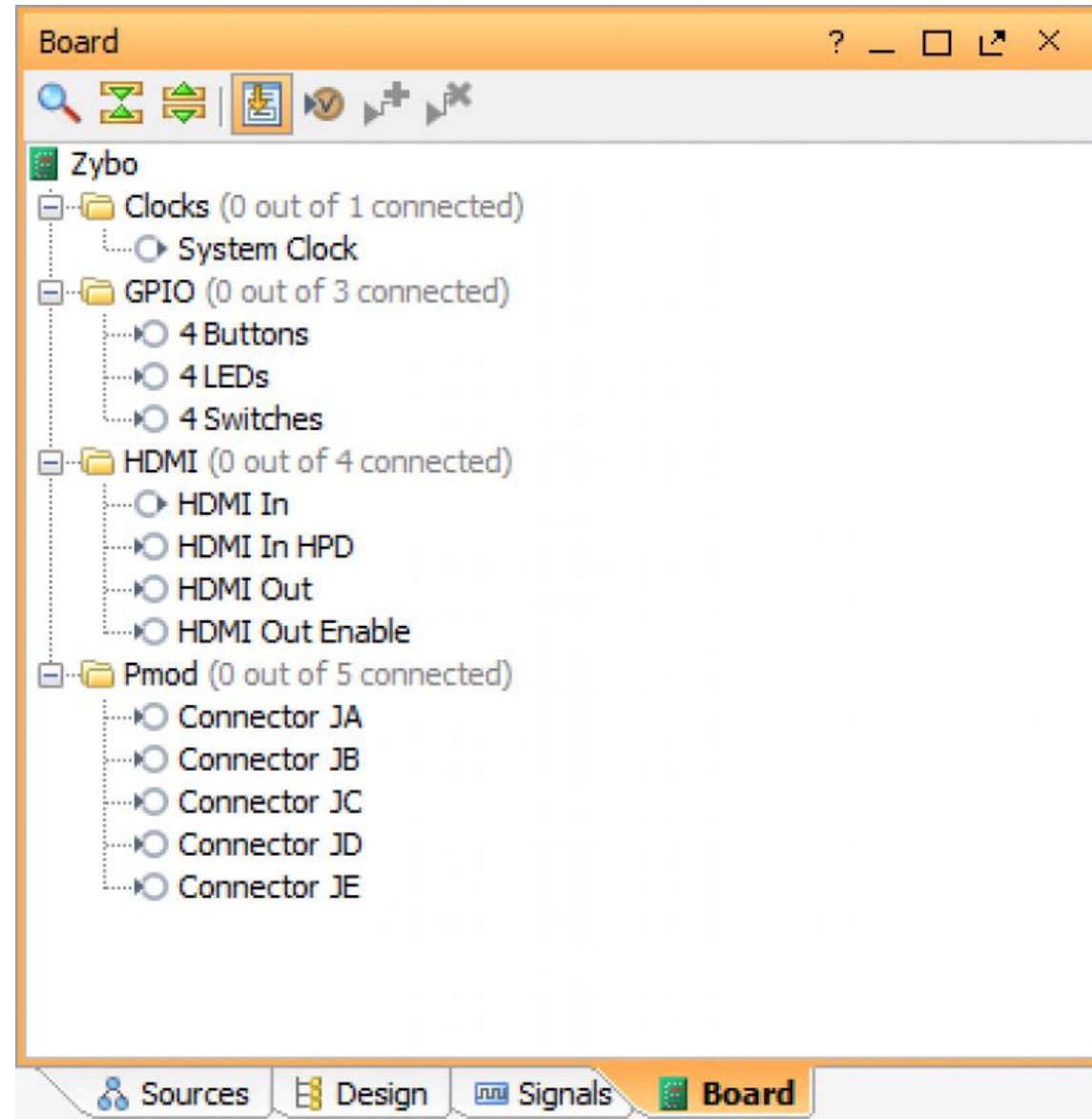
- 3.2 Design 탭을 선택하면 모든 입출력 포트, IP 코어 포트 및 IP 코어 간의 연결 목록이 표시된다.
이 목록에서 항목을 선택하면 Block Design Diagram 에서 해당 Object 가 강조 표시된다.



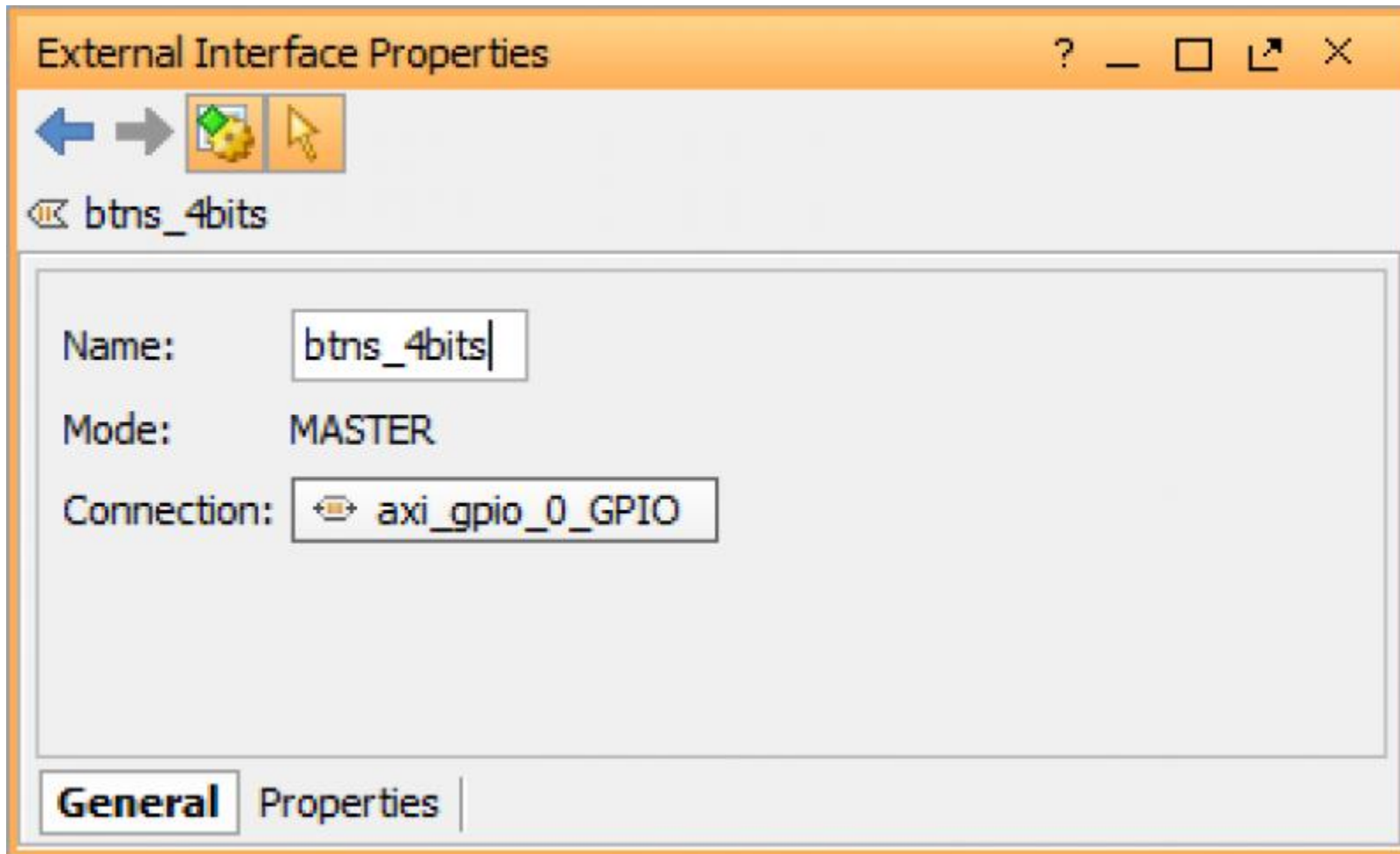
- 3.3 Signals 탭을 사용하면 Block Design 에서 모든 Clock 및 Reset 신호 목록을 볼 수 있다.
이전과 마찬가지로 이 목록 중 하나에서 항목을 선택하면 Block Design Diagram 에서 해당 신호가 강조 표시된다.
이 도구는 데이터가 복잡한 Design 에서 Clock Domain 을 통과하는 위치를 파악하는데 유용하다.



- 3.4 Board 탭은 Digilent 가 프로젝트 생성 과정에서 선택한 보드 파일의 일부로서 제공한 모든 외부 연결을 표시한다.
이 도구를 사용하면 이러한 포트를 Design 에 매우 쉽게 연결할 수 있다.
이 포트 중 하나를 연결하는 프로세스는 이 가이드의 뒷부분에서 설명한다.



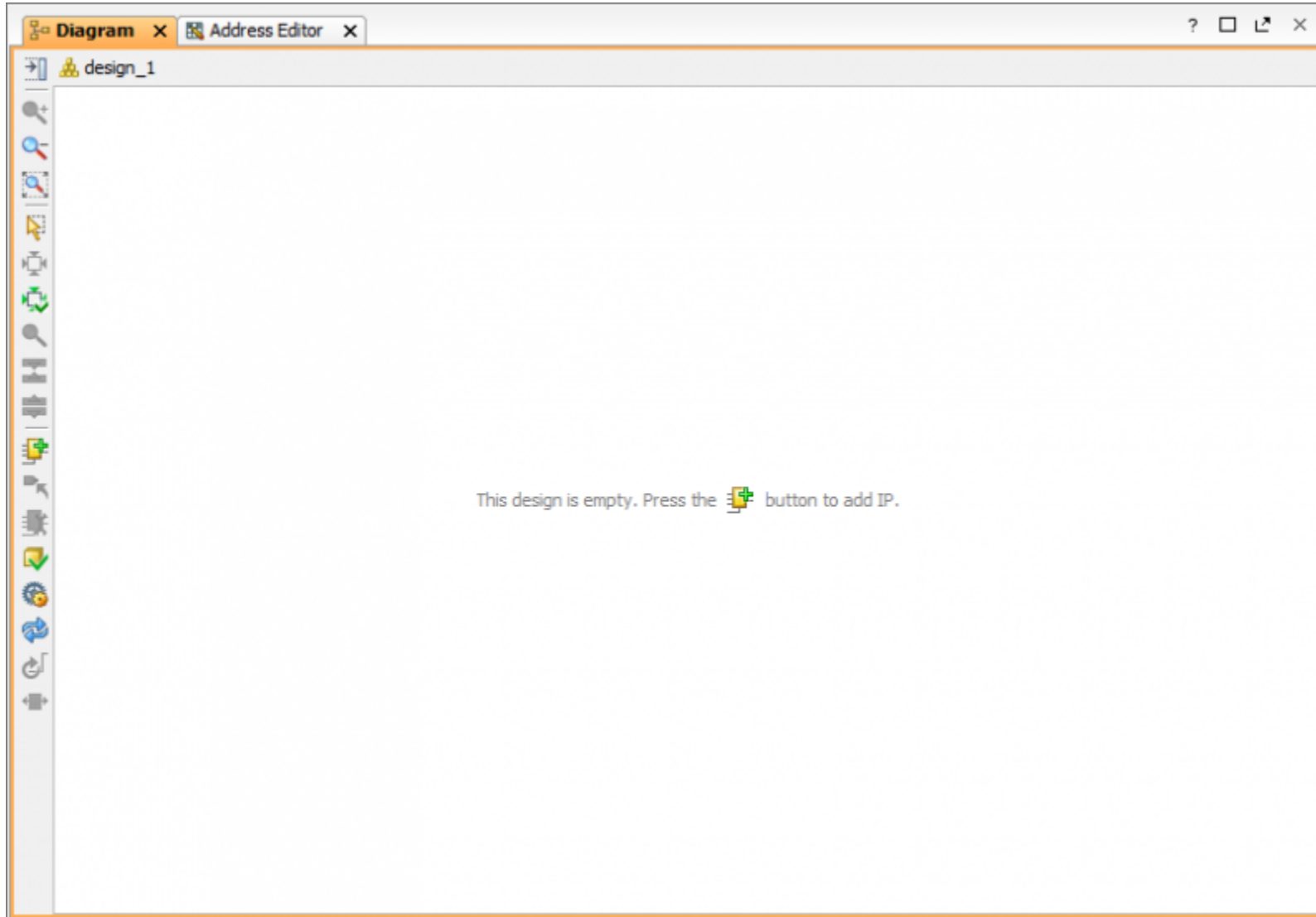
- 3.5 위에 설명된 Tabbed 패널 아래에 있는 Properties 창은 현재 선택된 Object 이 특성을 표시한다.
이 창은 일반적으로 선택한 클럭 핀의 클럭 주파수를 빠르게 보거나 IP 코어 또는 포트의 이름을 변경하는데 사용된다.



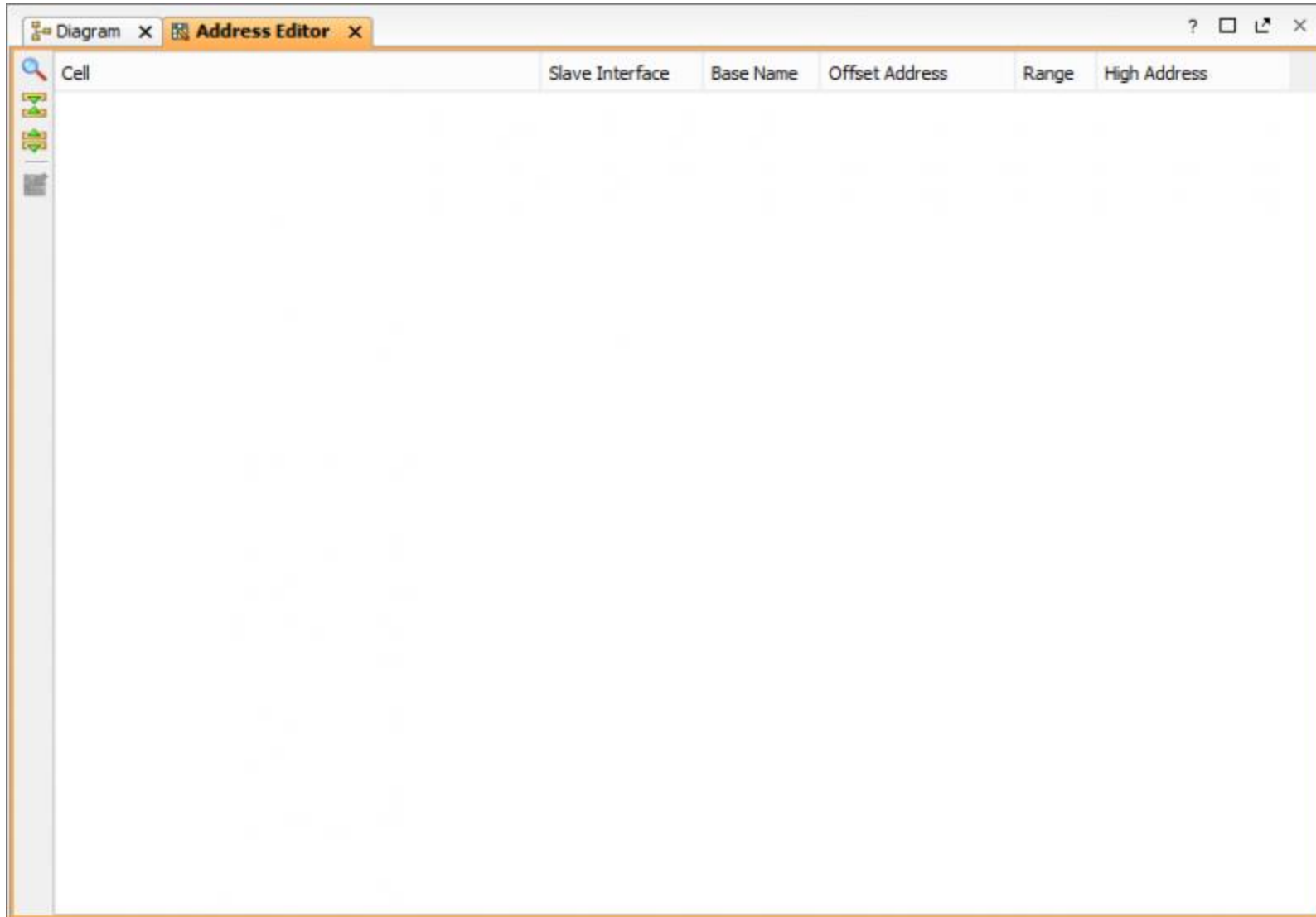
3.6 Vivado 창의 오른쪽 상단의 대부분은 Diagram 창에 있다.

이 창에는 현재 Block Design 의 그래픽 렌더링이 표시된다.

이 창의 왼쪽에 있는 도구 모음에 있는 많은 단추에 대해서는 이 가이드의 4 단계에서 설명한다.

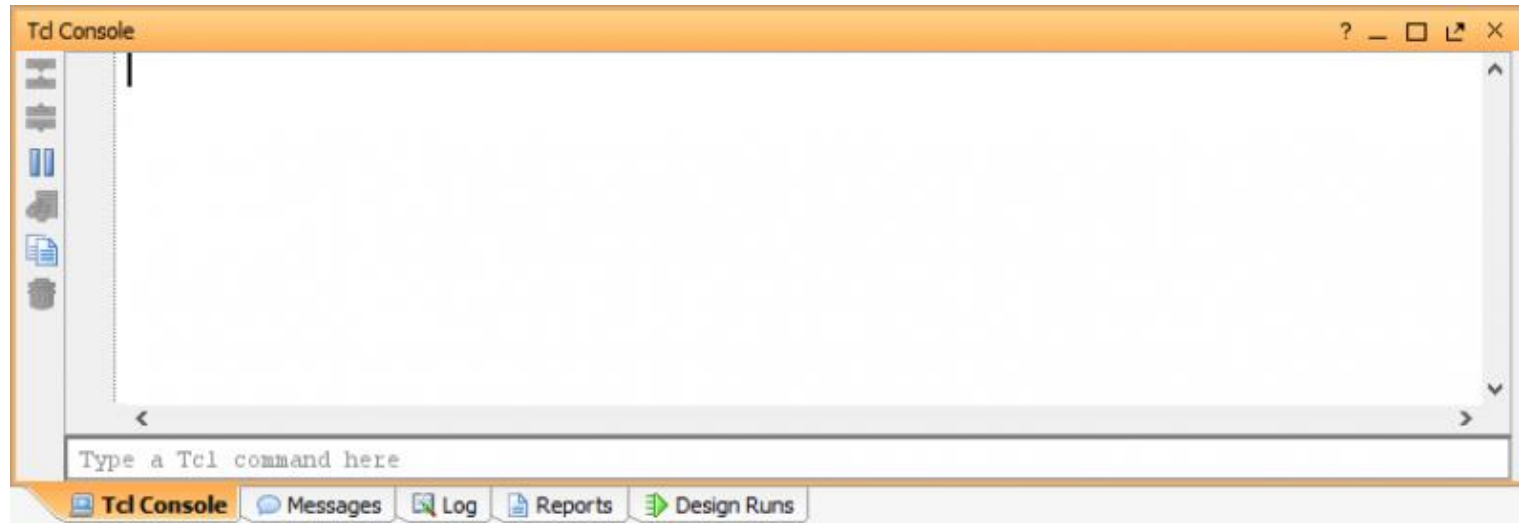


- 3.7 Diagram 과 동일한 창에 포함된 Address Editor(주소 편집기) 탭은 프로세서가 설치된 각 AXI 주변 장치를 찾을 수 있는 메모리 주소를 기술한다.
고급 사용자를 제외하고 이 창에서 값을 변경하는 것은 좋지 않다.
매핑되지 않은 주변 장치를 참조하는 블록 설계 유효성 검사 프로세스의 오류는
이 창의 아무 곳이나 마우스 우클릭이후 Auto-Assign Address 를 선택하여 해결할 수 있다.



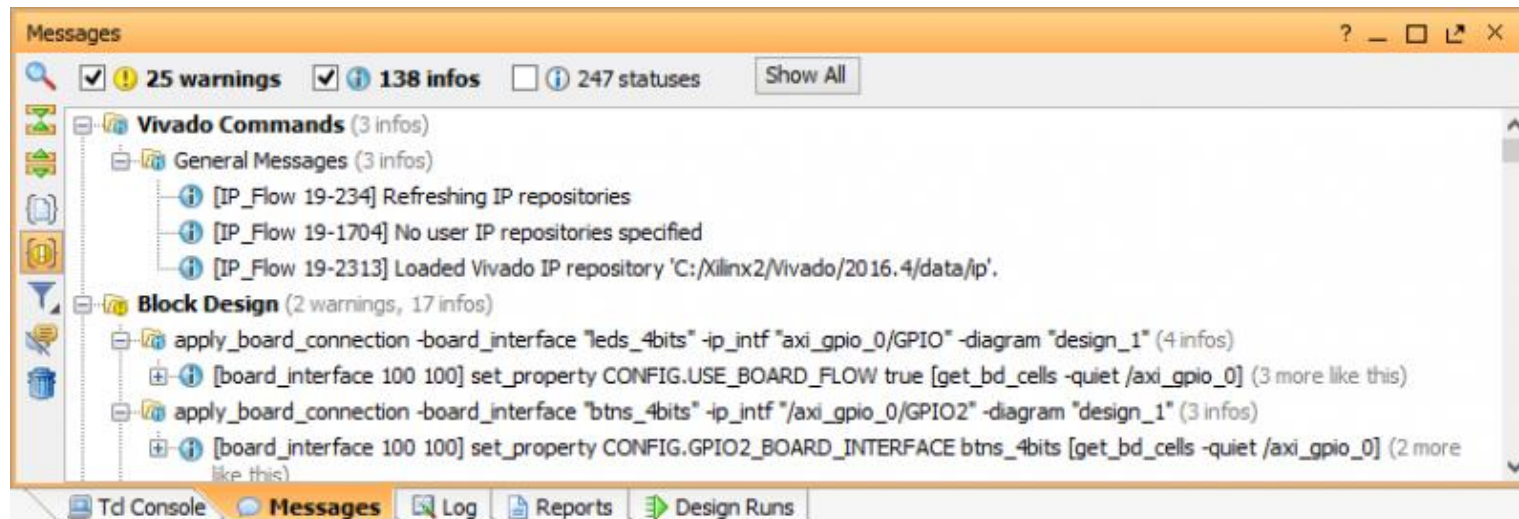
3.8 창의 맨 아래에 있는 첫 번째 탭은 TCL Console 이다.

이 도구는 그래픽 인터페이스에서 변경 될 때마다 Vivado 가 실행중인 스크립트 명령을 표시한다.
스크립트는 'source' 명령을 사용하여 실행되도록 만들 수 있다.

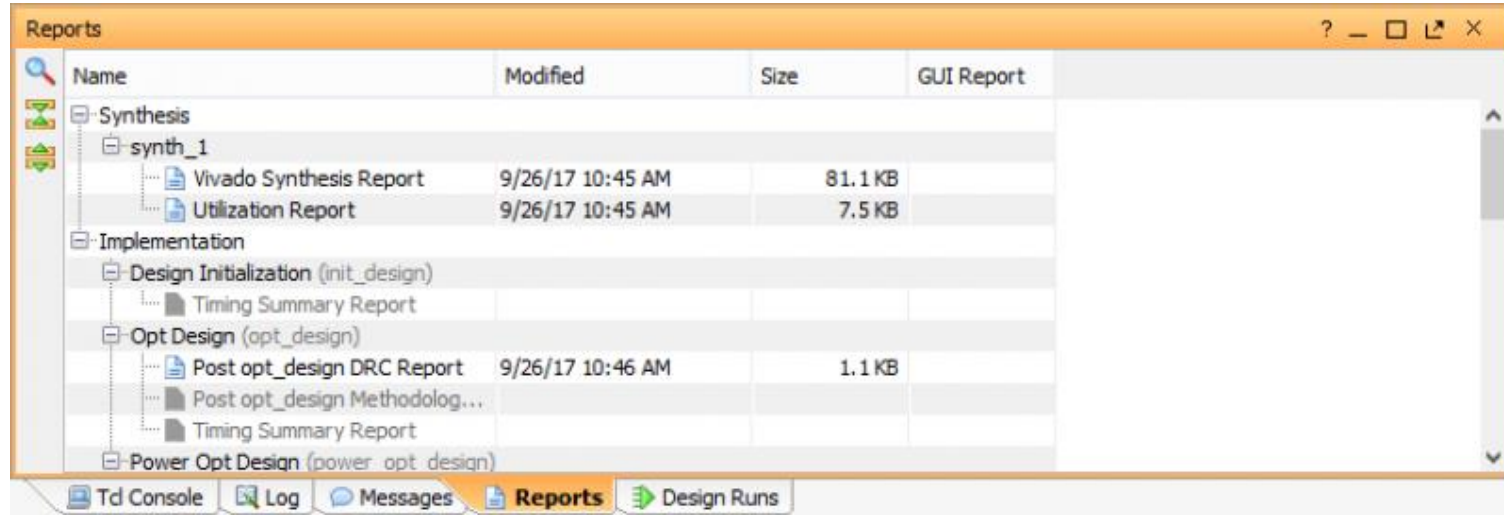


3.9 Messages 탭에는 Vivado 가 다른 작업을 수행 할 때 생성된 오류, 경고, 정보 및 상태 메시지가 표시된다.

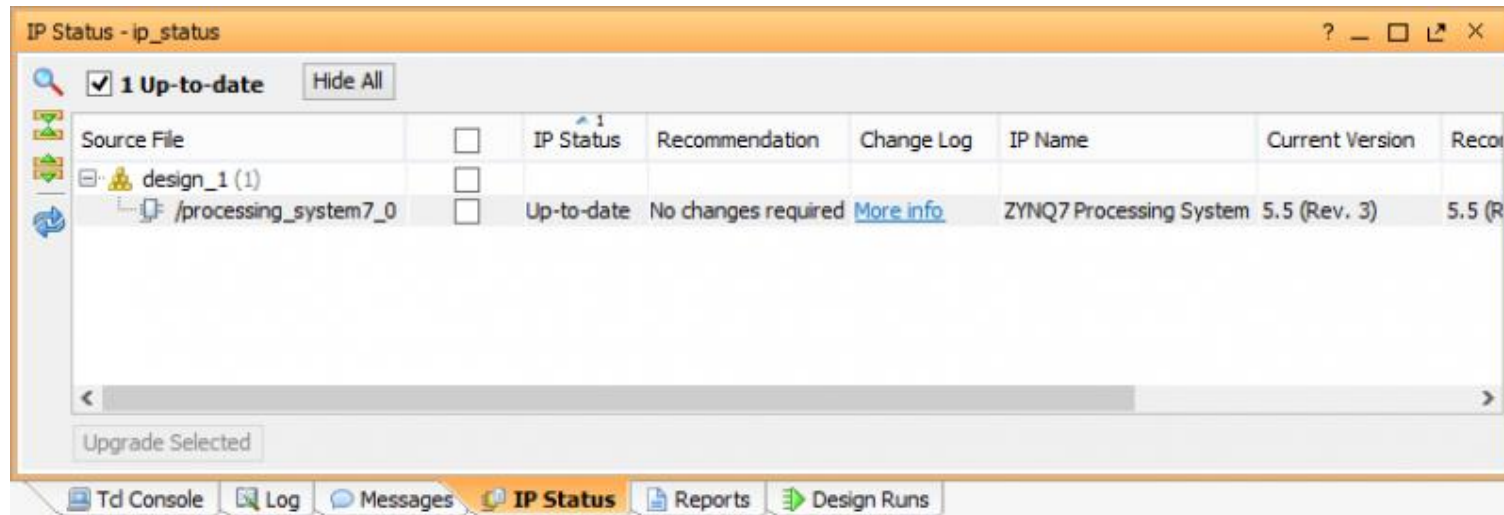
이 메시지는 TCL Console 에서도 찾을 수 있지만 비교적 쉽게 탐색 할 수 있는 형식으로 제공된다.



3.10 Reports 탭에는 bitstream 생성 프로세스의 일부로 Vivado 에서 생성하는 여러 보고서 목록이 있다.



3.11 Vivado 창의 맨 위에 있는 도구 모음에서 Tools -> Reports -> Report IP Status 를 선택하면 IP 가 Design 에 추가 될 때까지 다른 탭이 맨 아래 창에 추가된다.
이 IP Status 탭에는 프로젝트에 추가 된 각 IP 코어의 버전과 Target Devices 가 표시된다.
IP 가 최신이 아닌 경우 Upgrade All 버튼을 클릭하면 최신 버전의 IP 가 다시 로드 되거나
사용중인 Vivado 버전에서 IP 작업을 수행하는데 필요한 업데이트가 수행된다.



3.12 Diagram 창의 왼쪽에 있는 툴바에는 모두 다른 작업을 수행하는 많은 수의 단추가 있다.

- * Block Design Options 를 사용하면 Diagram 창에 표시되는 내용과 방법을 사용자가 변경할 수 있다.
- * Zoom In 은 현재 중심 위치를 유지하면서 뷰를 확대한다.
- * Zoom Out 은 현재 중심 위치를 유지하면서 뷰를 축소한다.
- * Zoom Fit 은 현재 Design 의 모든 Block 이 화면에 표시되도록 한다.
- * Select Area 는 그려진 사각형의 모든 Block 을 선택한다.
Block Group 을 한 번에 이동하는데 유용하다.
- * Fit Selection 은 뷰를 현재 선택된 블록의 중심으로 확대/축소한다.
- * Search 는 현재 Design 의 Ports 와 Blocks 에서 검색을 시작한다.
- * Add IP 는 Search Dialog 를 띄워 새로운 IP Block 을 Design 에 추가한다.
나중에 이것에 대한 자세한 내용을 다룬다.
- * Make External 은 현재 선택된 핀에 대한 입력 또는 출력 포트를 만든다.
이것은 Digilent 보드 파일이 아닌 XDC 파일로 포트를 제한 할 때 유용하다.
- * Customize Block 은 현재 선택된 Block 에 대한 Customization Dialog 를 시작한다.
나중에 이것에 대해서도 자세한 내용을 다룬다.
- * Validate Design 은 bitstream 을 생성하기 전에 Vivado 가 Block Design 에서 실수를 찾도록 지시한다.
나중에 이것에 대한 자세한 내용을 다룬다.
- * IP Settings 는 IP setting 페이지에서 Project Settings 를 연다.
비표준 IP 저장소를 포함할 때 매우 유용하다.
- * Regenerate Layout 은 Vivado 가 IP Block 을 이동하고 신호를 Rerouting 하여 Block Design 을 시각적으로 정리하도록 지시한다.
- * Optimize Routing 은 Vivado 에게 신호 경로 변경만으로 시각적으로 Block Design 을 정리하도록 지시한다.
- * Show Interface Connections Only 는 시계와 리셋 신호를 숨기며 매우 복잡한 Design 에만 유용하다.

또한 Diagram 창의 그래픽 인터페이스와 직접 상호 작용하여 Design 을 변경할 수 있다.

개체를 클릭하고 드래그하여 개체를 이동할 수 있다.

연필 커서가 나타날 때까지 포트를 가리키면 다른 포트, 핀 또는 와이어에 선을 그릴 수 있다.

IPI 에서 취할 수 있는 조치에 대한 일반적인 이해를 바탕으로 Design 을 만들 수 있다.

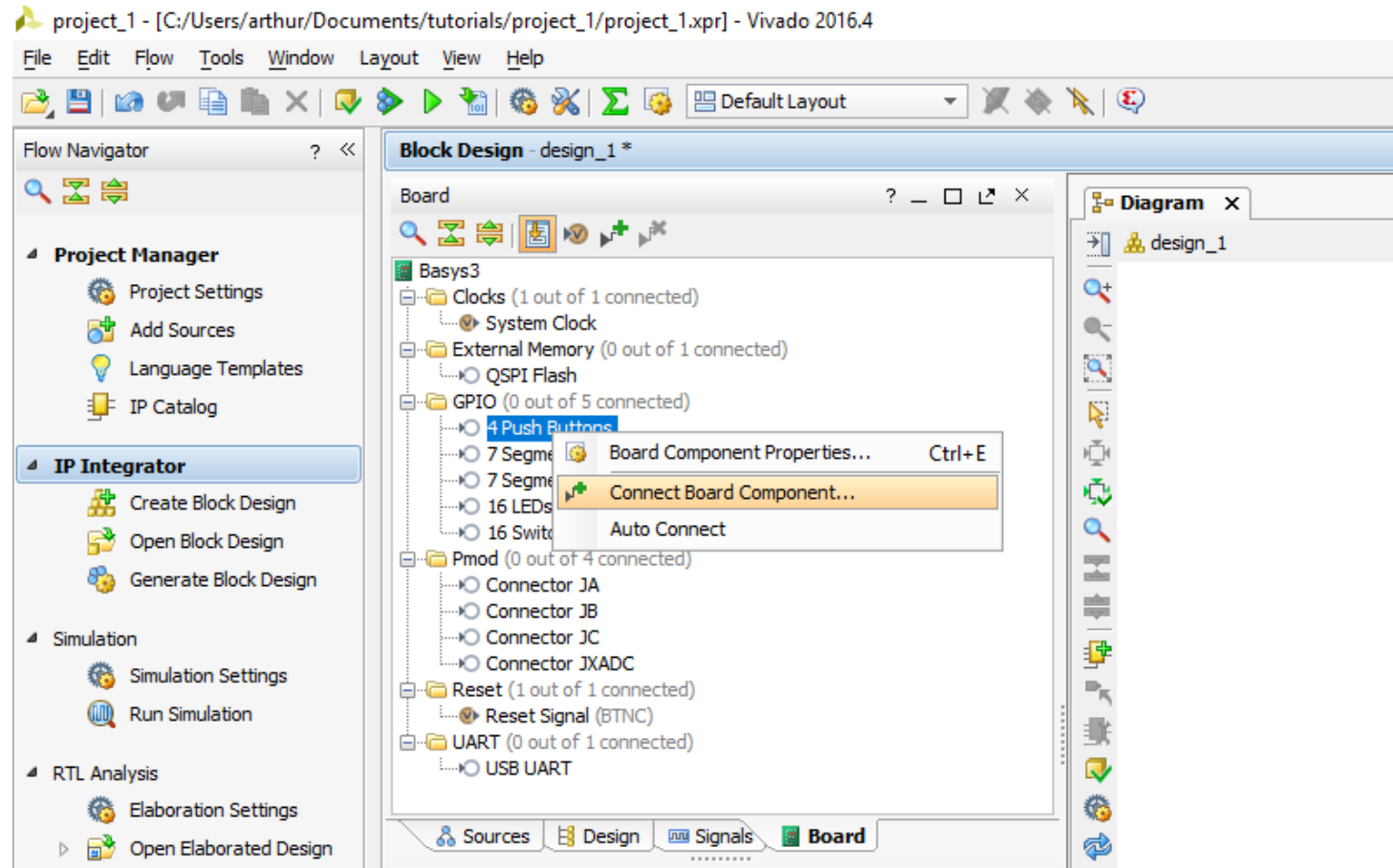
4. Create a Simple Block Design

4.1 Board 탭에서 Push Buttons 를 마우스 우클릭 한 다음 connect component 를 선택한다.
팝업창이 뜨면 Create new IP -> AXI GPIO 에서 GPIO 를 선택한다.

* Important

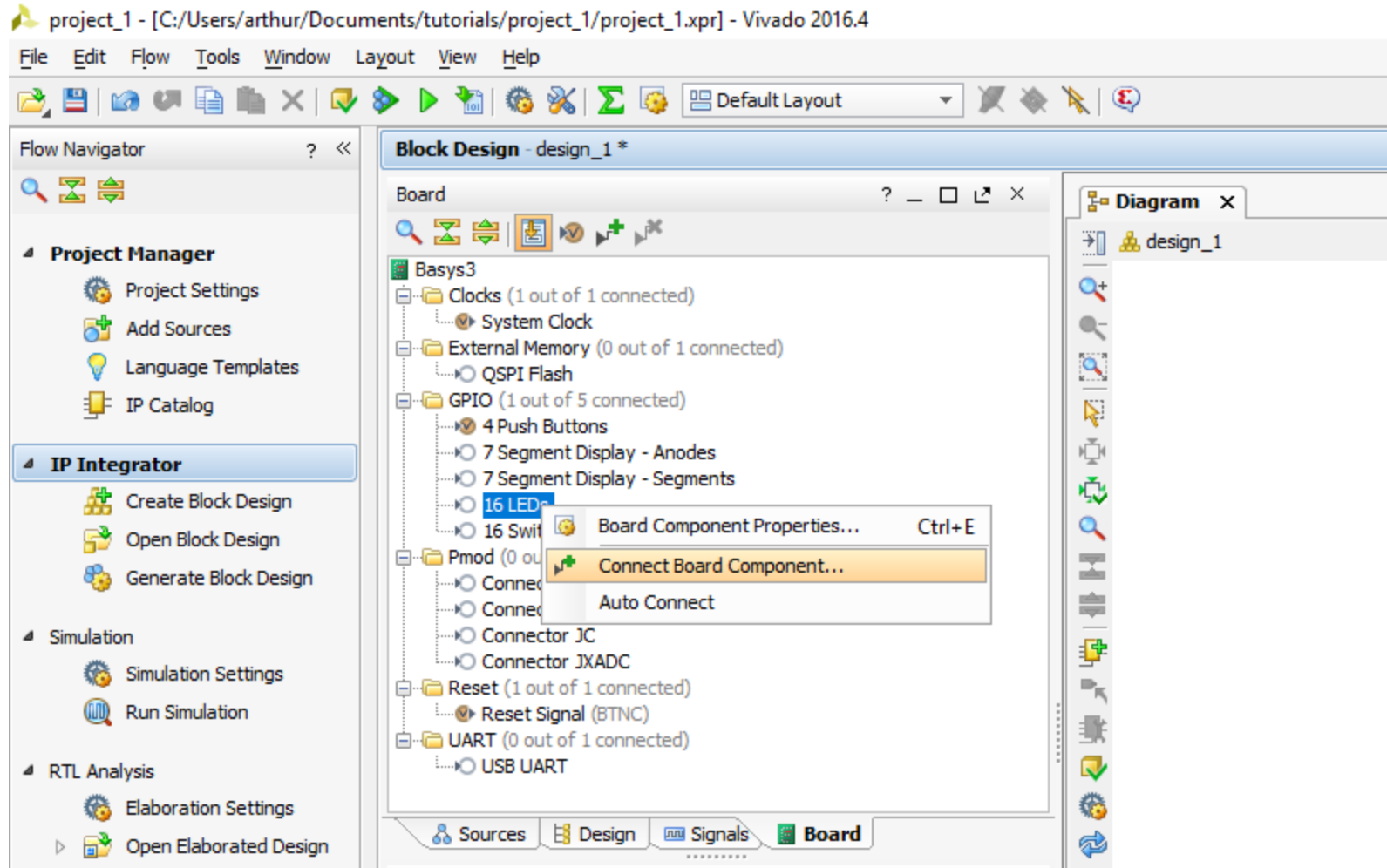
버튼 구성 요소는 GPIO2 가 아닌 GPIO 에 연결해야 한다.
나중에 제공된 C 소스 코드가 이 경우라고 가정한다.

계속하려면 OK 를 클릭한다.



4.2 Board 탭에서 LED 를 마우스 우클릭 한 다음에 connect component 를 선택한다.
팝업창이 뜨면 Connect to existing IP -> axi_gpio_0 에서 GPIO2 를 선택한다.
계속하려면 OK 를 클릭한다.

Board 탭에 RGB LED 에 대한 항목이 있으면 해당 구성 요소를 새로운 AXI GPIO 컨트롤러에 연결하라.



이 섹션의 나머지 부분은 프로젝트가 설계되는 보드에 따라 분기된다.
Target Board 가 Zynq 칩을 사용하는 경우와 Microblaze 를 사용하는 경우가 다르다.

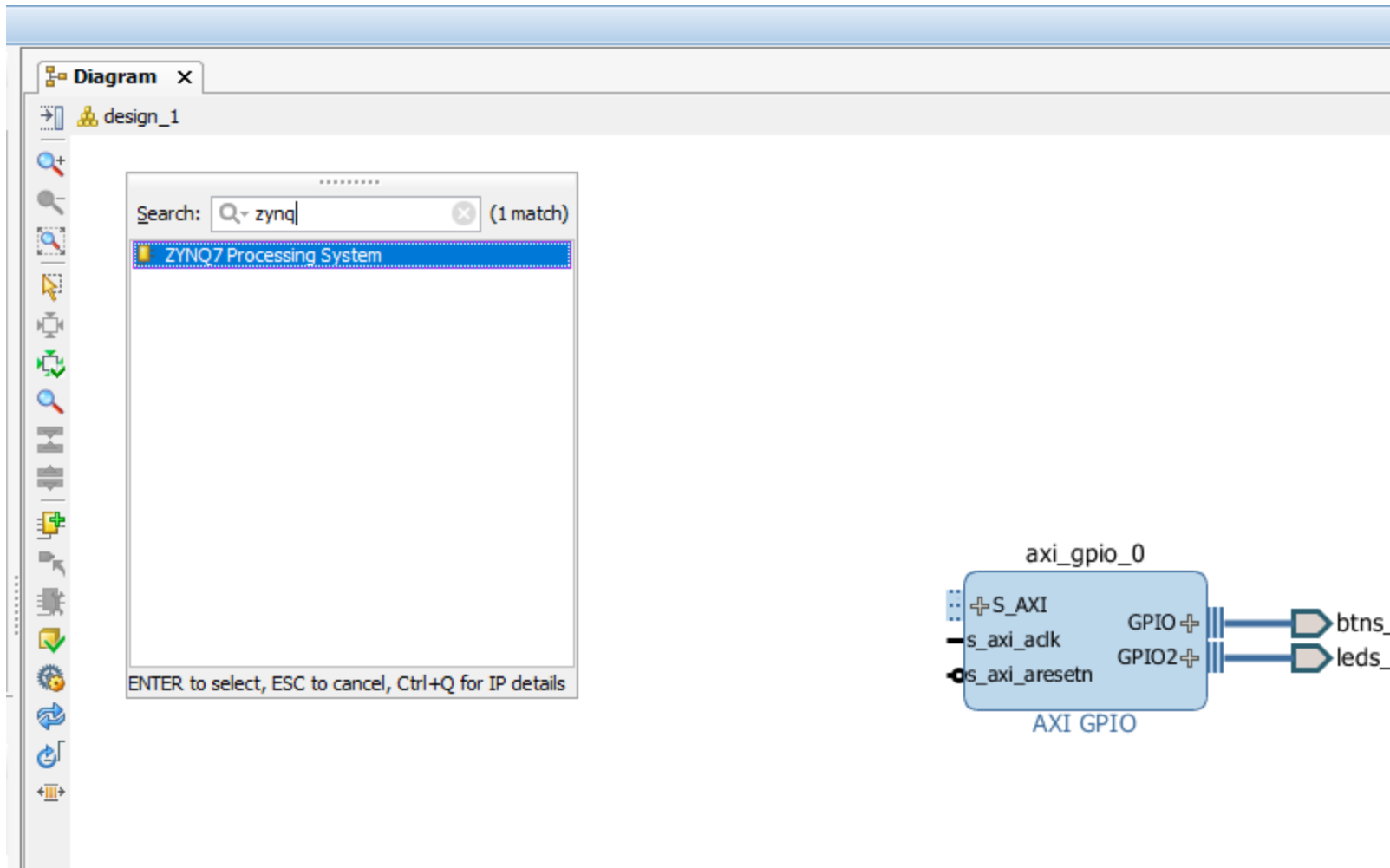
4.3 Add IP 버튼을 클릭하고 Zynq 를 검색한다.

결과 목록에서 Zynq7 Processing System 을 선택하고 키보드의 Enter 를 누른다.

이렇게 하면 Block Design 에 Zynq 프로세서가 추가된다.

Zynq 칩은 FPGA Fabric 과 HW 프로세서를 모두 포함한다.

이 Block 은 프로세서뿐만 아니라 FPGA 의 일부가 아닌 다른 HW 구성 요소를 나타낸다.



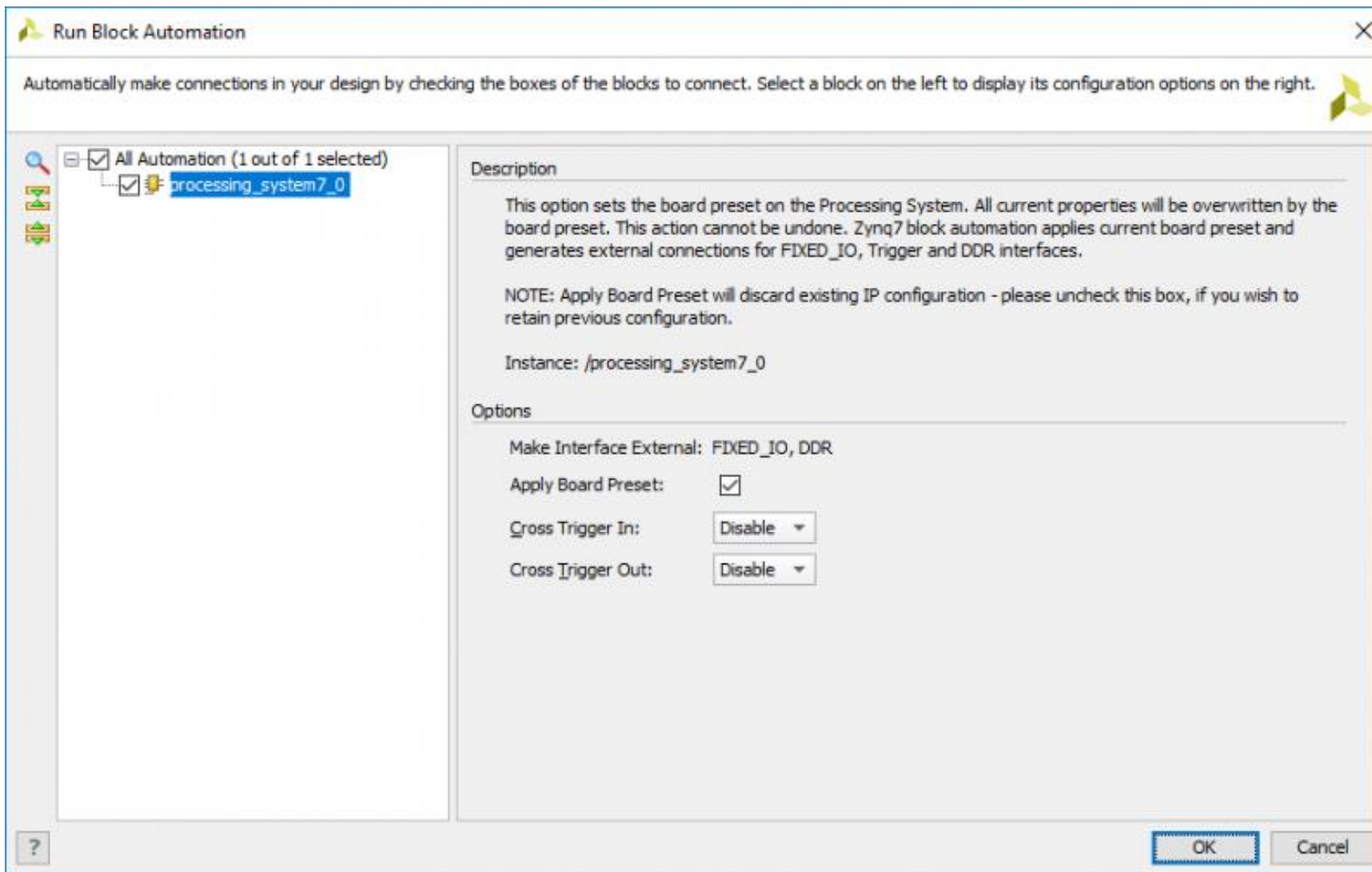
4.4 화면 상단의 녹색 막대에 Run Block Automation 을 클릭한다.
그러면 Zynq 블록의 초기 구성을 허용하는 대화 상자가 열린다.

Run Block Automation 대화 상자에서 왼쪽 패널의 All Automation 아래에 있는 FIXME 가 선택되고 체크되었는지 확인하라.

다양한 옵션을 사용할 수 있다.
이 가이드의 목적에 따라 옵션을 아래 괄호 안에 있는 값으로 설정한다.

- Make Interfaces External(구성 불가능)
이 필드는 Design 에서 자동으로 연결될 Zynq Block 의 다양한 인터페이스 포트를 보여준다.
 - FIXED IO 는 Zynq 칩의 Programmable System 측 HW 주변 장치에 대한 연결을 나타낸다.
여기에는 Host Computer 에 대한 UART 연결이 포함된다.
 - DDR 은 Board 의 외부 DDR 메모리에 대한 연결을 나타낸다.

계속하려면 OK 를 클릭한다.

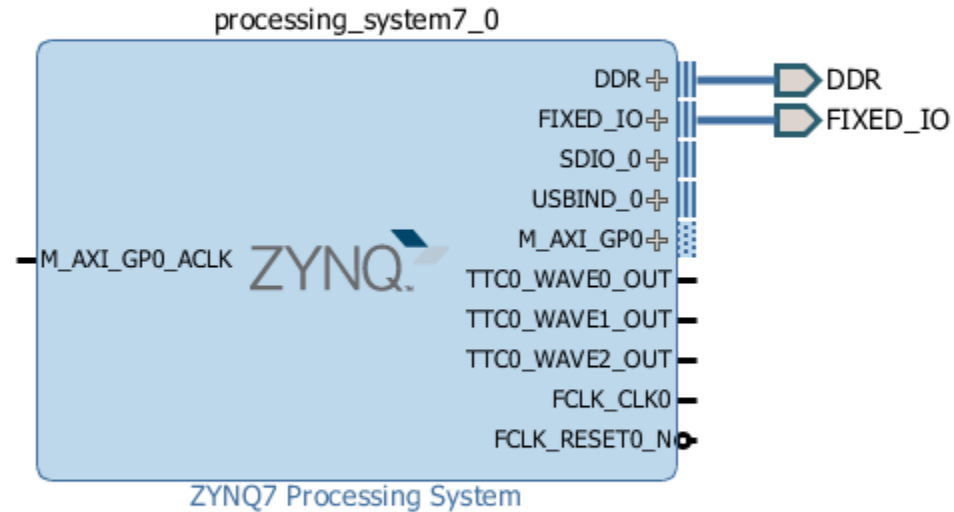


4.5 프로젝트의 요구 사항에 따라 Zynq Block 의 구성을 추가로 변경할 수 있다.

예로 Zynq Block 을 사용하여 서로 다른 주파수의 새로운 Clock 을 생성 할 수 있다.

Zynq Block 을 클릭하고 Customize Block 버튼을 클릭하거나 Zynq 블록을 더블 클릭하여 선택한다.

아래의 Zynq Configuration 드롭 다운은 Zynq 의 Re-Customize IP 대화 상자에 있는 각 화면을 안내한다.



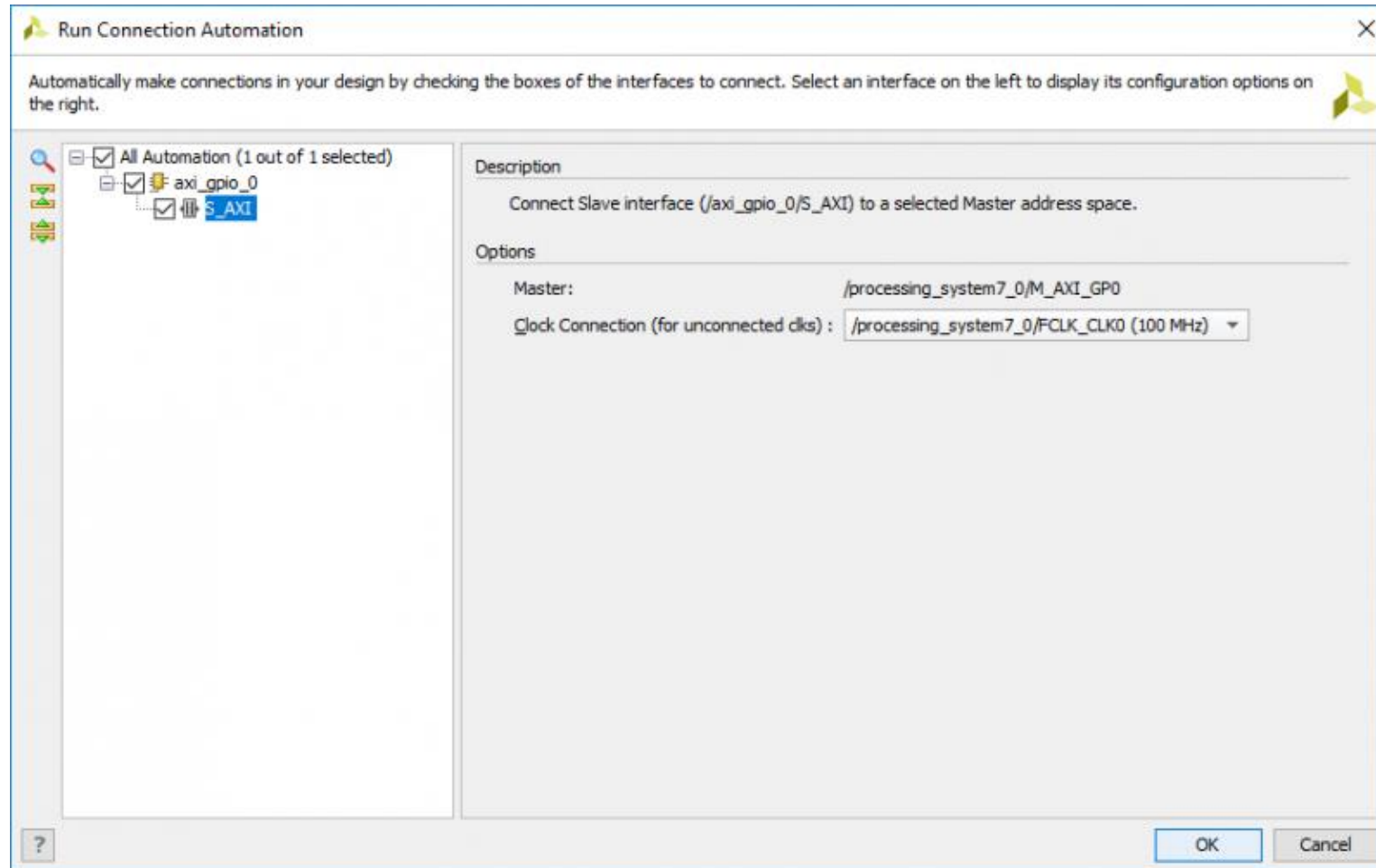
4.6 GPIO 주변 장치를 Zynq Block 에 연결하려면

Block Design Diagram 창의 상단에 있는 녹색 막대에서 Run Connection Automation 을 클릭한다.

Run Connection Automation 대화 상자의 목록 창에서 S_AXI 항목이 선택되고 체크되어 있는지 확인한다.

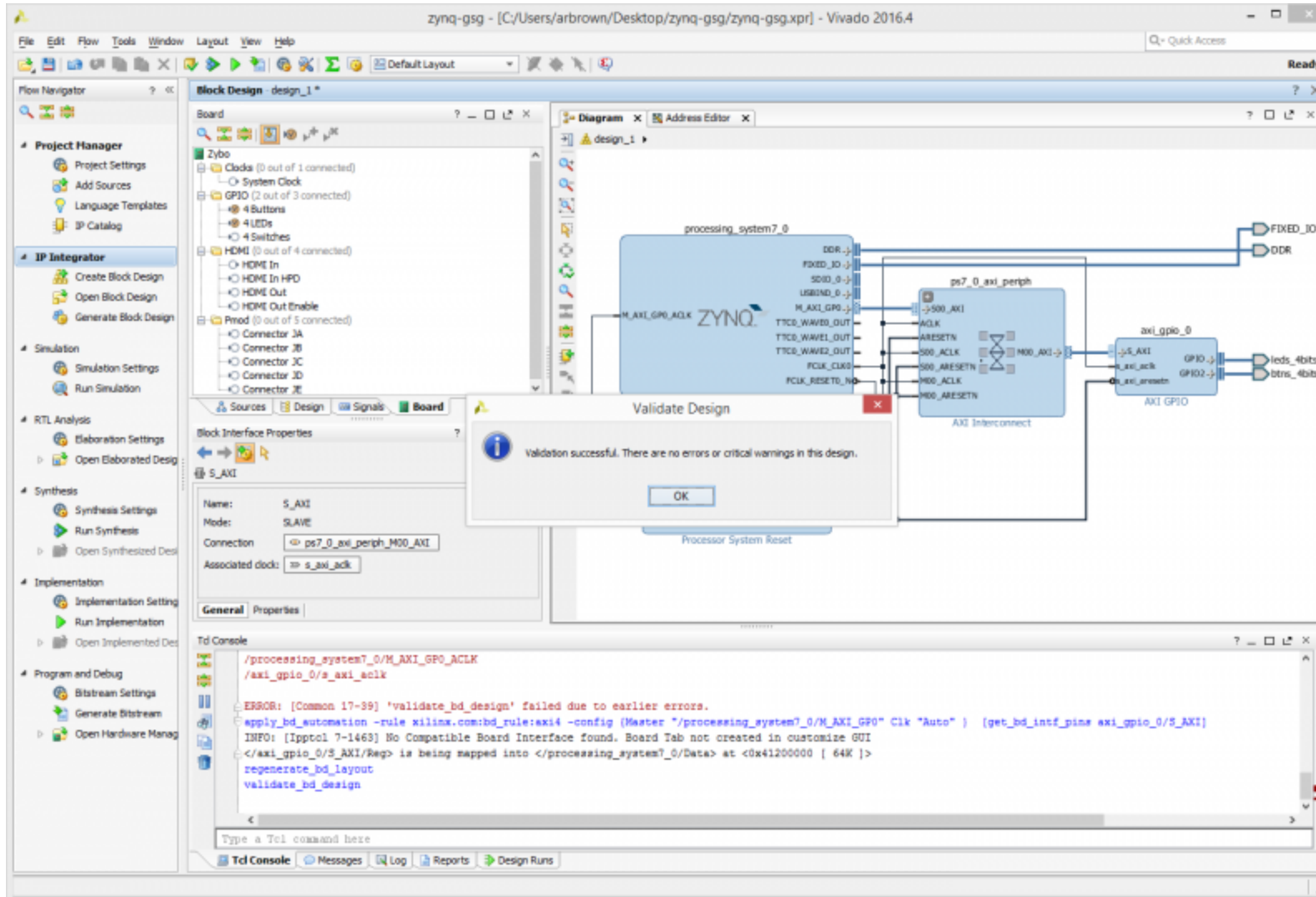
대화 상자의 오른쪽에 있는 옵션 창에서 GPIO IP 를
Zynq Block 에 연결하는 AXI Bus 를 실행하기 위한 Clock Source 를 선택할 수 있다.
드롭 다운 목록에서 /processing_system7_0/FCLK_CLK0' 을 선택한다.

계속하려면 OK 를 클릭한다.



5. Generate the Bitstream

5.1 Vivado 가 모든 것이 잘 보일 것이라고 확인하려면 Validate Design 버튼을 클릭하라.
계속 하려면 OK 를 클릭한다.



5.2 bitstream 을 생성하기 전에 수행해야 할 마지막 작업은 최상위 모듈 파일을 만드는 것이다.

이 파일은 Block Design 을 가져와서 이를 HW Design Language 로 해석하여 합성 및 구현 도구가 올바르게 작동하게 한다.

Block Design 창의 왼쪽에 있는 패널의 Sources 탭에서 Block Design 을 마우스 우클릭한다.

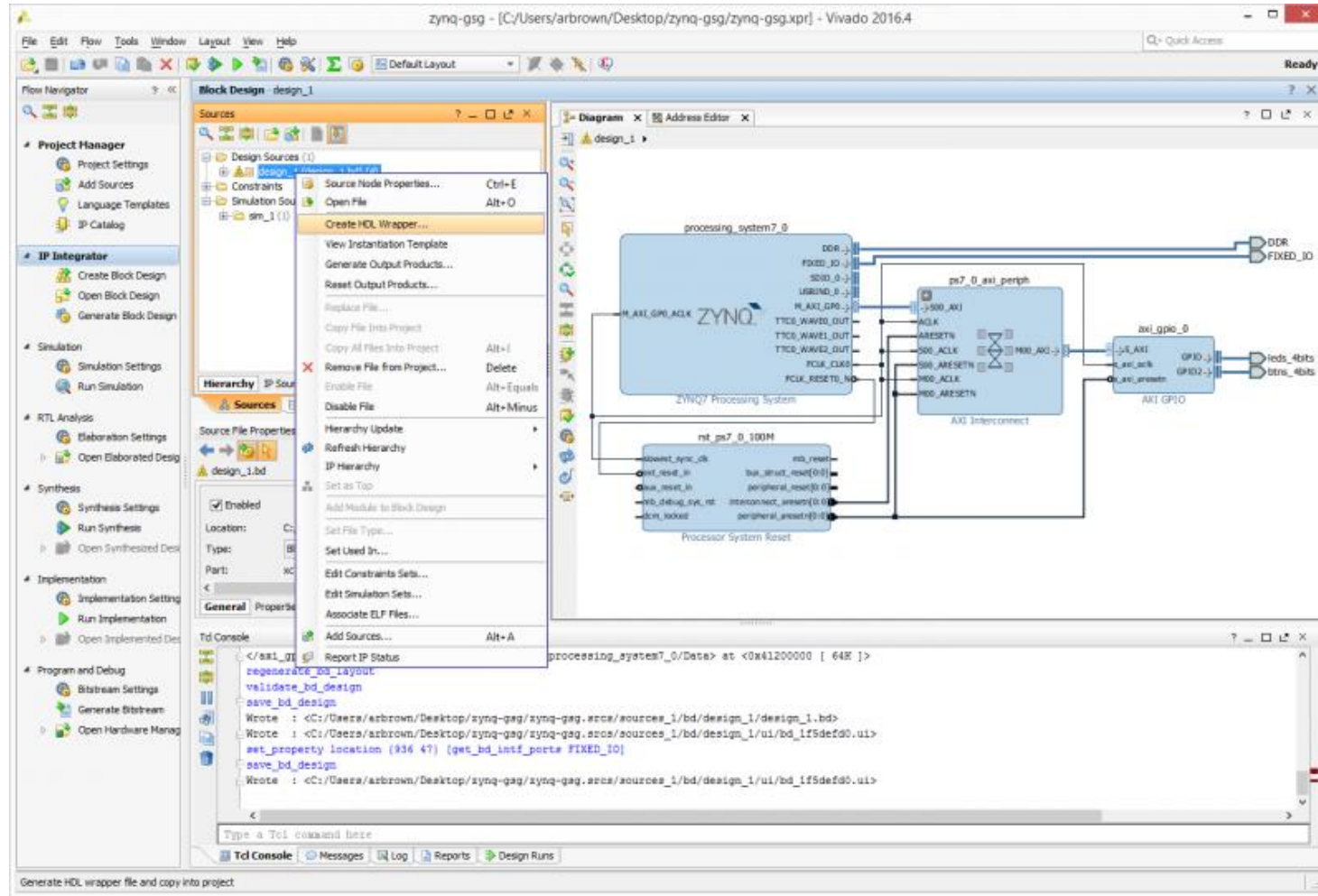
이 패널에는 현재 Board 탭이 선택되어 있을 것이다.

우클릭 메뉴에서 Create HDL Wrapper 를 선택한다.

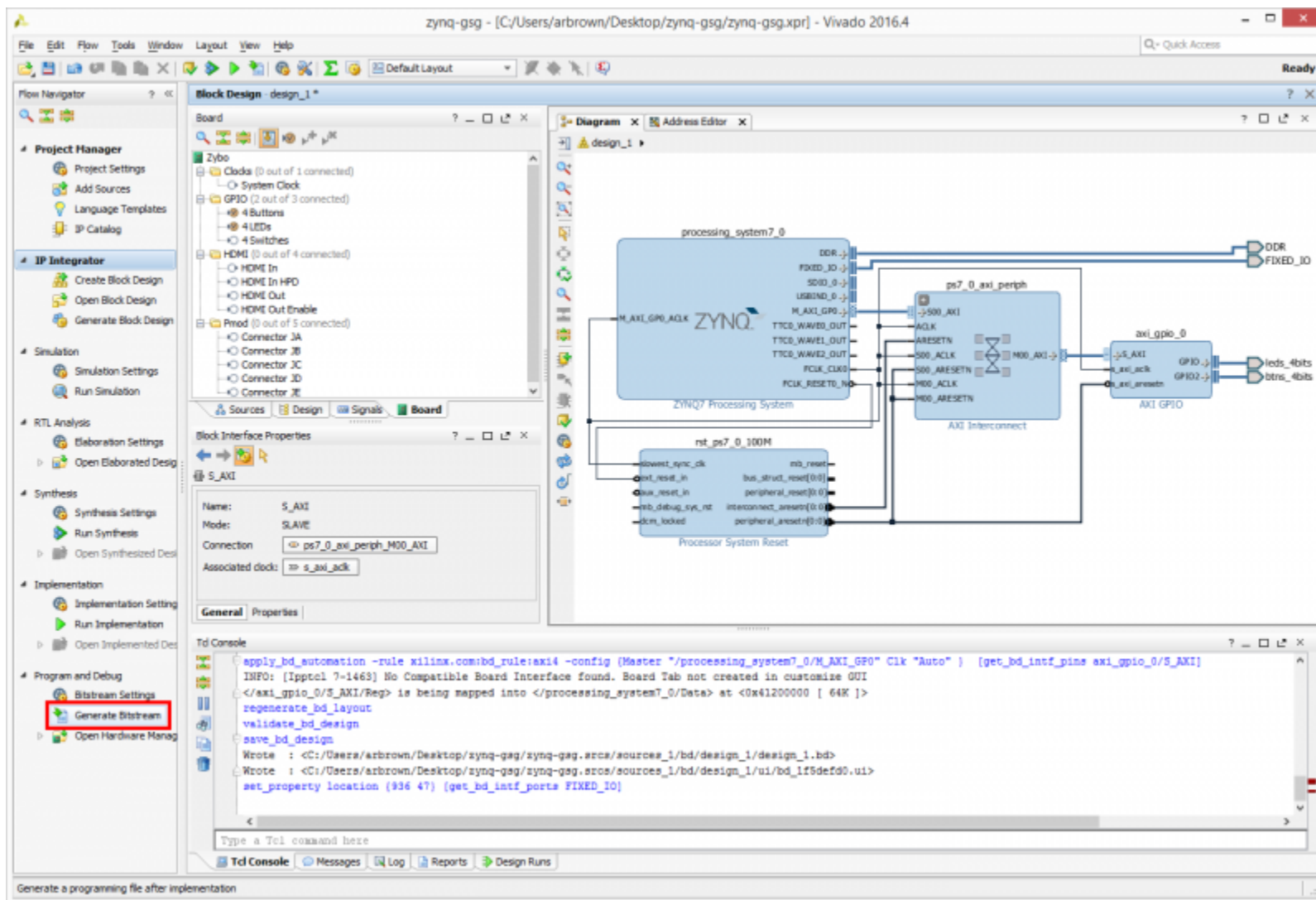
팝업 대화 상자가 나타나면 Vivado 가 Wrapper 를 관리하고 자동 업데이트가 옵션 목록에서 선택되었는지 확인한다.

Wrapper 파일을 수동으로 변경해야 하는 경우 여기에서 다른 옵션을 선택할 수 있지만 고급 사용자를 제외하고는 권장되지 않는다.

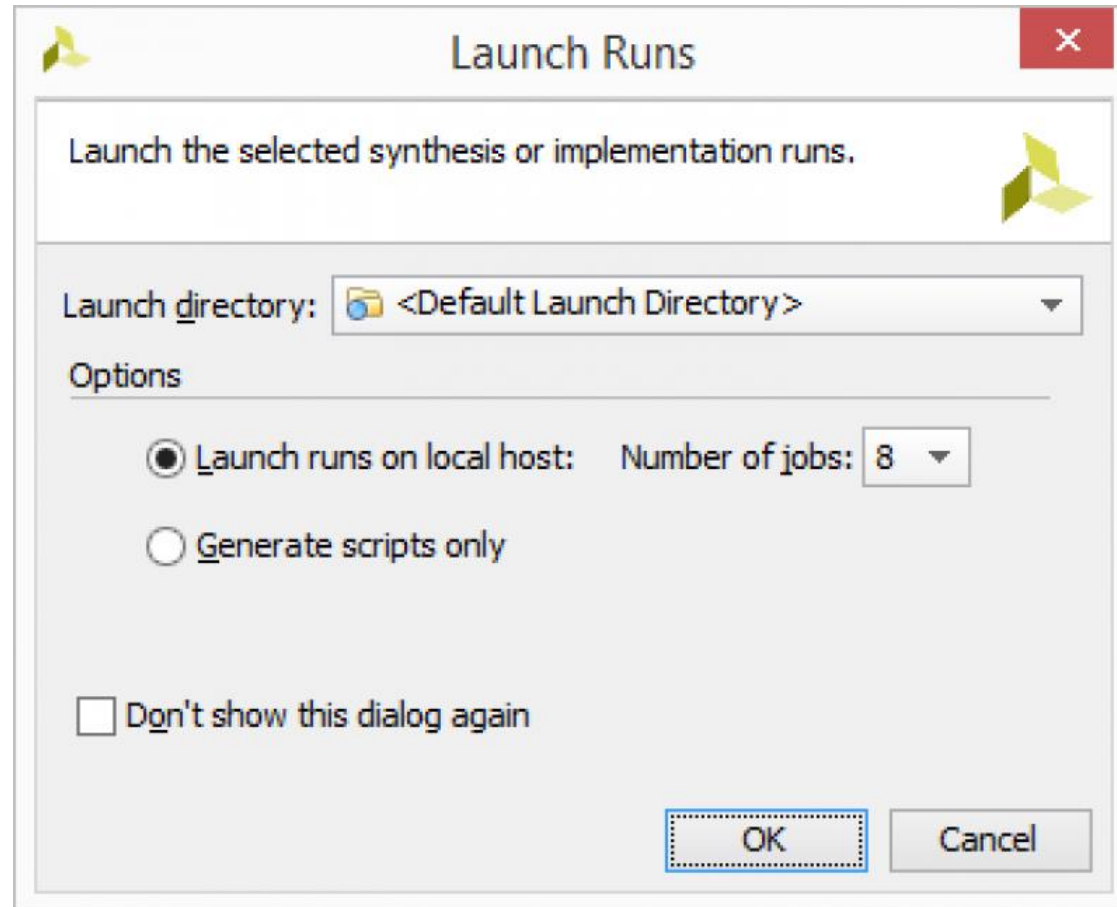
OK 를 클릭하여 Vivado 에서 Wrapper 파일 만들기를 완료한다.



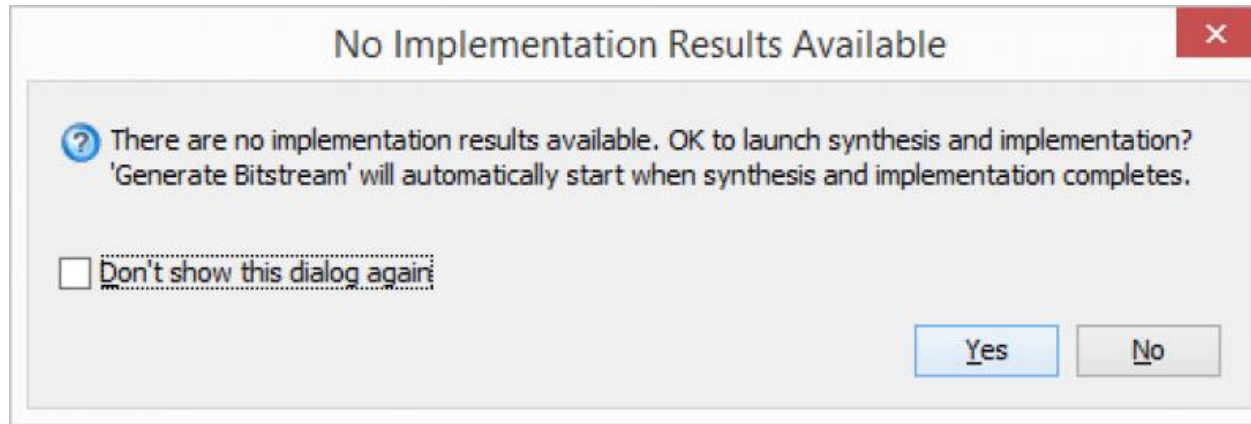
5.3 검증된 Design 과 최상위 모듈을 통해 이제 bitstream 을 생성 할 수 있다. Getting Started with Vivado 가이드는 이 과정에 대해 좀 더 자세히 설명하지만 지금은 Flow Navigator 에서 Generate Bitstream 버튼을 클릭한다.



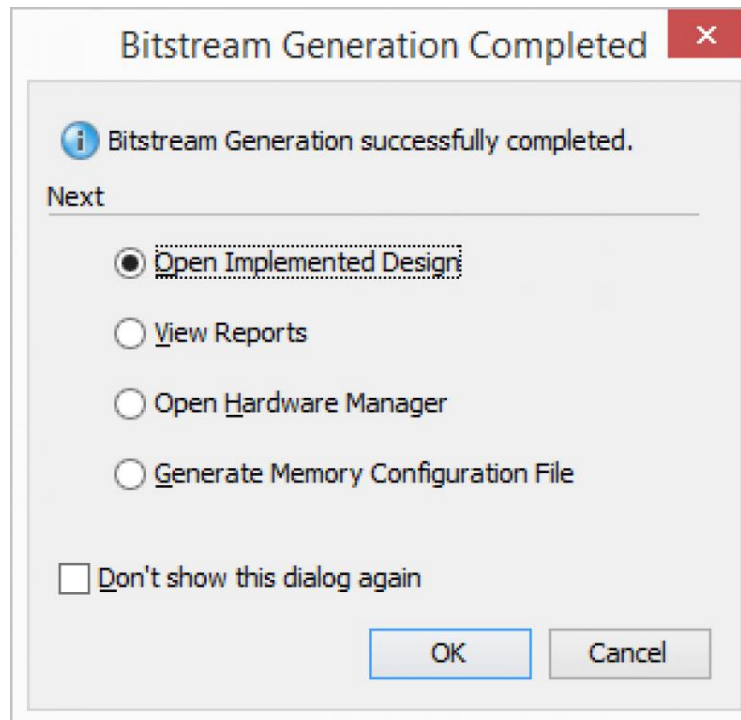
- 5.4 팝업 실행 대화 상자에는 선택할 수 있는 몇 가지 옵션이 있다.
실제로 bitstream 을 생성하려면 Local Host 에서 실행을 선택하라.
다른 옵션은 bitstream 을 생성하지 않은 스크립트만 생성하는 것이지만
다른 컴퓨터에서 bitstream 을 생성하는데 필요한 스크립트다.
Number of jobs(작업 수) 필드는 Vivado 가 실행중인 컴퓨터 CPU 의 리소스 사용량을 조정할 수 있게 한다.
이 프로세스는 복잡한 설계에 시간이 걸릴 수 있으므로 최대한 많은 리소스를 할당하는 것이 좋다.
계속하려면 OK 를 클릭한다.



5.5 다음 대화 상자에서 No Implementation Results Available 을 선택하면 Vivado 는 Synthesis 및 Implementation 을 실행 여부를 묻는다.
이 단계는 bitstream 을 생성하기 위해 필요하므로 Yes 를 클릭한다.



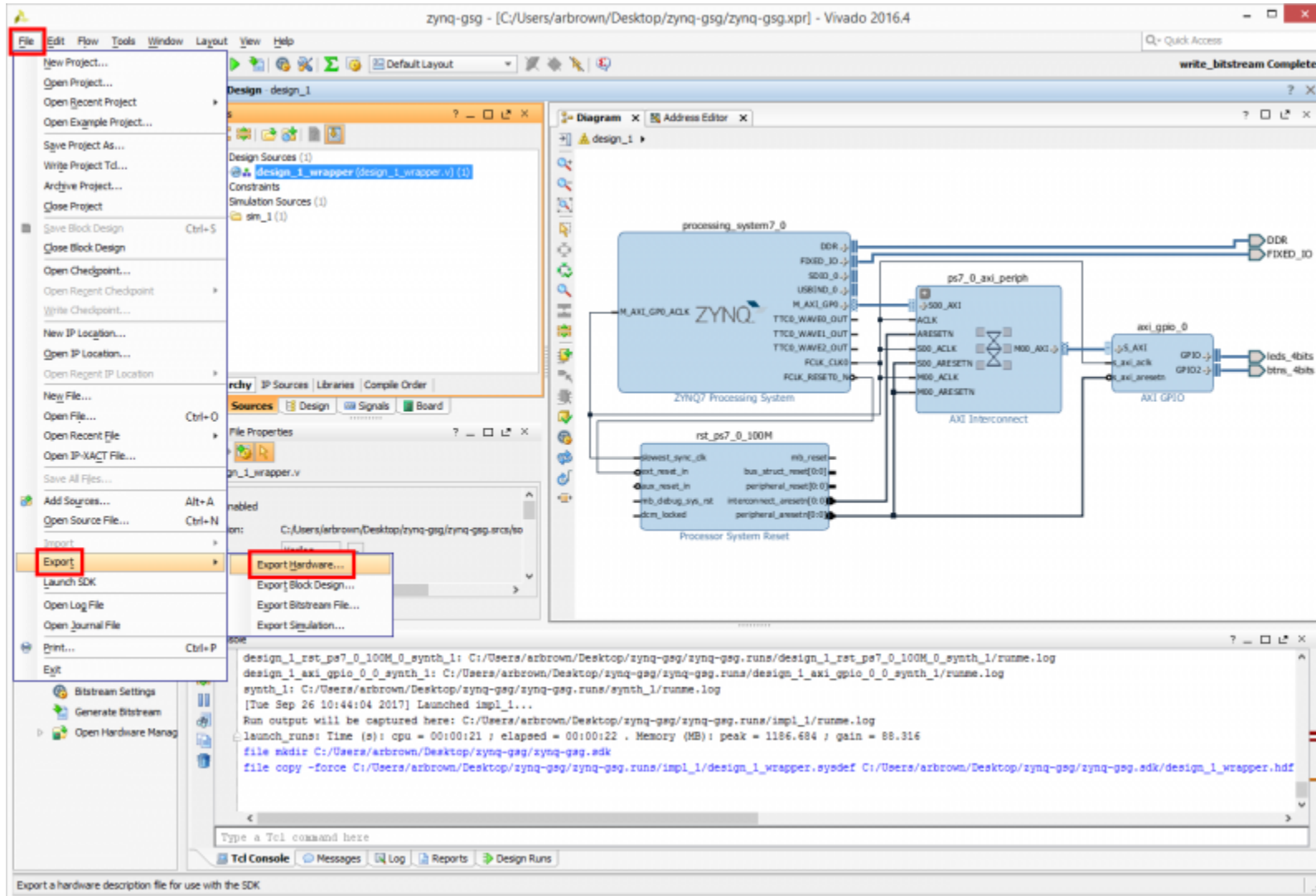
5.6 잠시 시간이 걸릴 수 있는 bitstream 이 생성되면 Vivado 는 다음에 할 일을 묻는다.
이 가이드에서는 사용 가능한 옵션이 없으므로 Cancel 을 클릭한다.



6. Launch Vivado SDK

Vivado SDK 를 시작하기 전에 bitstream 과 Design 에 설치된 주변 장치에 대한 다양한 정보를 포함하는 HW Handoff 파일을 제공해야 한다.

6.1 File 드롭 다운에서 Export 를 선택한 다음 Export to SDK 를 선택한다.



6.2 Vivado SDK 에서 FPGA 를 프로그래밍 할 수 있도록 Include Bitstream 이 선택되어 있는지 확인한다.

Export Location 은 기본적으로 <Local To Project> 이다.

즉 Vivado 는 <project name>.sdk 라는 프로젝트 디렉토리에 <HDL wrapper name>.hdf 라는 HW Handoff 파일을 찾을 수 있는 새 파일을 만든다.

HW Handoff 파일에 대해 다른 위치를 선택한 경우 위치를 기억하라.

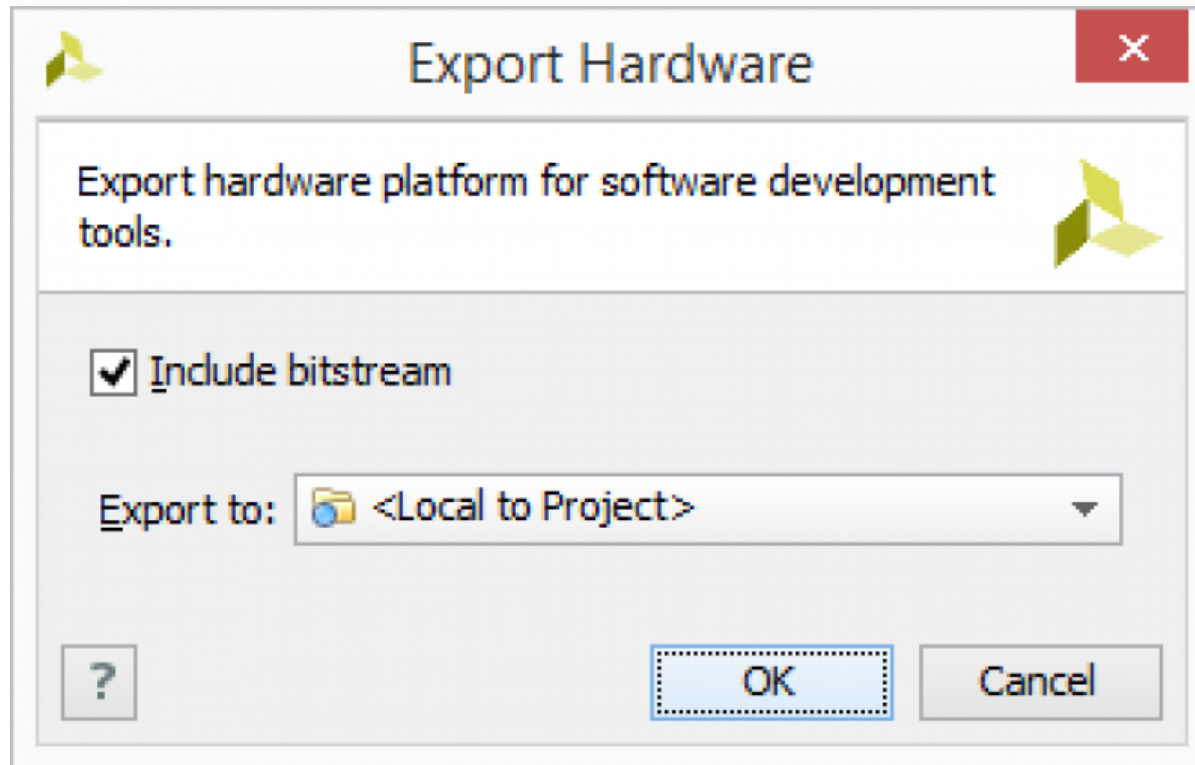
*** Important**

Export Location 경로에 공백을 사용하지 말라.

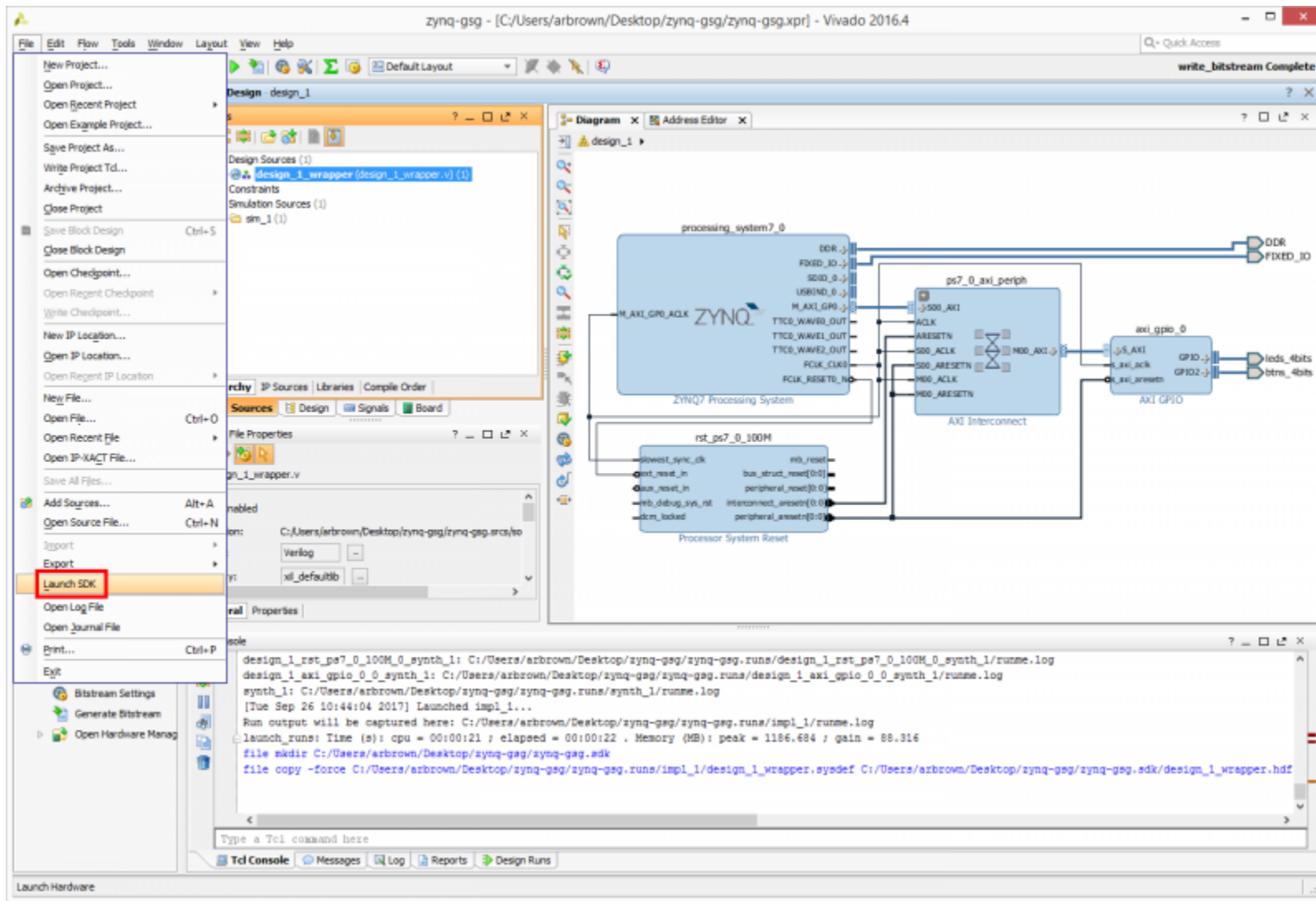
이것은 Vivado 에 문제를 일으킬 것이다.

대신 밑줄, 대시 또는 CamelCase 를 사용하라.

계속하려면 OK 를 클릭하라.



6.3 File 드롭 다운에서 Export 바로 아래의 Launch SDK 를 선택하라.



6.4 나타나는 대화 상자는 Vivado 에서 export 한 파일을 찾을 위치와 SDK 작업 영역을 만들 위치를 알려준다.

이전에 선택한 위치로 Exported Location 을 설정해야 한다.

Workspace 필드는 기본적으로 <Project To Local> 이다.

즉 Vivado 는 프로젝트 디렉토리에 <project name>.sdk 라는 디렉토리를 사용하여 Vivado SDK 에서 생성 된 소스 및 프로젝트를 저장한다.

이 디렉토리가 존재하지 않으면 생성된다.

이 위치는 다른 곳에서 원할 경우 변경할 수 있지만 그대로 두는 것이 좋다.

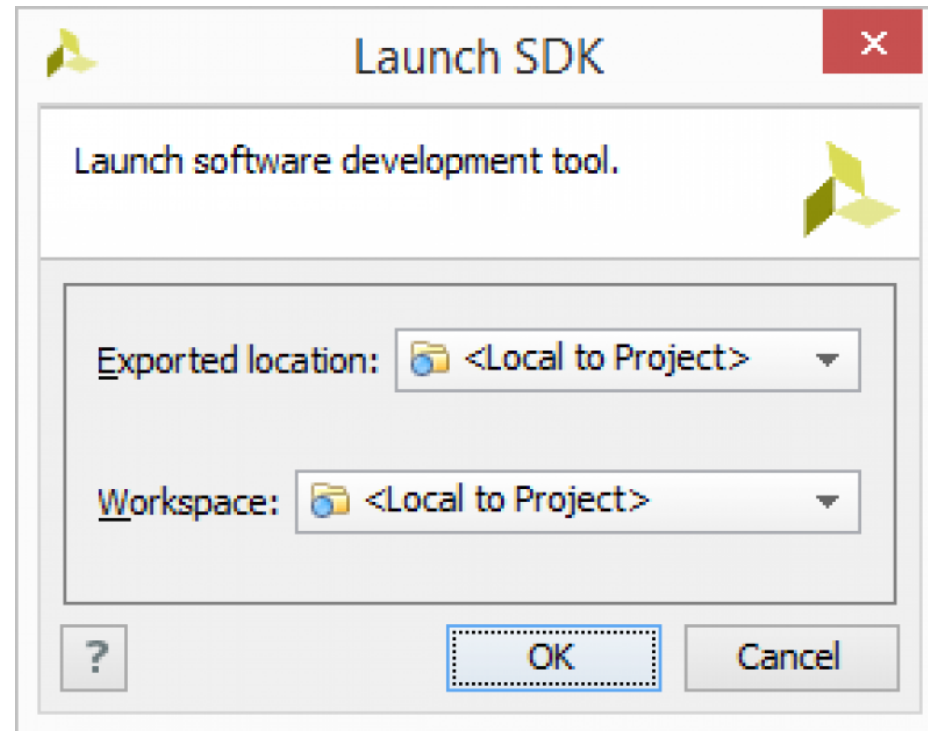
*** Important**

작업 공간 경로에 공백을 사용하지 말라.

이것은 Vivado 에 문제를 일으킬 것이다.

대신 밑줄, 대시 또는 CamelCase 를 사용하라.

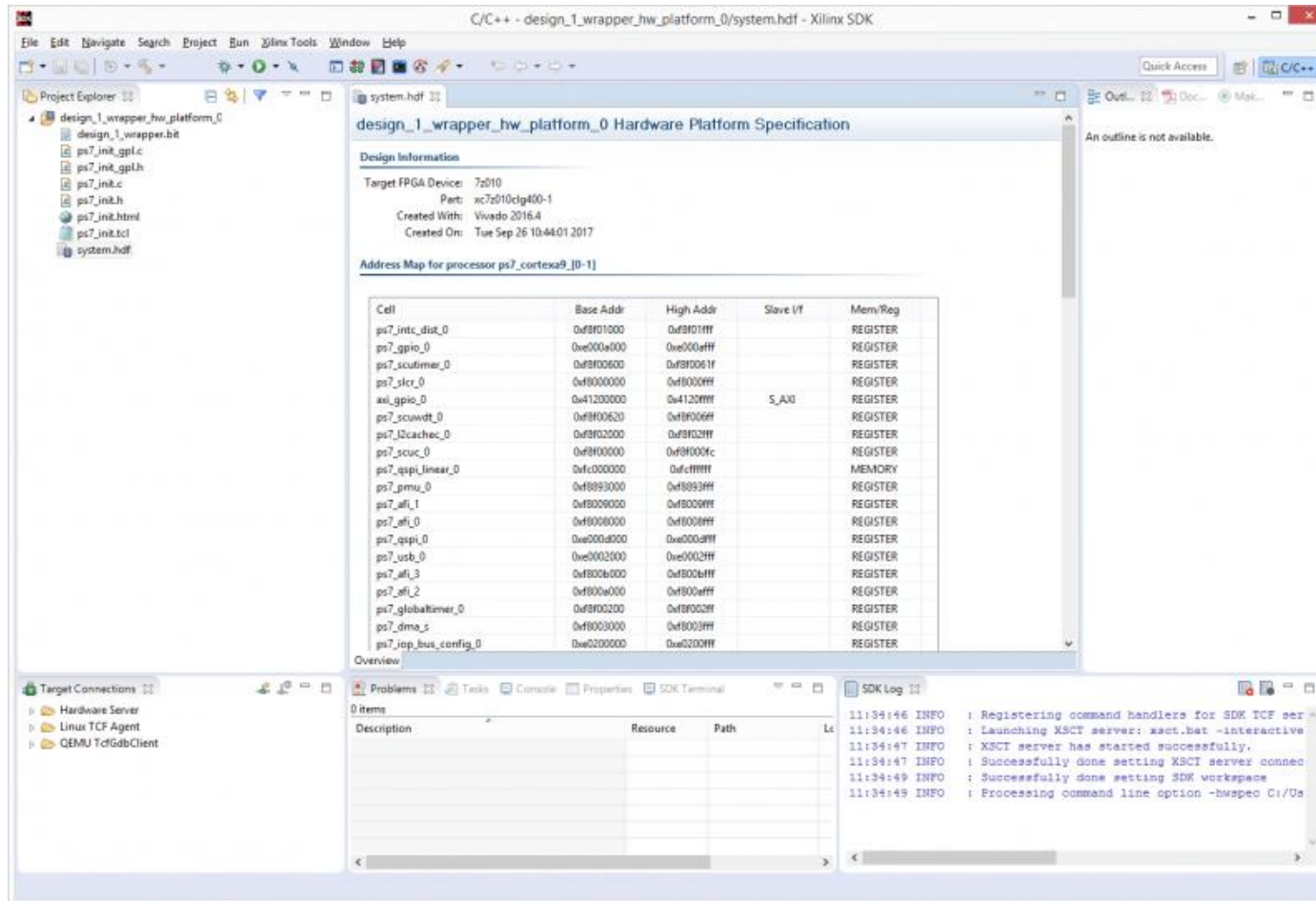
계속하려면 OK 를 클릭하라.



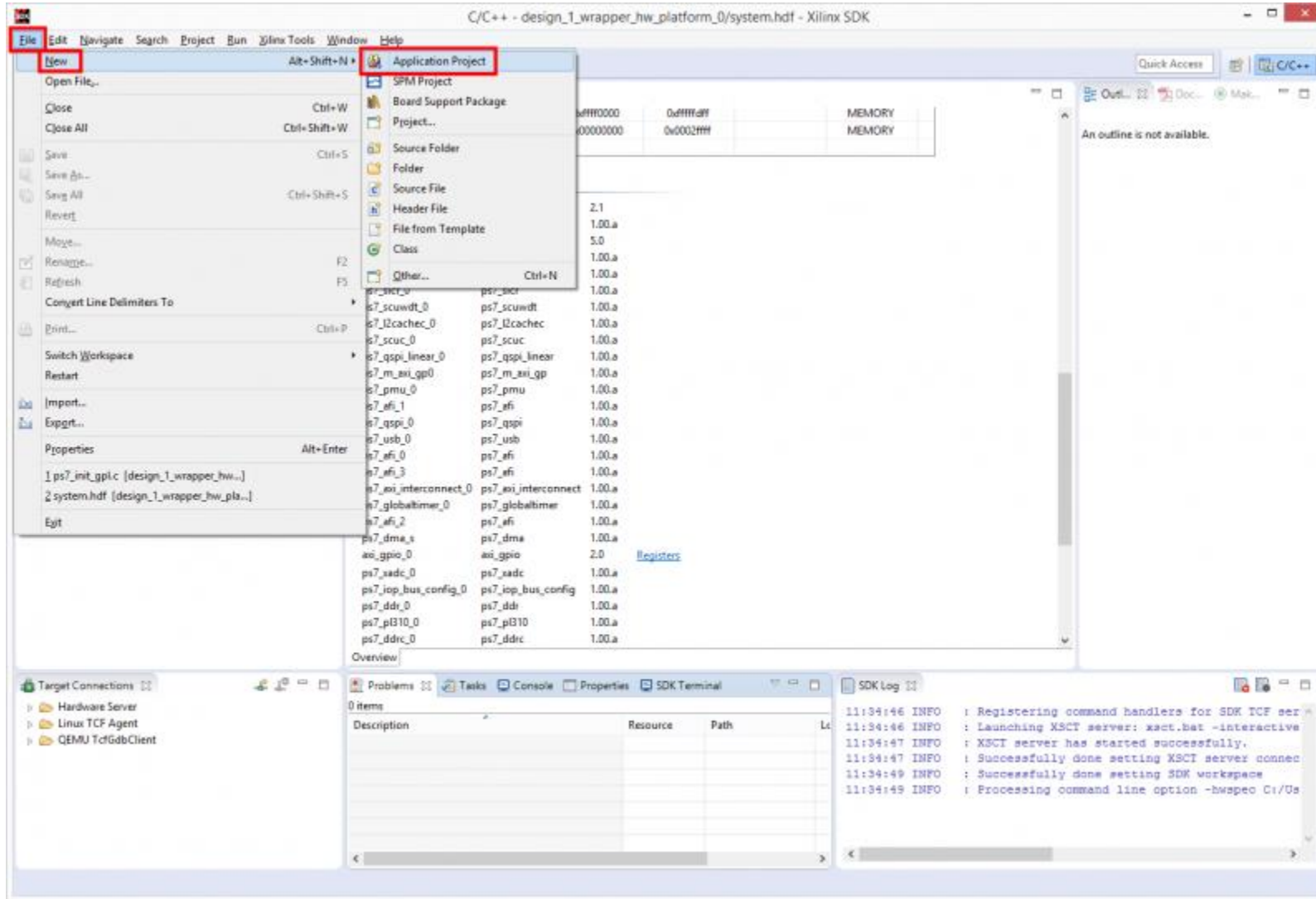
7. Create a New C Project

7.1 Vivado SDK 가 시작되고 HW 정의를 가져 와서 hw_platform 프로젝트만 포함하는 작업 영역을 만든다.
이 프로젝트에는 Vivado 에서 export 한 파일이 포함되어 있다.
이 가이드에는 몇 가지 관련 창만 있다.

- * 창의 왼쪽에 있는 Project Explorer 는 workspace 로 가져온 모든 프로젝트를 표시하며 각 프로젝트는 소스, 스크립트 및 포함된 다른 파일을 볼 수 있도록 확장 할 수 있다.
- * File View 는 창 중앙에 바로 있으며 현재 열려있는 파일을 표시한다.
SDK 가 시작되면 Vivado 에서 내보낸 HW 플랫폼에 대한 Overview 가 표시된다.
Address Map 에 포함 된 AXI GPIO 셀과 IP Block 목록의 해당 항목을 확인하라.



7.2 C Source Code 를 저장할 프로젝트를 만들려면 상단 툴바에서 File 을 클릭하고 New -> Application Project 를 클릭한다.



7.3 New Project 대화 상자에서 프로젝트 이름을 저장하고 Target Software 박스에서 C 가 Language 로 선택되어 있는지 확인한다.

*** Important**

프로젝트 이름에 공백을 사용하지 말라.
이것은 Vivado 에 문제를 일으킬 것이다.
대신 밑줄, 대시 또는 CamelCase 를 사용하라.

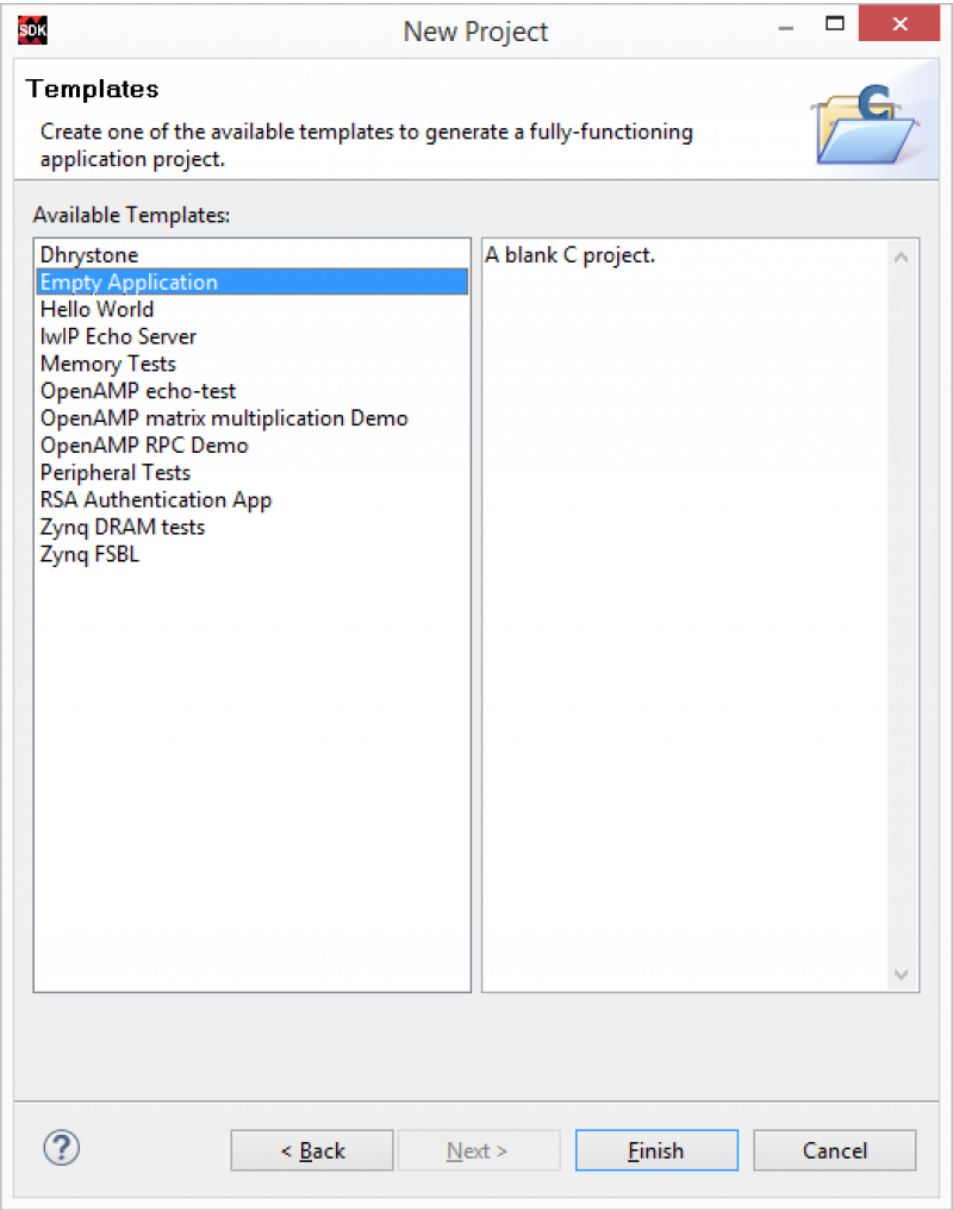
Next 를 클릭하여 계속한다.

The screenshot shows the 'New Project' dialog box in Vivado. The dialog is titled 'New Project' and 'Application Project'. It contains the following fields and options:

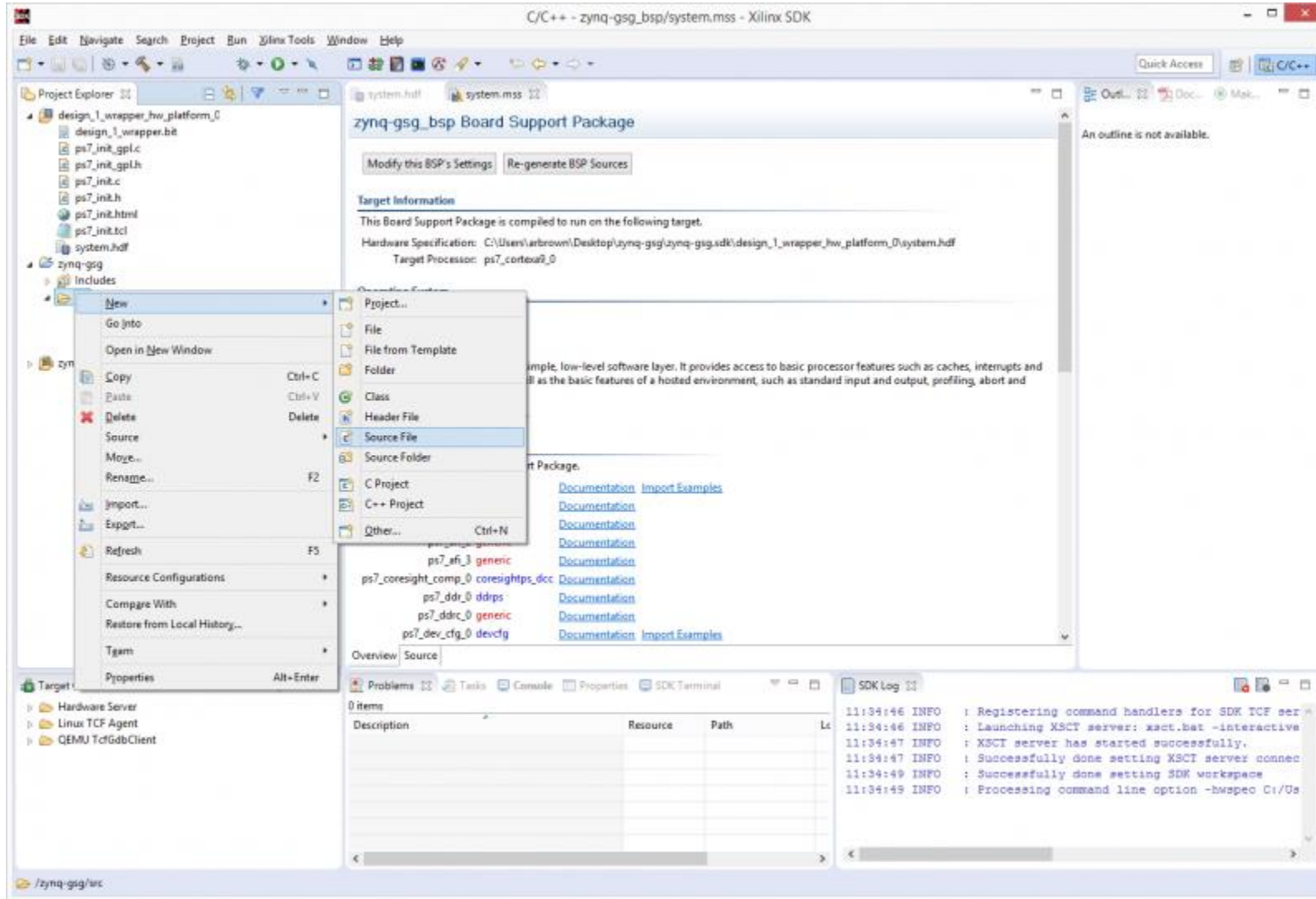
- Project name:** zynq-gsg
- Use default location:** ☒
- Location:** C:\Users\arbrown\Desktop\zynq-gsg\zynq-gsg.sdk\zynq-gs (with a 'Browse...' button)
- Choose file system:** default
- OS Platform:** standalone
- Target Hardware:**
 - Hardware Platform:** design_1_wrapper_hw_platform_0 (with a 'New...' button)
 - Processor:** ps7_cortexa9_0
- Target Software:**
 - Language:** C (selected), C++
 - Compiler:** 32-bit
 - Board Support Package:** Create New (selected), zynq-gsg_bsp (with a 'Use existing' option and a dropdown menu)

At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted), and 'Cancel'.

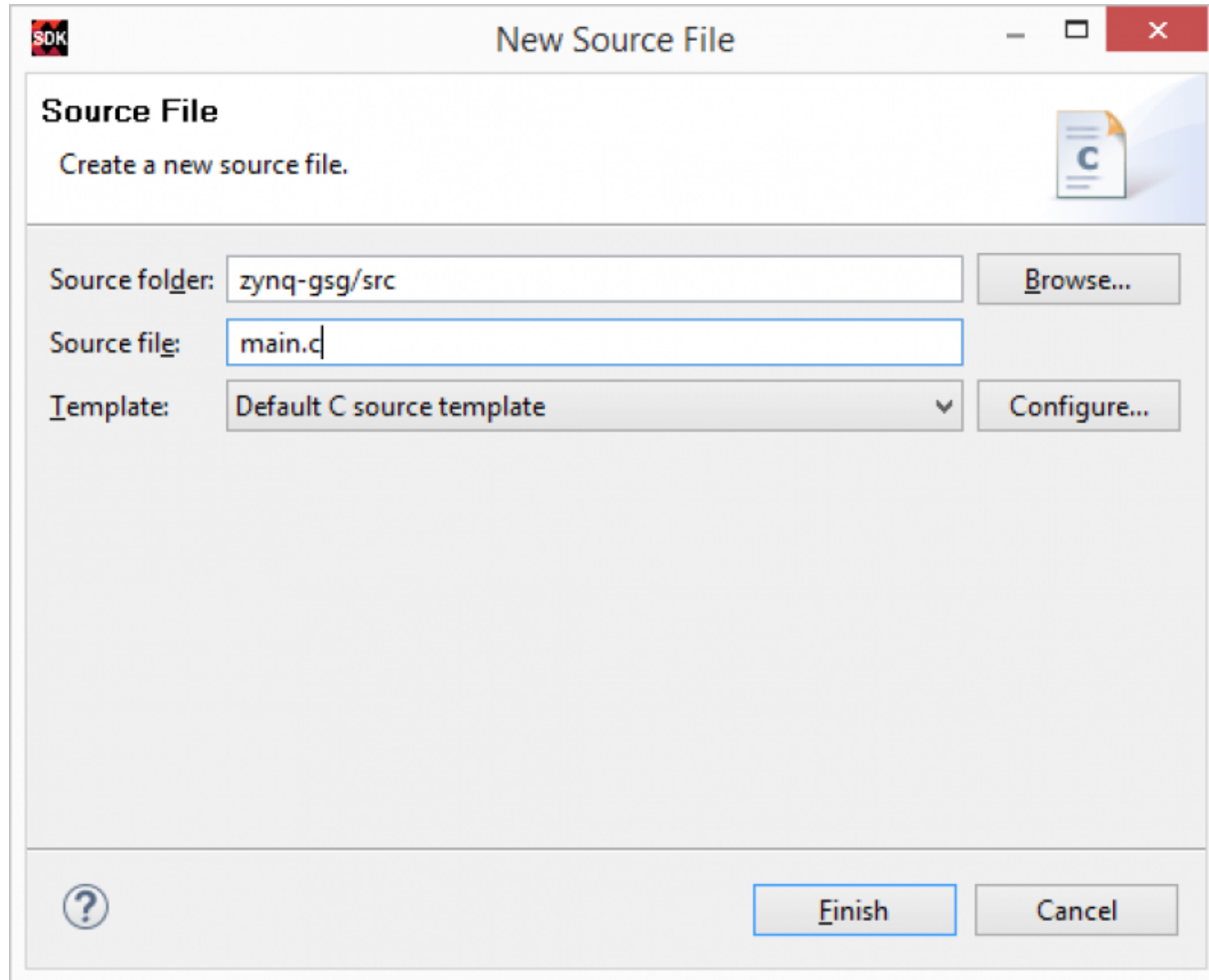
7.4 사용 가능한 템플릿 목록에서 Empty Application 을 선택한 다음 Finish 를 클릭한다.



7.5 Board Support Package 가 아닌 새로 생성 된 프로젝트를 확장 한 다음 'src' 폴더를 확장하라.
'src' 폴더를 마우스 우클릭하고 메뉴에서 New -> Source File 을 선택한다.



7.6 새 소스 파일의 이름을 main.c 로 지정한 다음 Finish 를 클릭한다.



7.7 새로운 main 파일이 이제 File View 창에 열린다.

다음 내용을 입력하고 Ctrl-S 를 눌러 파일을 저장하고 프로젝트를 빌드한다.

```
//AXI GPIO driver
#include "xgpio.h"
```

```
//send data over UART
#include "xil_printf.h"
```

```
//information about AXI peripherals
#include "xparameters.h"
```

```
int main()
{
    XGpio gpio;
    u32 btn, led;

    XGpio_Initialize(&gpio, 0);

    XGpio_SetDataDirection(&gpio, 2, 0x00000000); // set LED GPIO channel tristates to All Output
    XGpio_SetDataDirection(&gpio, 1, 0xFFFFFFFF); // set BTN GPIO channel tristates to All Input

    while (1)
    {
        btn = XGpio_DiscreteRead(&gpio, 1);

        if (btn != 0) // turn all LEDs on when any button is pressed
            led = 0xFFFFFFFF;
        else
            led = 0x00000000;

        XGpio_DiscreteWrite(&gpio, 2, led);

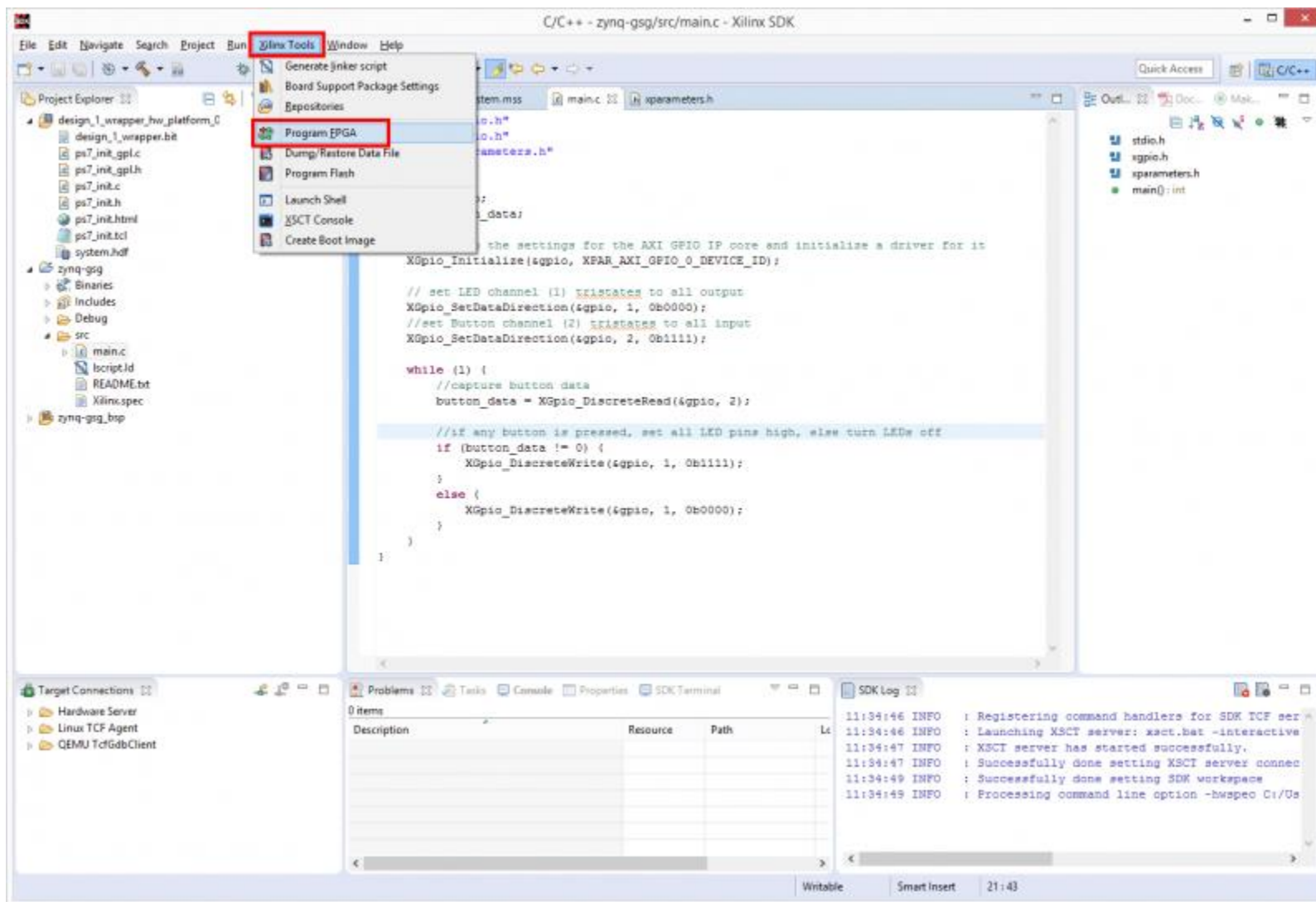
        xil_printf("Button state: %08x", btn);

    }
}
```

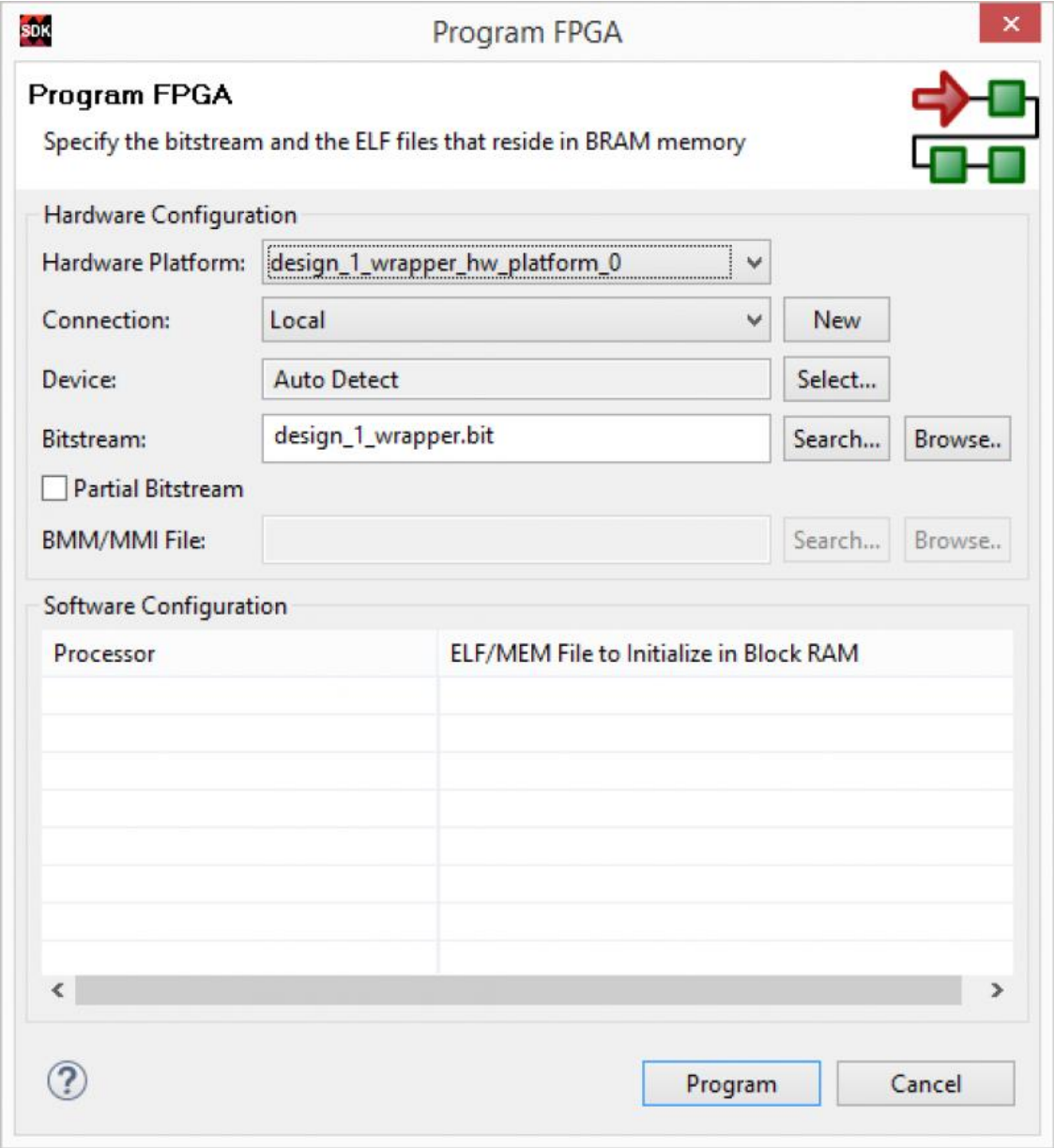
8. Program and Run the Design

다음 단계는 FPGA 에 비트 파일을 프로그래밍하고 프로세서에서 C Source 를 실행하는 것이다.

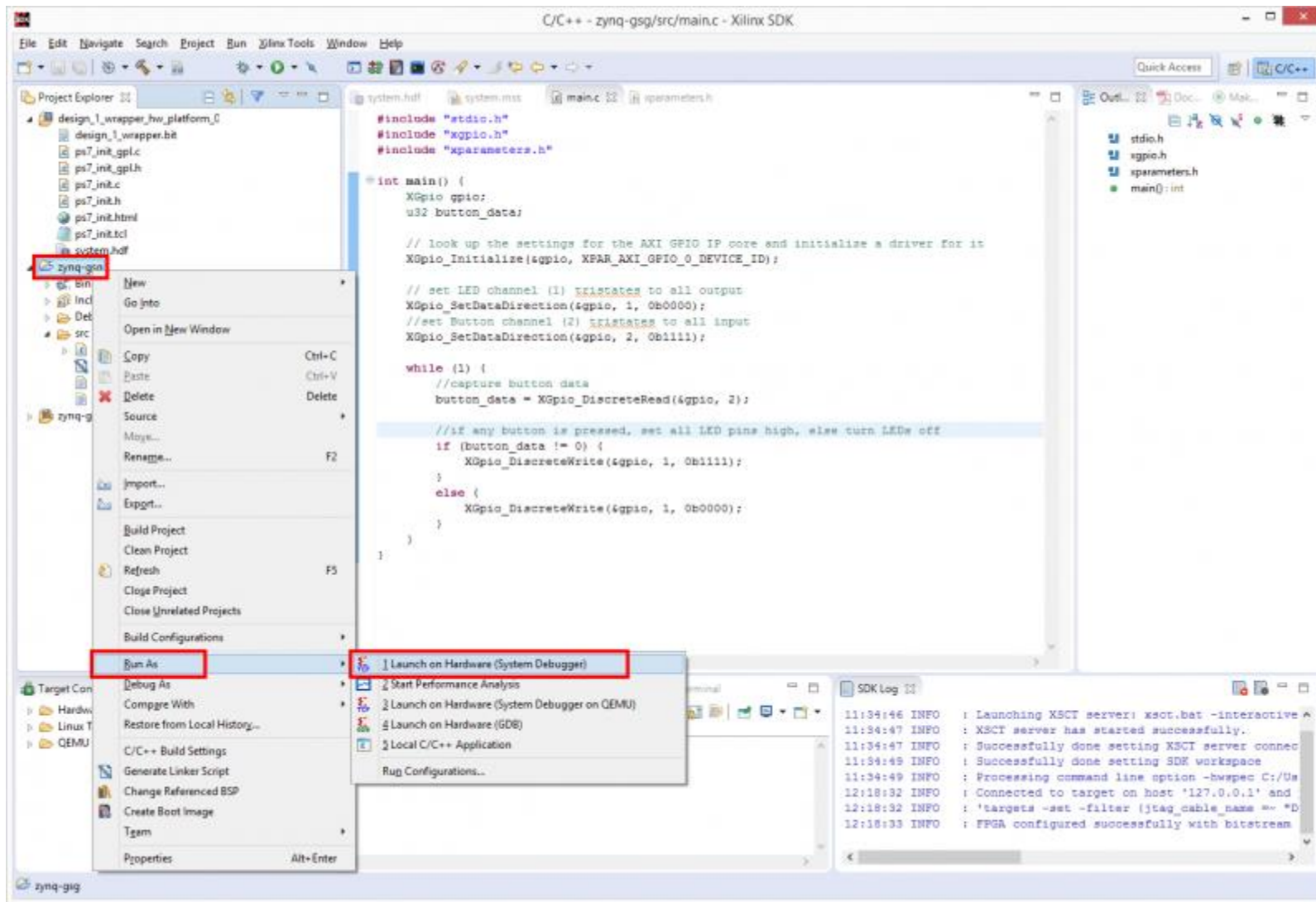
8.1 상단 툴바에서 Xilinx Tools 를 선택한 다음 Program FPGA 를 선택한다.



8.2 Program FPGA 대화 상자의 모든 설정을 그대로 둔 다음 Program 을 클릭한다.
이는 프로세서가 아닌 FPGA 를 프로그램 하는 것이다.



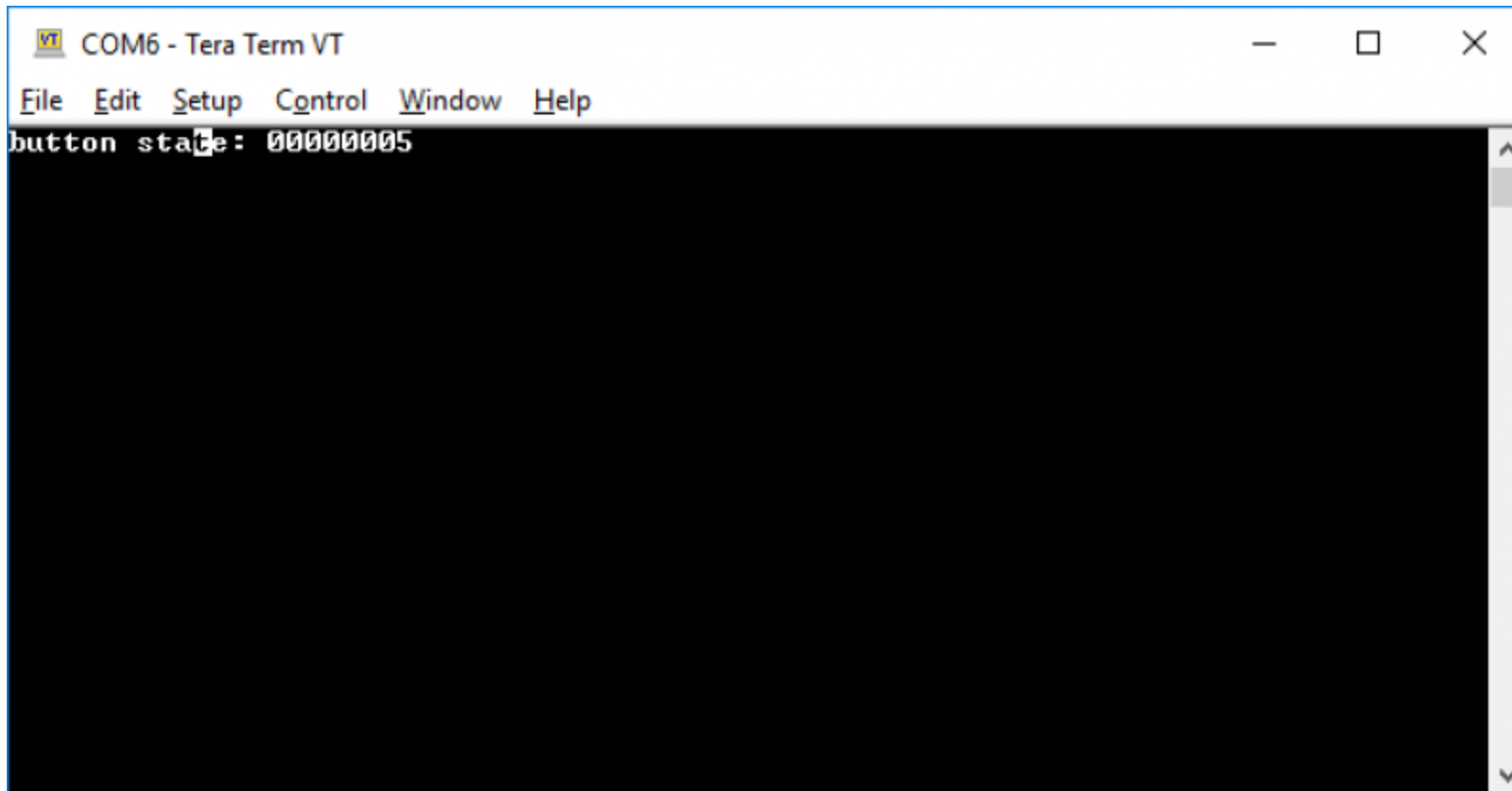
8.3 Project Explorer 에서 응용 프로그램 프로젝트를 마우스 우클릭 한 다음 Run As -> Launch on Hardware(System Debugger) 를 선택한다.



9. Receiving Messages over UART

- 9.1 프로젝트의 C 소스 코드에서 `xil_printf` 문에 의해 보내지는 UART 메시지를 수신하려면 Tera Term 과 같은 직렬 콘솔 애플리케이션을 사용하는 것이 좋다.
연결할 포트는 Windows 의 장치 관리자를 검토하여 확인할 수 있다.
직렬 포트 설정은 Zynq Block 또는 AXI Uartlite Block 의 Customization 설정에 따라 결정된다.

일반적으로 이러한 설정은 8 Data Bits, No Parity Bit, 1 Stop Bit 가 되며
Zynq 의 경우 115200, MicroBlaze 의 경우 9600 의 전송 속도가 기본값이다.



Congratulations!

이제 프로젝트가 Board 에서 실행된다.

버튼을 눌러 LED 가 켜지는지 확인한다.

Tera Term 을 사용하여 버튼 데이터를 16 진수로 본다.

References

1. <https://reference.digilentinc.com/vivado/getting-started-with-ipi/start>