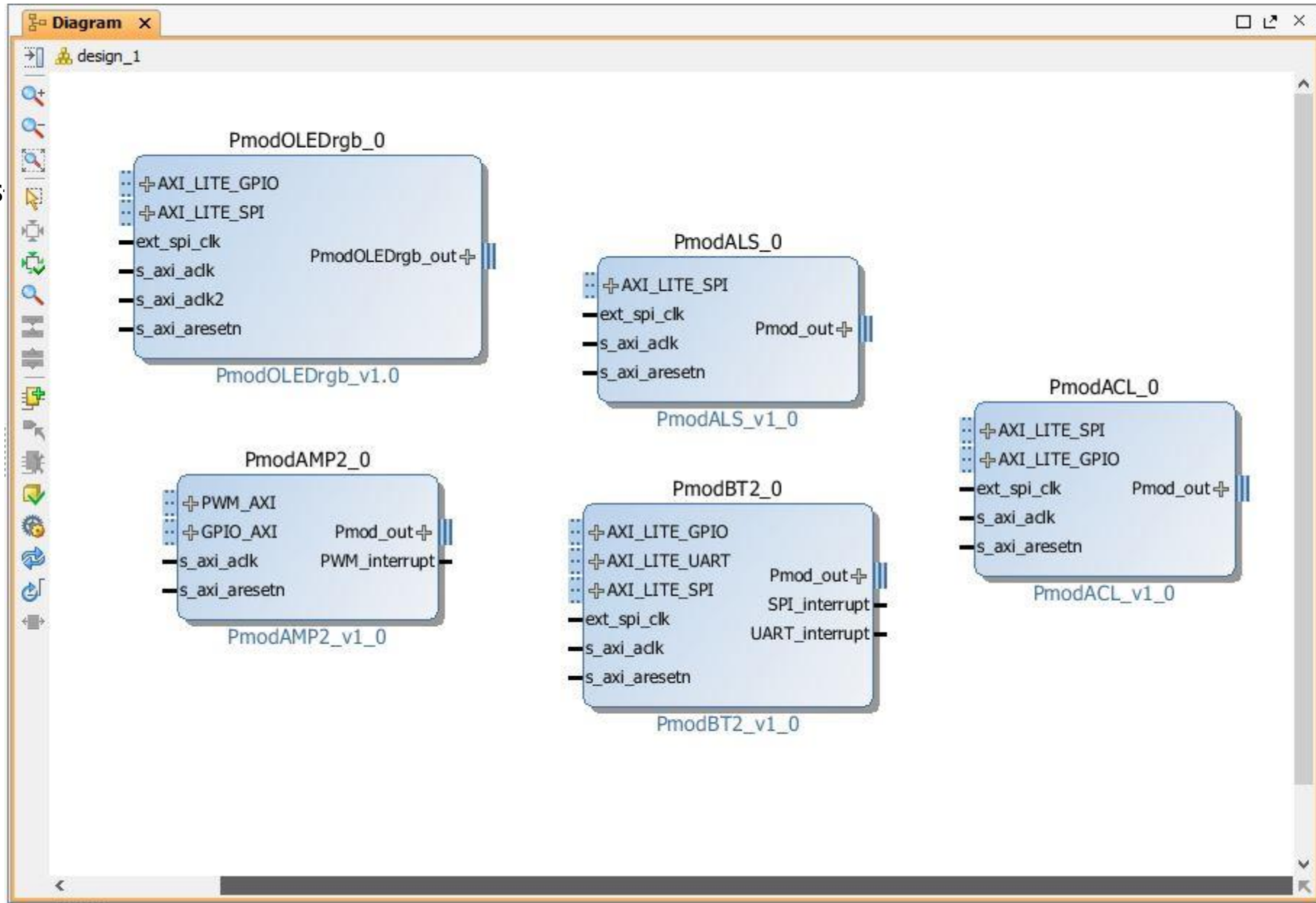


Getting Started with Digilent Pmod IPs

Innova Lee(이상훈)
gcccompil3r@gmail.com

Getting S



Overview

Digilent 는 가능한 한 간단하게 FPGA 에서 Pmod 를 구현하고 사용하도록 설계된 여러 가지 IP 를 제공한다.
이 가이드는 Vivado Microblaze 또는 Zynq 설계에서 Pmod IP 코어를 사용하는 방법을 설명한다.

이 튜토리얼의 마지막에는 Digilent Pmod IP 코어를 사용하는 FPGA 또는 Zynq 플랫폼을 위한 Vivado 설계 및 데모가 제공된다.

아래 2 개의 정보는 이 튜토리얼에서 지원되는 플랫폼과 각 튜토리얼을 완료하는데 필요한 세부 정보를 보여준다.

Platform	Processor Type
Arty	Microblaze
Arty S7	Microblaze
Arty Z7	Zynq
Basys3	Microblaze
Cmod A7	Microblaze
Genesys2	Microblaze
Nexys4	Microblaze
Nexys4-DDR	Microblaze
Nexys Video	Microblaze
Zybo	Zynq
Zybo Z7	Zynq

▼ Pmods Supported

Pmod	Interface Type	Reference clock frequency (MHz)	Reference Clock signal name	Interrupt pin name/s	Uses PmodGPIO
8LD	GPIO	-	-	-	Yes
ACL	SPI	80	ext_spi_clk	-	-
ACL2	SPI	50	ext_spi_clk	-	-
AD1	SPI	-	-	-	-
AD2	IIC	-	-	-	-
ALS	SPI	50	ext_spi_clk	-	-
AMPs	GPIO	-	-	timer_interrupt	-
Breadboard	GPIO	-	-	-	-
BB	GPIO	-	-	-	Yes
BT2	UART	-	-	-	-
BTN	GPIO	-	-	-	Yes
CAN	SPI	100	ext_spi_clk	-	-
CLS	SPI	50	ext_spi_clk	-	-
CMPS2	IIC	-	-	-	-
COLOR	IIC	-	-	-	-

Prerequisites

* Hardware

지원되는 Digilent 7 시리즈 FPGA 또는 Zynq 보드
USB 케이블
하나 이상의 지원되는 Digilent Pmod

* Software

- Xilinx Vivado 2015.4 혹은 2016.4, 자일링스 SDK 포함
Vivado 2015.4 가 이 튜토리얼에서 사용된다.
Vivado 의 다른 버전에서도 작동하지만 기능이 보장되지 않는다.
- Digilent 보드 지원 파일
Vivado 2015.X 용 보드 지원 파일을 설치하는 방법에 대한 위키 가이드를 따르라.
<https://reference.digilentinc.com/learn/software/tutorials/vivado-board-files/start?redirect=1>
- Digilent Vivado IP Library
Vivado Library Releases
<https://github.com/Digilent/vivado-library/releases>
이 파일의 설치는 이 튜토리얼의 2 단계에서 다룬다.

* Important

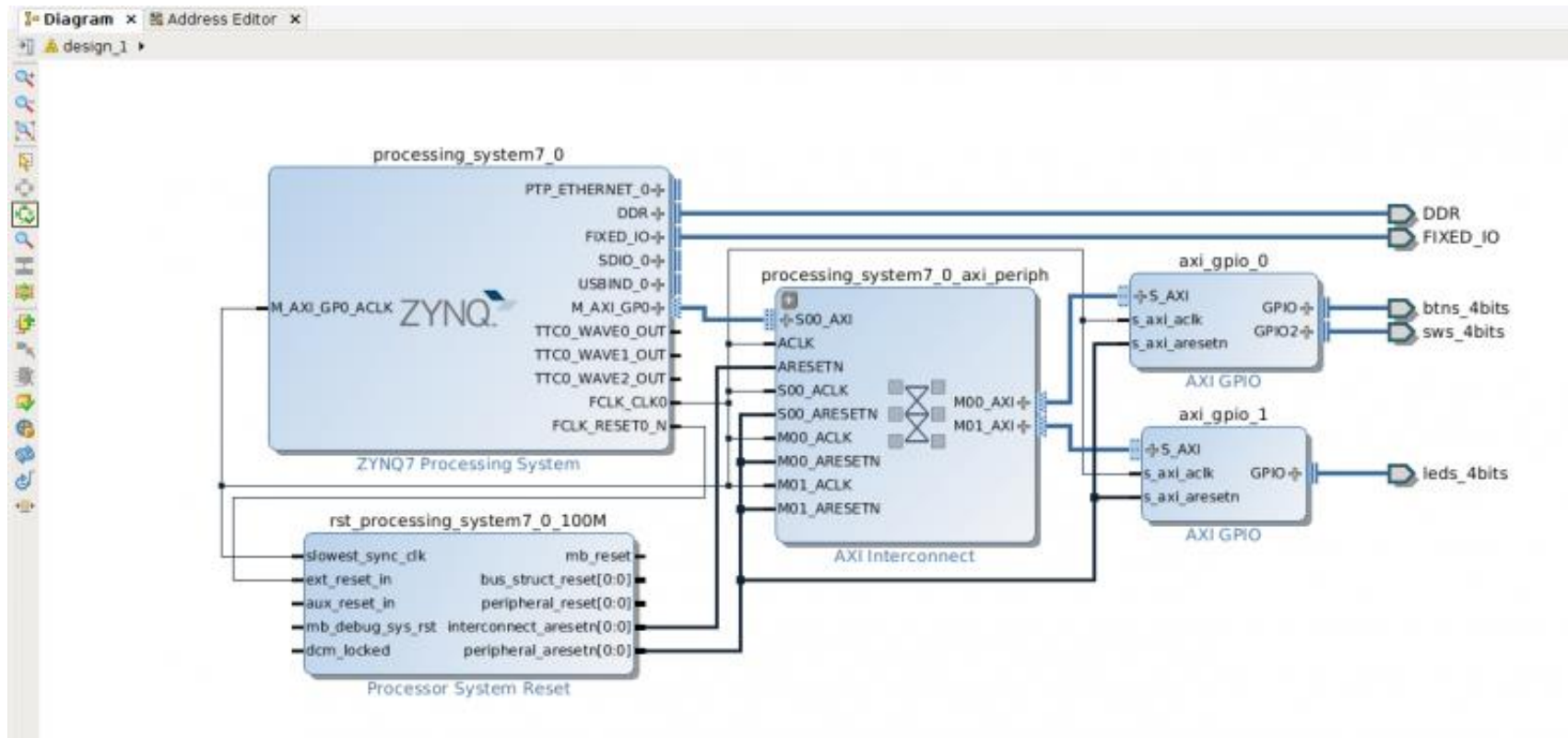
사용되는 Pmod IP 에 README 파일이 있으면 이 튜토리얼을 시작하기 전에 확인하라.
이 파일은 vivado-library/ip/Pmods/"your pmod" 디렉토리에서 찾을 수 있다.

Tutorial

1. 새로운 Microblaze / Zynq 블록 설계 만들기

이 튜토리얼에서 Microblaze 또는 Zynq 를 사용해야 하는지 여부를 결정하려면 이 튜토리얼의 개요 섹션에 있는 Platform Supported 표의 항목을 참조하라.
또는 플랫폼에 맞는 Resource Center 로 이동하라.

플랫폼의 Resource Center 에서 "Getting Started with Zynq" 튜토리얼을 따라 기본 Zynq 블록 설계를 얻으라.



2. Digilent 라이브러리 저장소 추가

2.1) 버전 번호가 사용중인 Vivado 버전과 일치하는 Digilent 의 vivado 라이브러리 저장소의 최신 버전을 찾는다.

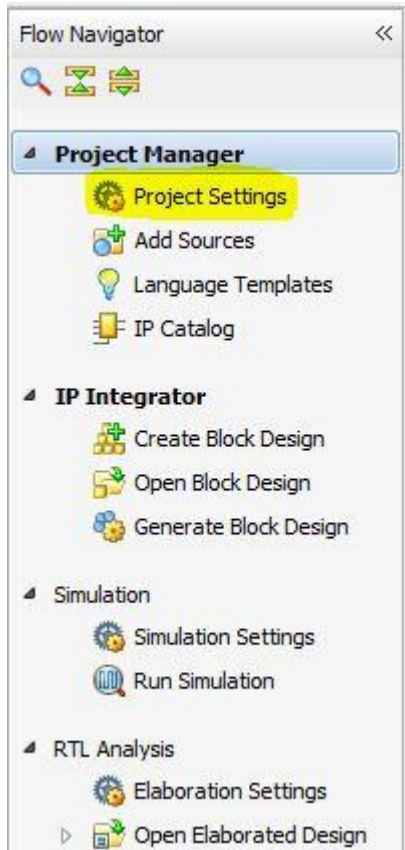
(예: "v2016.4-1" 은 Vivado 2016.4 의 첫 번째 배포에 해당한다)

vivado-library-<version>.zip 파일(소스 코드 아카이브 중 하나가 아님)을

다운로드 한 이후에 아카이브를 기억하기 쉬운 위치에서 압축해제한다.

이 GitHub 저장소에는 Digilent 의 모든 Pmod IP 코어 및 Pmod 인터페이스 설명을 포함하여 Digilent 보드 용으로 사용하려는 다수의 IP 코어가 포함되어 있다.

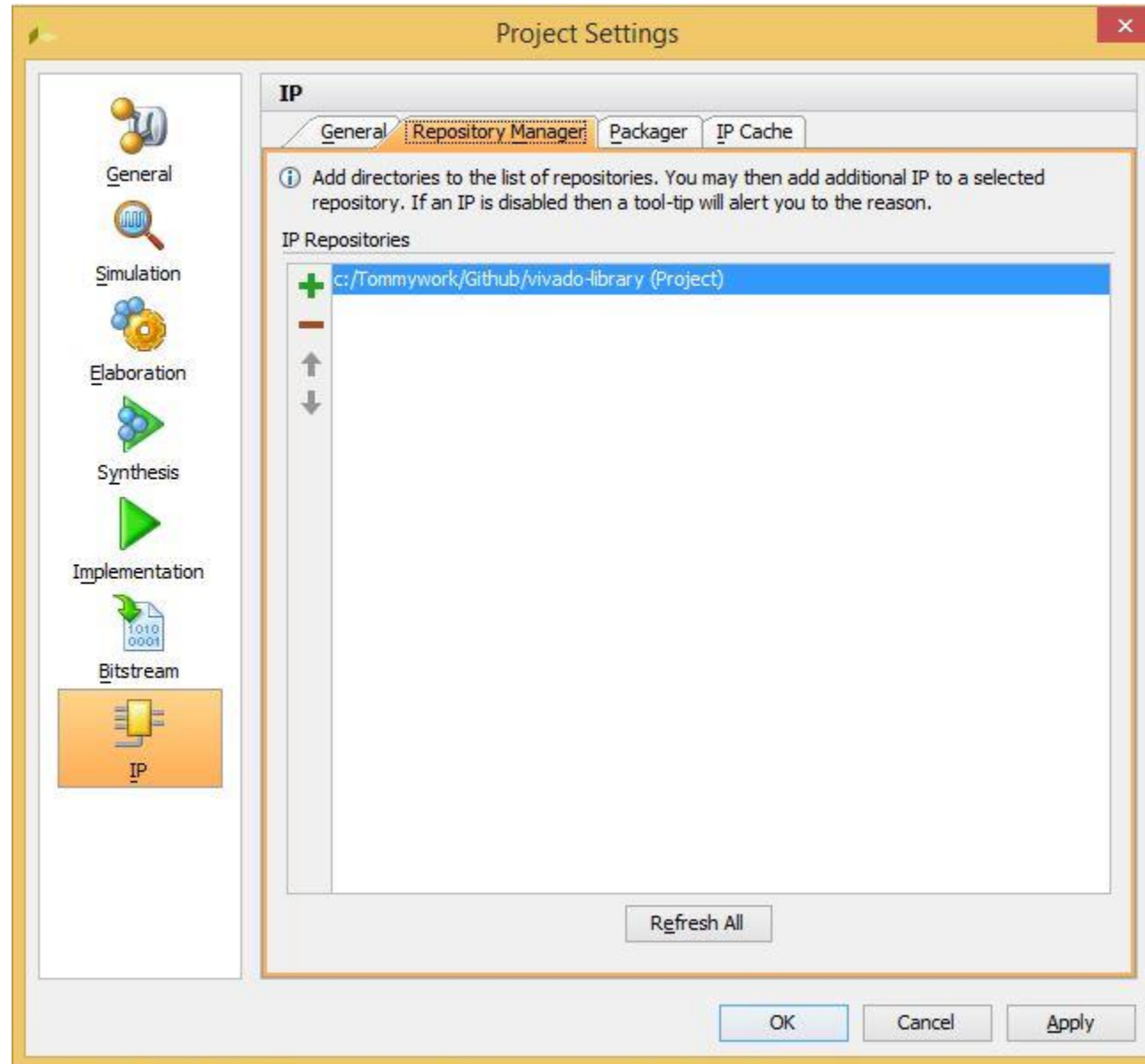
2.2) Project Manager 에서 Project Settings 를 클릭하라.



2.3) IP 를 클릭하고 Repository Manager 탭을 연다.

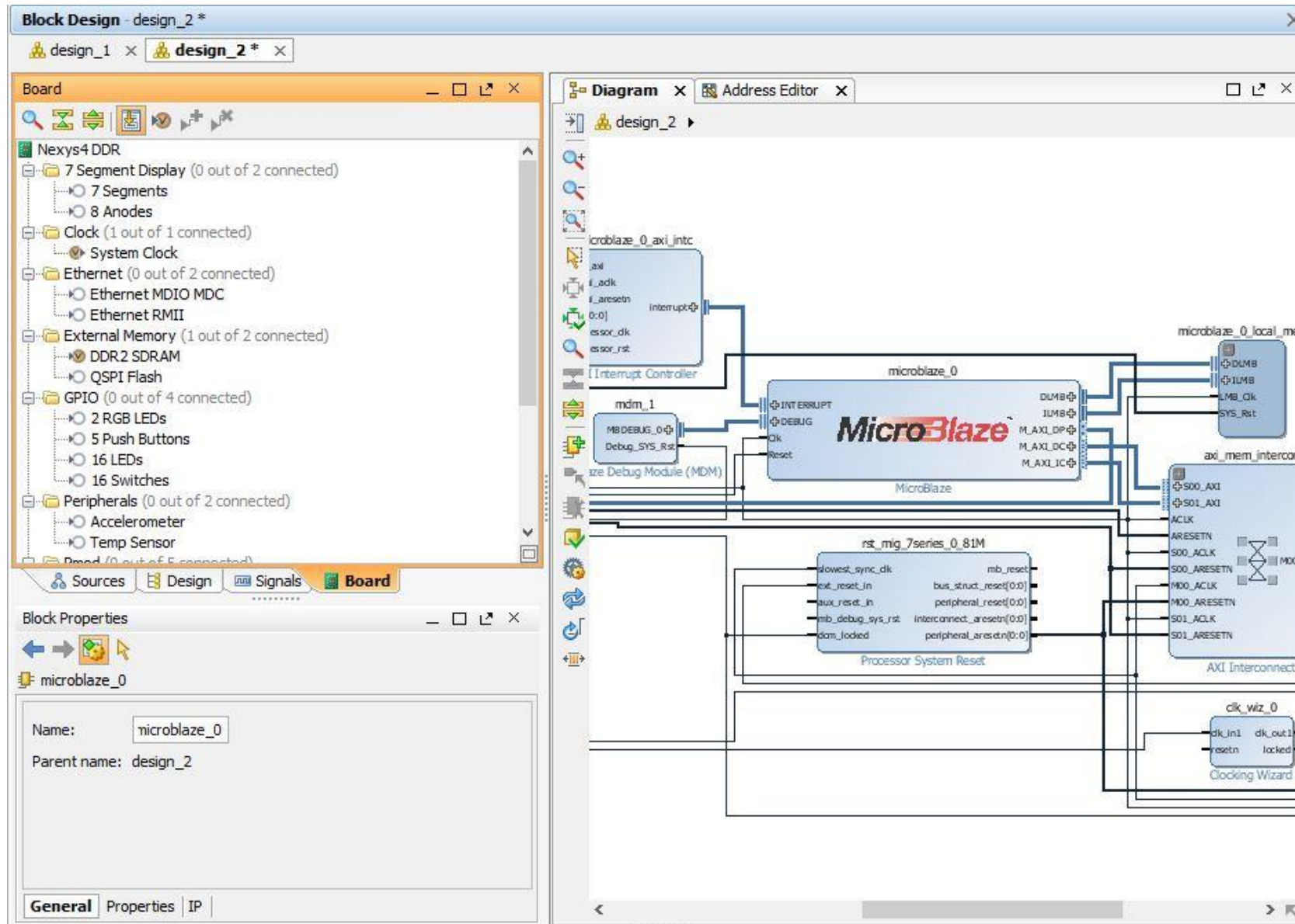
Add 버튼을 클릭하고 ZIP 아카이브를 압축 해제한 위치의 vivado-library 폴더를 선택한다.

그리고 OK 를 클릭하나.



3. Block Design 에 Pmod 추가

3.1) Board 탭을 클릭한다(아래의 주황색으로 강조 표시)



Info

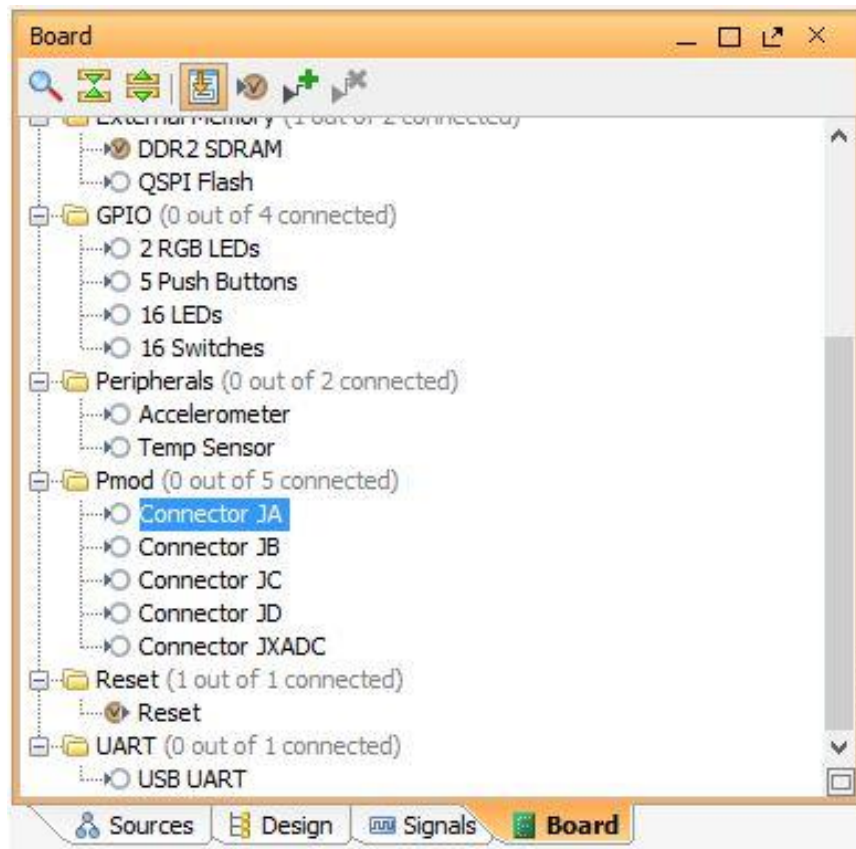
이 목록에는 플랫폼의 보드 파일에 정의된 모든 구성 요소가 들어 있다.

이더넷 포트 또는 범용 LED 와 같이 플랫폼에 있는 HW 와 함께 작동할 IP 블록을 쉽게 삽입할 수 있다.

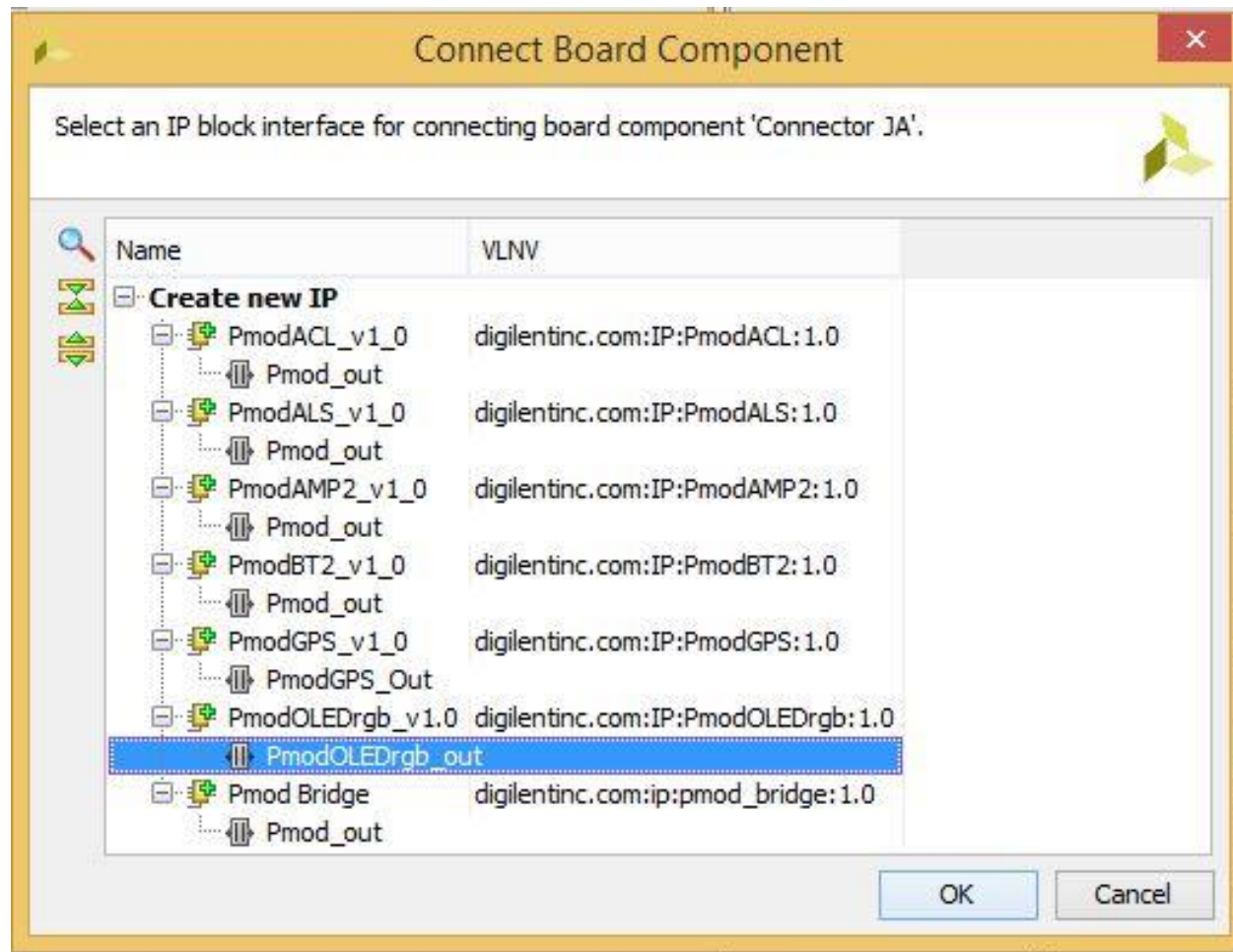
이 중 몇 개는 1.1 단계에서 초기 설계를 만들 때 이미 선택되어 있어야 한다.

3.2) 보드 구성 요소의 Pmod 섹션까지 아래로 스크롤 한다.

설정하려는 커넥터를 더블 클릭한다.



- 3.3) 특정 Pmod 에 대한 Pmod IP 를 선택하고 OK 를 클릭한다.
아래 사진은 선택된 PmodOLEDrgb 에 대한 IP 코어를 보여준다.



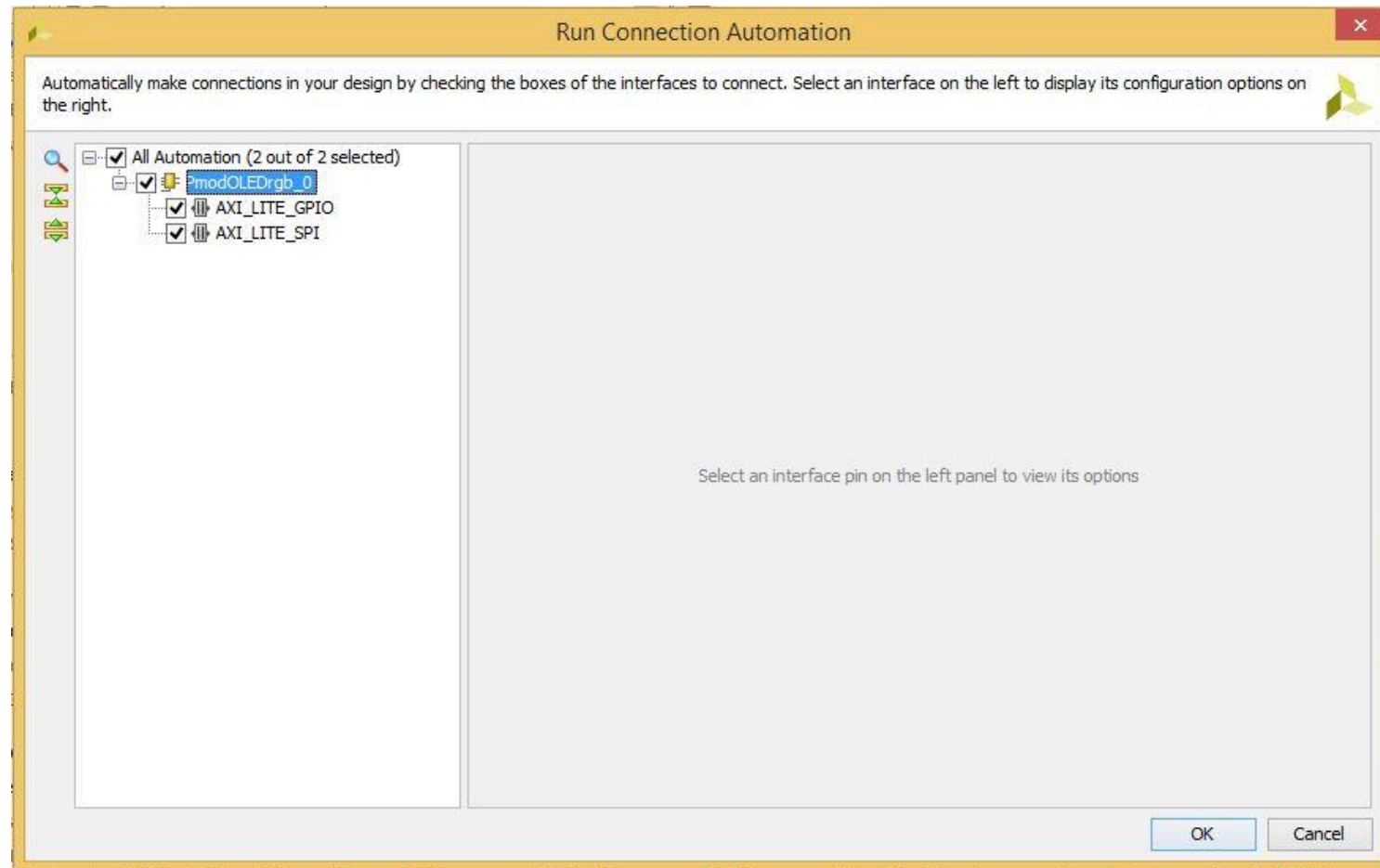
Tip

더 간단한 GPIO Pmod 중 일부는 PmodGPIO IP 코어와 함께 사용할 수 있다.

이 IP 코어에서 Pmod 가 지원되는지 확인하려면 이 튜토리얼의 Overview 에 있는 Pmod 호환성 표를 참조하라.

4. Run Connection Automation

4.1) Run Connection Automation 을 클릭한 다음 Pmod IP 코어의 이름 옆에 있는 상자를 선택하고 OK 를 클릭한다.



5. Connect Reference Clocks

* Important

일부 Pmod IP 코어가 제대로 작동하려면 Reference Clock 이 필요하다.

Pmod 에 Reference Clock 이 필요한지 확인하려면 Overview 에 있는 Pmod Compatibility 표를 참조하라.

Pmod 에 Reference Clock 이 필요하지 않으면 6 단계로 넘어간다.

Pmod IP 코어에 Reference Clock 을 연결하는 것은 사용중인 플랫폼에 따라 다르다.

플랫폼을 가장 잘 설명하는 탭을 선택하라.

(모르는 경우 Overview 의 Platform Compatibility 표를 참조하라)

5.1) ZYNQ Processing System 블록을 더블 클릭하여 Re-Customize 를 수행한다.

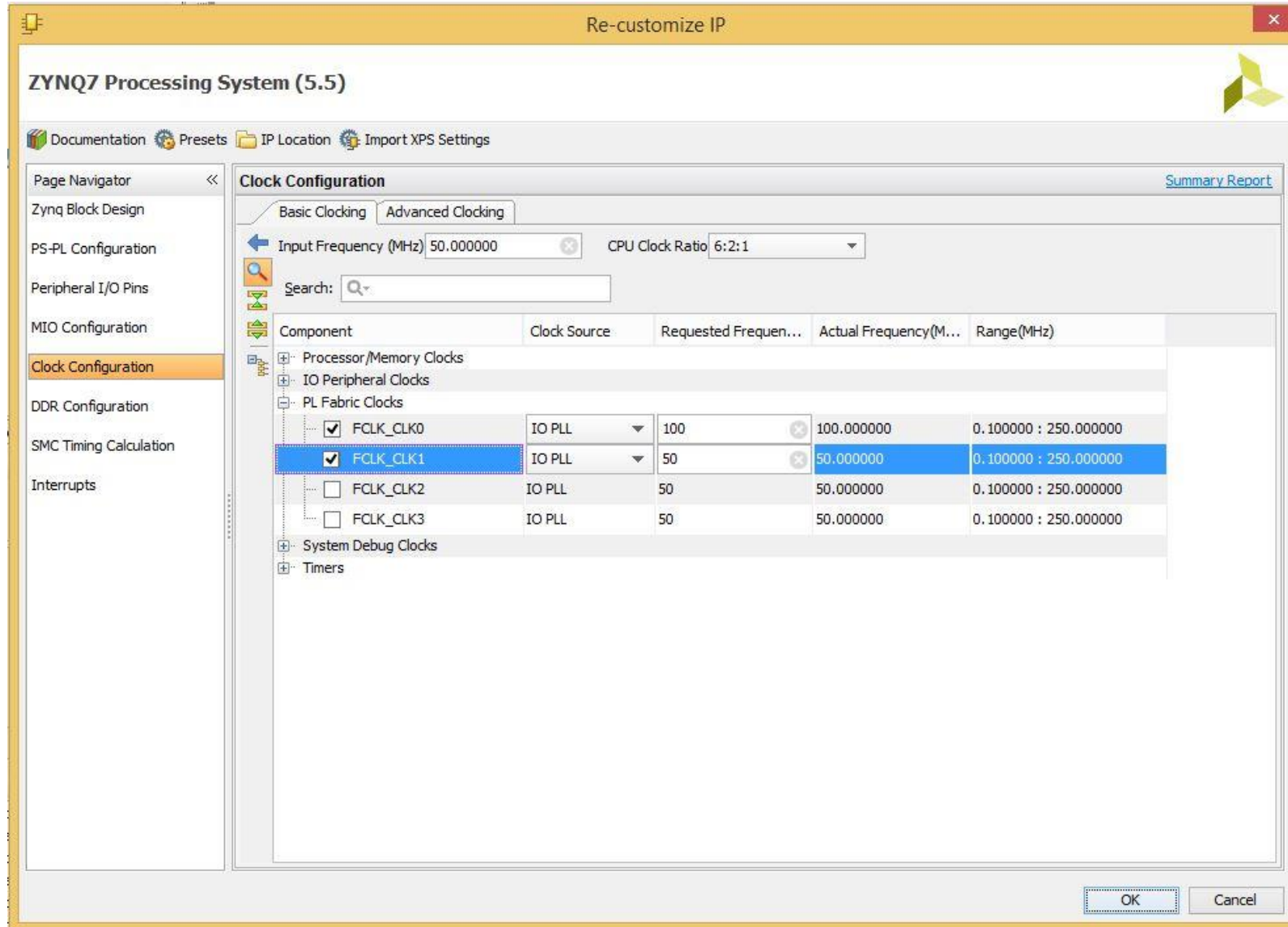
왼쪽의 메뉴에서 Clock Configuration 을 클릭한다.

PL Fabric Clock 을 Drop Down 하고 아직 활성화되지 않은 첫 번째 FCLK_CLK 를 Check 한다.

요청한 주파수를 Pmod 에 필요한 주파수로 설정한다.

이 주파수를 튜토리얼의 Overview 에 있는 Pmod Compatibility 표에서 확인할 수 있다.

OK 를 클릭한다.

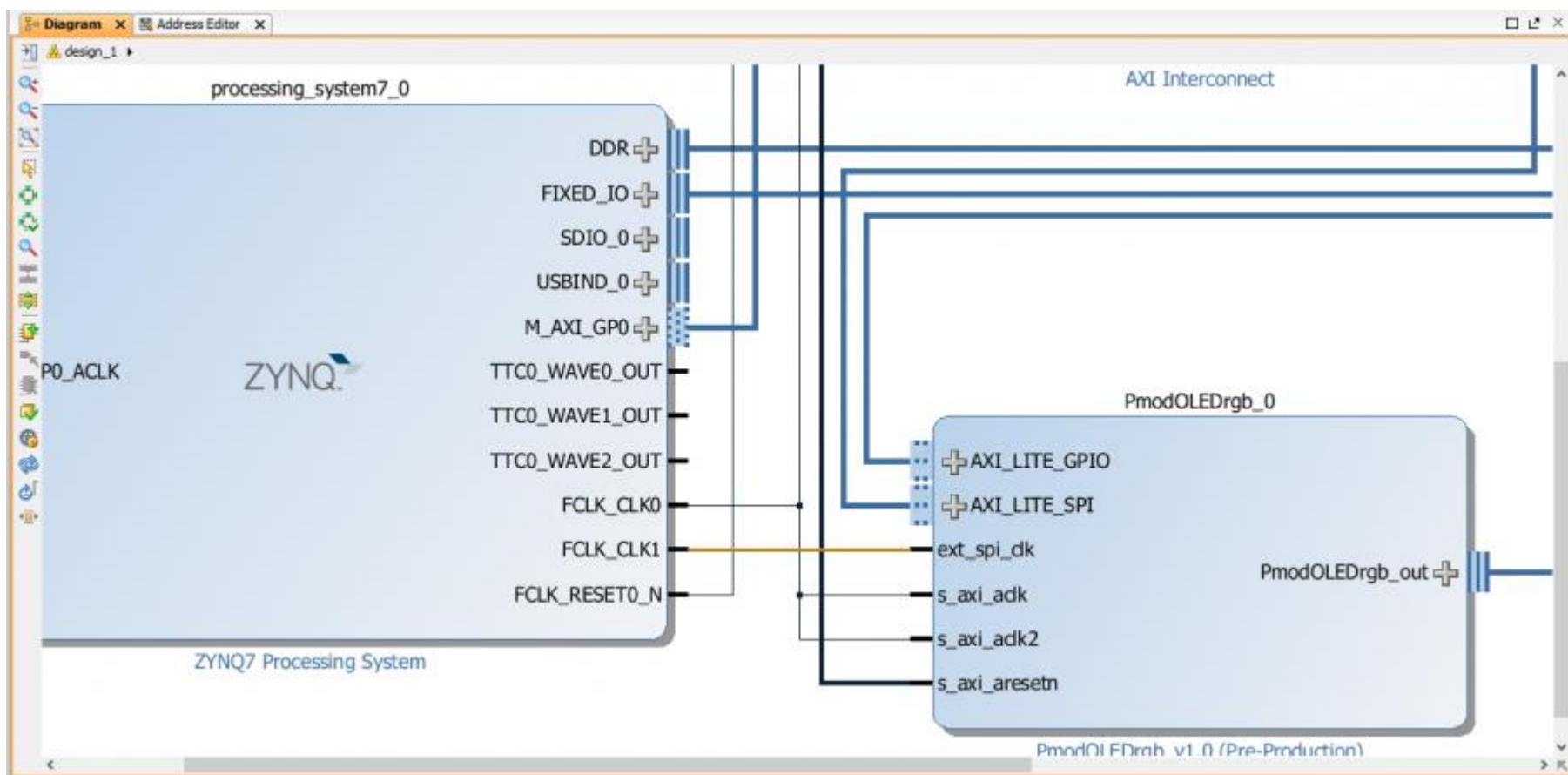


Tip

이미 체크된 FCLK_CLK 중 하나가 pmod 에 필요한 주파수와 일치하는 주파수를 가지고 있다면 이 클록을 사용할 수 있다.
단일 클록을 여러 대상에 연결할 수 있다.

5.2) 이 새로운 클럭을 Pmod IP 코어의 클럭 입력에 연결한다.

Pmod IP 코어의 클럭 입력 이름은 이 튜토리얼의 Overview 에 있는 Pmod Compatibility 표에서 확인할 수 있다.



6. Connect Interrupts

* Important

일부 Pmod IP 코어는 제대로 작동하려면 인터럽트가 필요하다.

Pmod 에 인터럽트가 필요한지 확인하려면 Overview 에 있는 Pmod Compatibility 표를 참조하라.

Pmod 에 인터럽트가 필요하지 않으면 7 단계로 이동한다.

Pmod IP 코어 인터럽트를 프로세서에 연결하는 것은 사용중인 플랫폼에 따라 다르다.

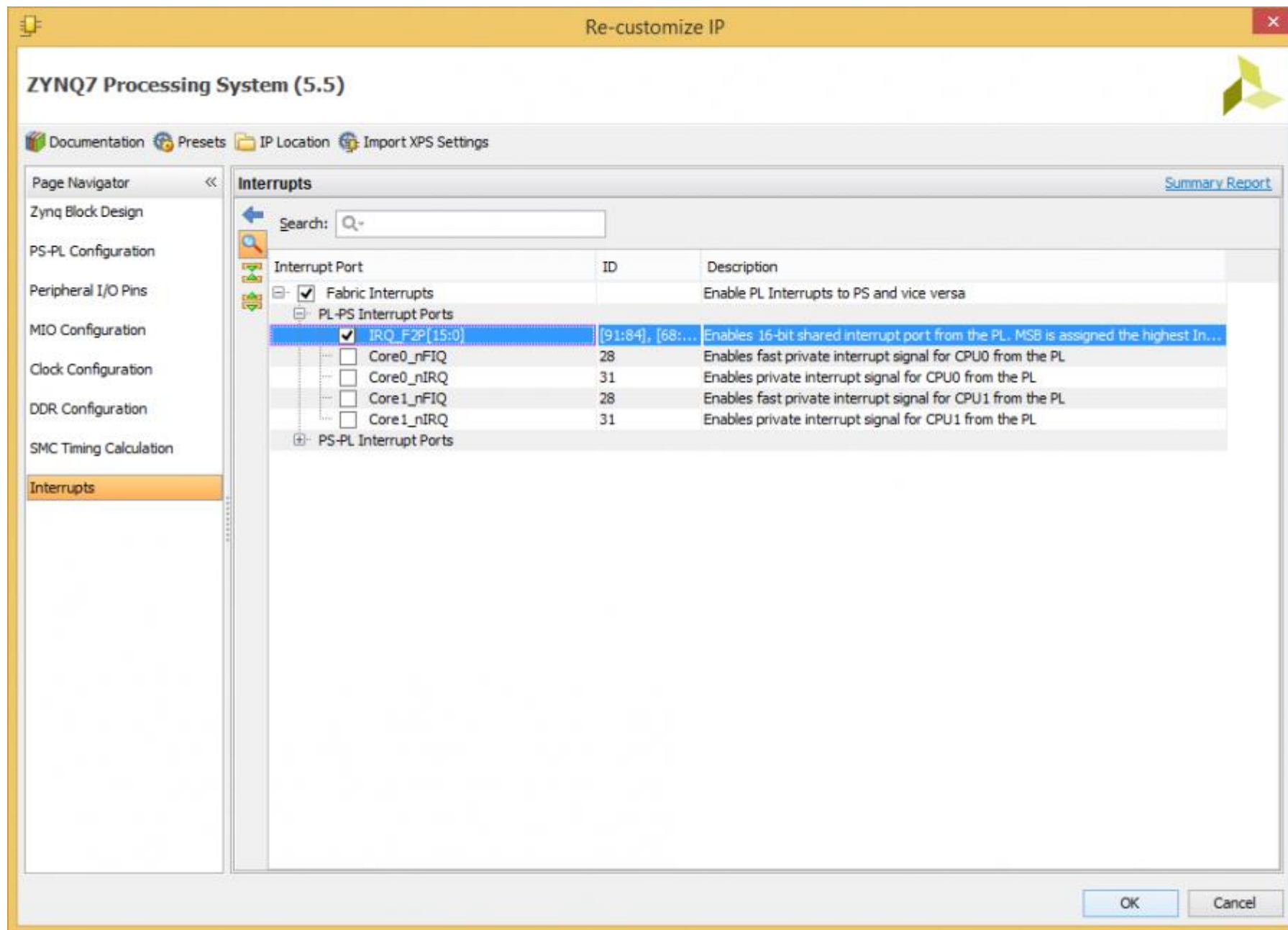
플랫폼을 가장 잘 설명하는 탭을 선택하라.

(모르는 경우 Overview 에 Platform Compatibility 표를 참조하라)

6.1) IRQ_F2P 라는 이름의 포트가 Zynq Processing System 블록에 없다면 블록을 더블 클릭하여 Re-Customize 한다.

왼쪽 메뉴에서 Interrupts(인터럽트)를 클릭한다.

Fabric Interrupts 를 확인하고 확장 한 다음 PL-PS Interrupts 를 확장하고 IRQ_F2P 옆의 상자를 선택한다.

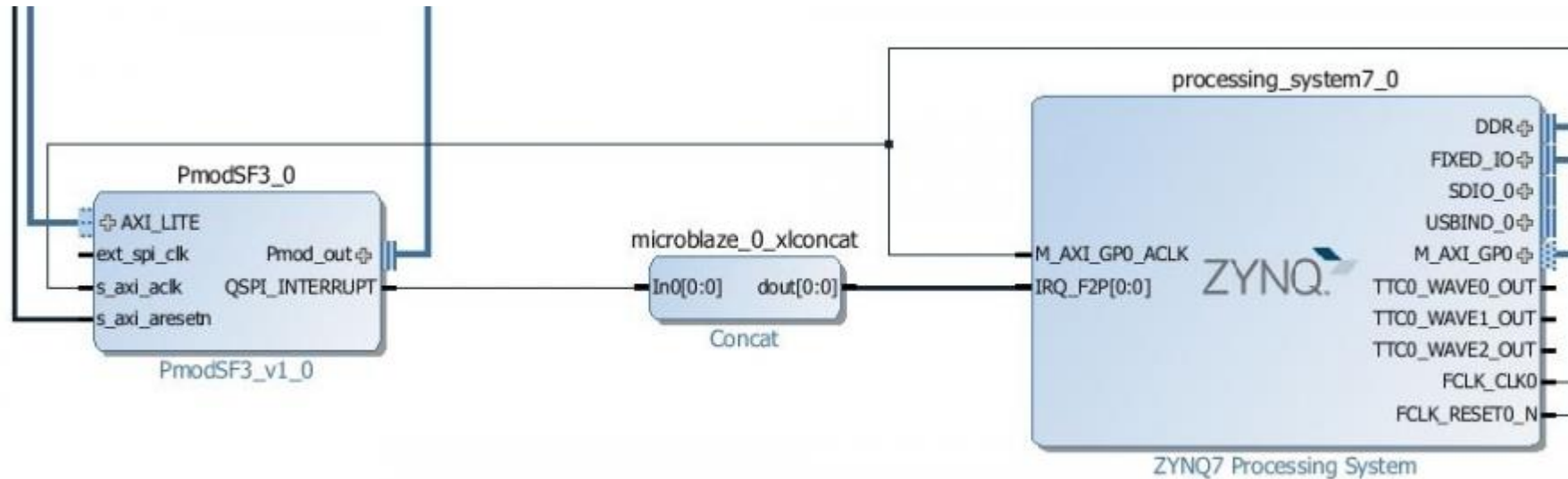


6.2) Block Desing 에 Concat IP 코어를 추가한다.

concat 블록을 Re-Customize 하여 입력 수가 Zynq 프로세서에 연결하는데 필요한 인터럽트 수와 일치하는지 확인한다.
OK 를 클릭한다.

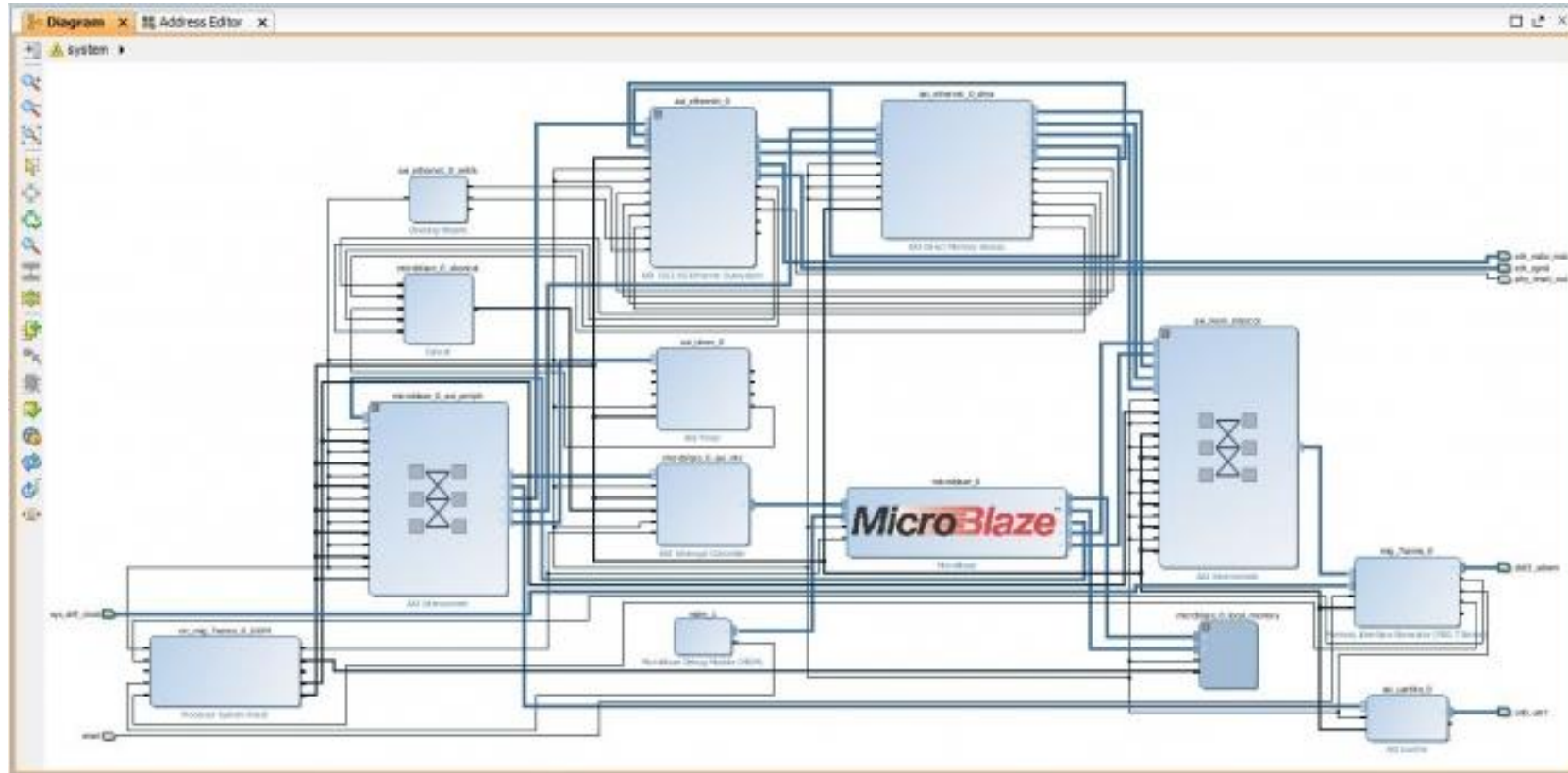
6.3) Pmod IP 코어의 인터럽트 포트를 concat 블록의 입력 포트에 연결하고

concat 의 출력 포트를 Zynq Processing SYstem 의 IRQ_F2P 포트에 연결한다.



7. Validate the Design

7.1) Regenerate Layout 버튼을 클릭하여 Block Design 을 재정렬한다.



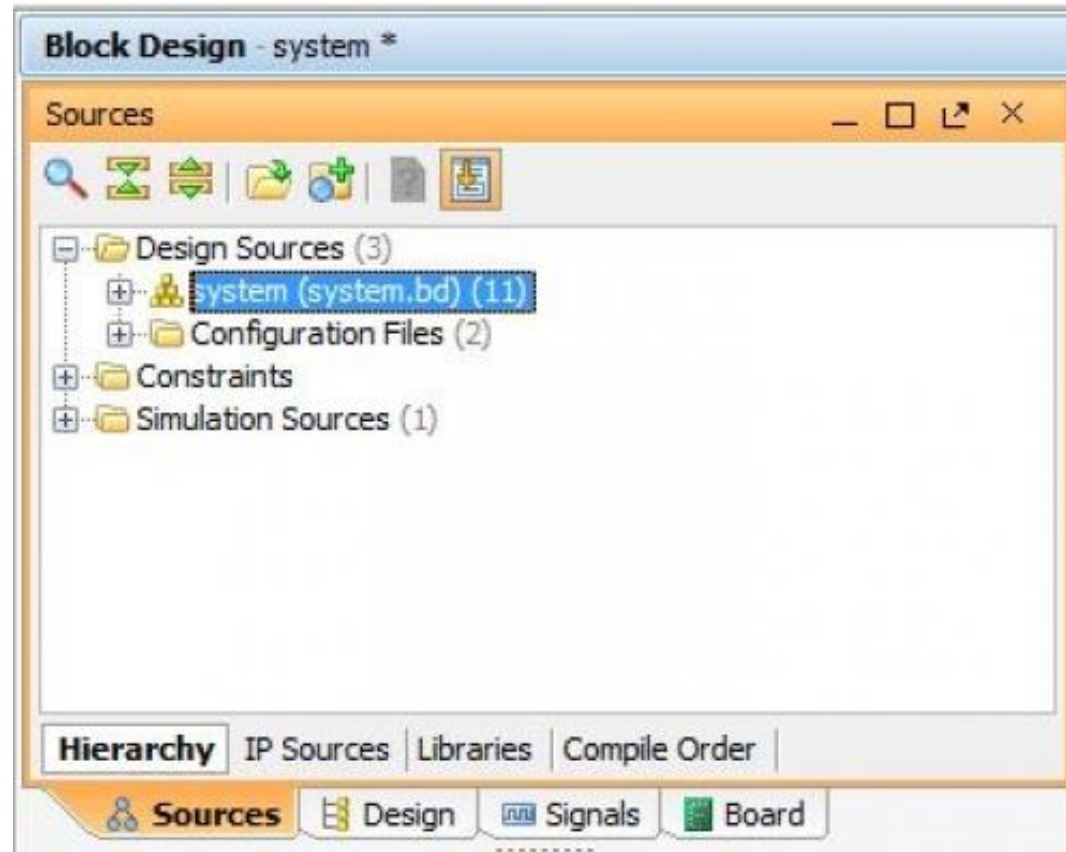
7.2) Validate Design 을 선택한다.

이렇게 Design 및 Connection 오류를 검사한다.

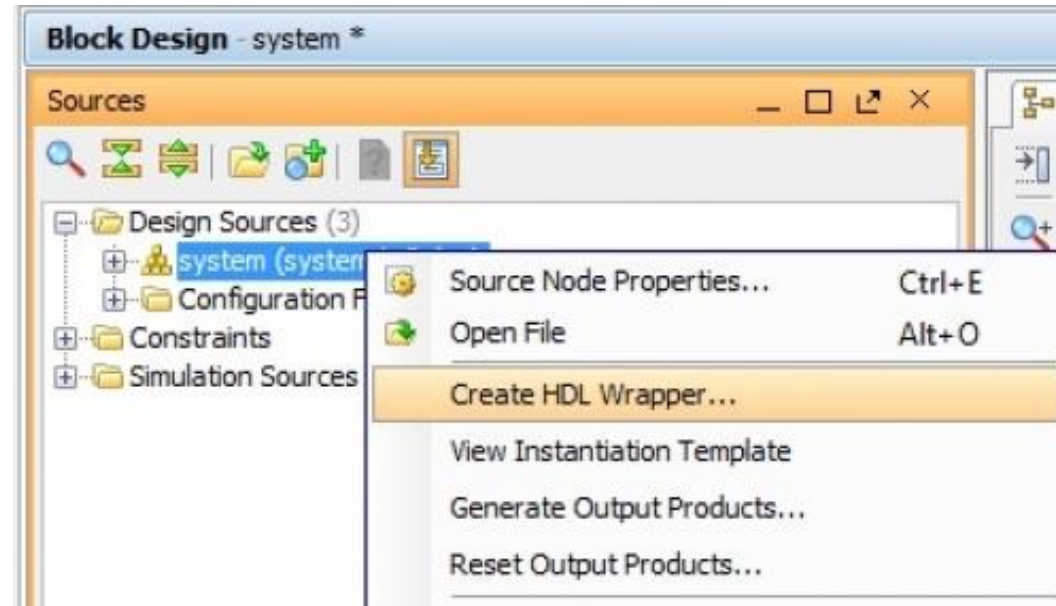
7.3) Board 별 'Getting Started With ...' 튜토리얼의 일부로 Block Design 을 위한 HDL Wrapper 를 이미 만든 경우 이 단계의 나머지 부분을 건너뛴다.

그렇지 않으면 Design Validation 단계 후에 HDL System Wrapper 만들기를 진행한다.

Source 탭을 클릭하고 Block Design 을 찾는다.



- 7.4) Block Design 을 우클릭하고 Create HDL Wrapper 를 클릭한다.
Vivado 가 Wrapper 를 관리하고 자동으로 업데이트하고 OK 를 클릭한다.



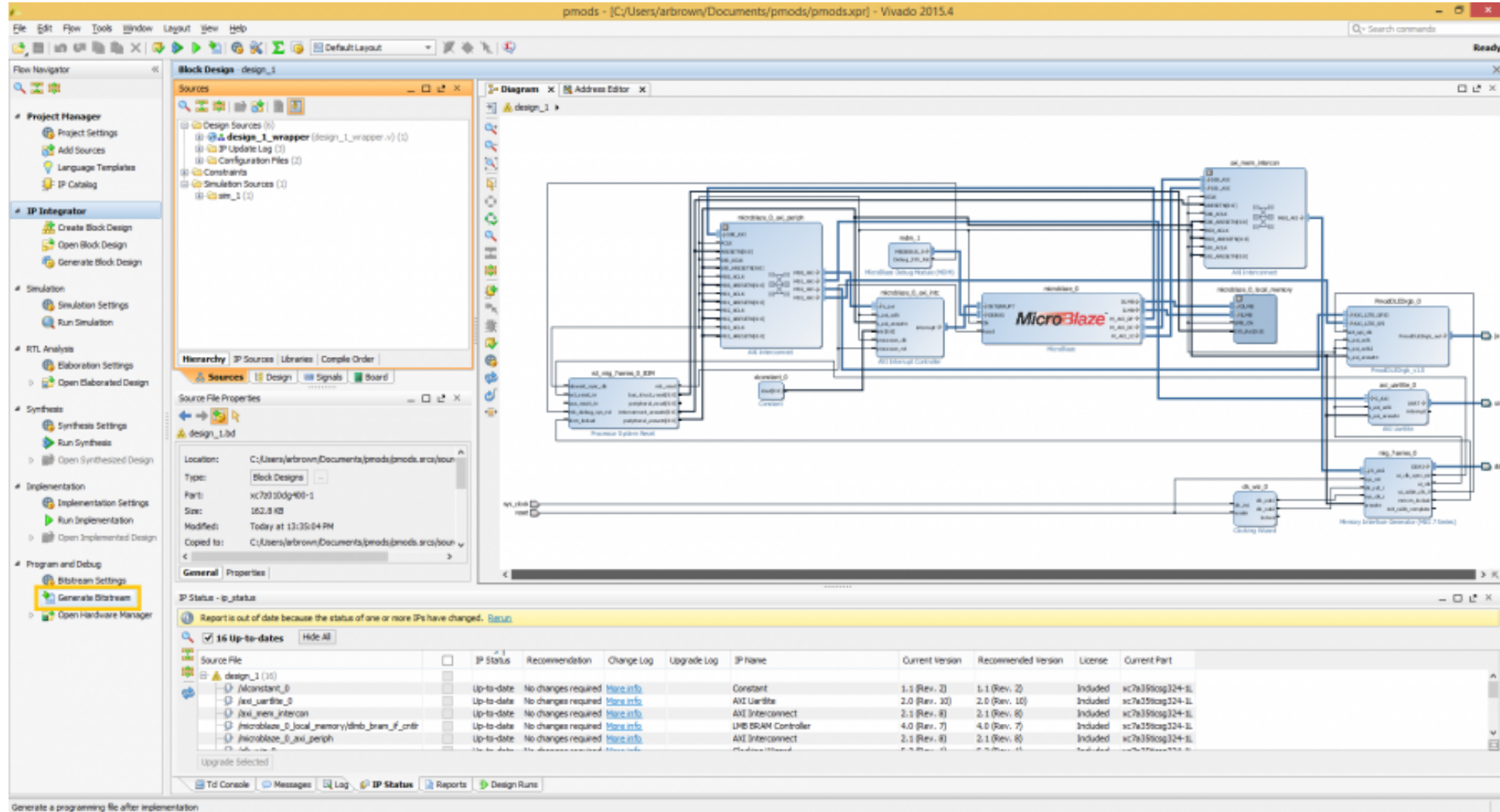
이렇게하면 VHDL 에 TOP Module 이 생성되어 bitstream 을 생성 할 수 있다.

8. Generate the Bit File

8.1) Vivado 의 상단 톨바에서 Generate Bitstream 을 클릭한다.

Flow Navigator 패널의 왼쪽 아래에 Program and Debug 를 찾을 수 있다.

아직 Design 을 저장하지 않았다면 Block Design 을 저장하라는 메시지가 나타난다.



8.2) 비트 파일 생성이 시작된다.

이 도구는 Synthesis and Implementation 을 실행한다.

두 가지 모두 성공적으로 완료되면 비트 파일이 생성된다.

프로젝트 창의 오른쪽 상단 모서리에 Synthesis 및 Implementation 상태 표시 줄이 있다.

이 절차는 컴퓨터 및 Target Board 에 따라 5 분 ~ 60 분 정도 소요될 수 있다.

8.3) bitstream 이 생성 된 후 화면에 Message Prompt 가 Pop-Up 된다.

이 데모에서는 구현 된 Design 을 열 필요가 없다.

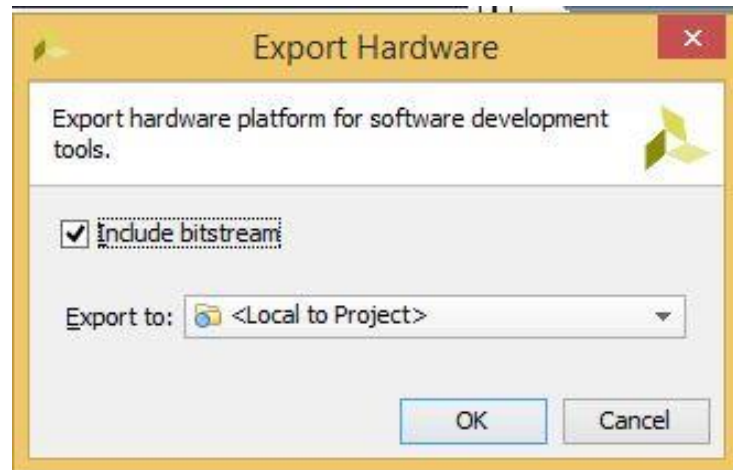
단순히 Cancel 을 클릭하면 된다.

9. Export the Hardware Design to SDK

9.1) 기본 도구 모음에서 파일을 클릭하고 Export Hardware 를 선택하라.

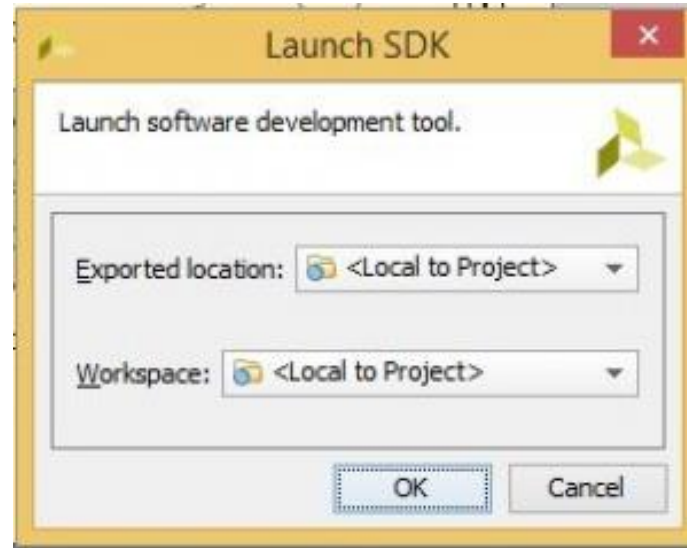
Include Bitstream 에 대한 박스를 체크하고 OK 를 클릭한다.

그러면 SW 개발 도구인 Xilinx SDK 용 System Wrapper 를 사용하여 HW Design 을 Export 한다.



Vivado HW Design 프로젝트 이름과 비슷한 새 디렉토리를 echo_server.SDK 아래에 프로젝트 디렉토리에 생성될 것이다.
두 개의 다른 파일인 .sysdef 와 .hdf 도 작성된다.
이 단계는 기본적으로 새로운 SDK 작업 영역을 만든다.

9.2) Main Toolbar 에서 파일을 클릭하고 SDK 를 띄운다.
두 개의 Dropdown 메뉴를 기본 프로젝트로 로컬로 두고 OK 를 클릭한다.
그러면 Xilinx SDK 가 열리고 HW 를 가져올 것이다.



10. Tour Xilinx SDK

HW Design 사양 및 포함된 IP Block 은 system.hdf 파일에 표시된다.

Xilinx SDK 는 Vivado 와 독립적이다.

즉 이 시점부터 Export 된 HW Design 에 C/C++ 로 SW 프로젝트를 만들 수 있다.

필요한 경우 기본 Vivado 프로젝트 디렉토리에서 생성된 .SDK 폴더를 작업 영역으로 사용하여 SDK 를 직접 실행할 수도 있다.

이 시점에서 Vivado 로 돌아가서 HW Design 을 변경해야 하는 경우 SDK 창을 닫고 Vivado 에서 필요한 HW Design 을 수정하는 것이 좋다.

그런 다음 Vivado 가 HDL Wrapper 를 재생성하고 새로운 비트 파일을 생성 할 수 있도록 Design 저장 순서를 따라야 한다.

이 새로운 비트 파일은 SDK 로 Export 되어야 한다.

왼쪽의 프로젝트 탐색기 탭에서 HW 플랫폼을 볼 수 있다.

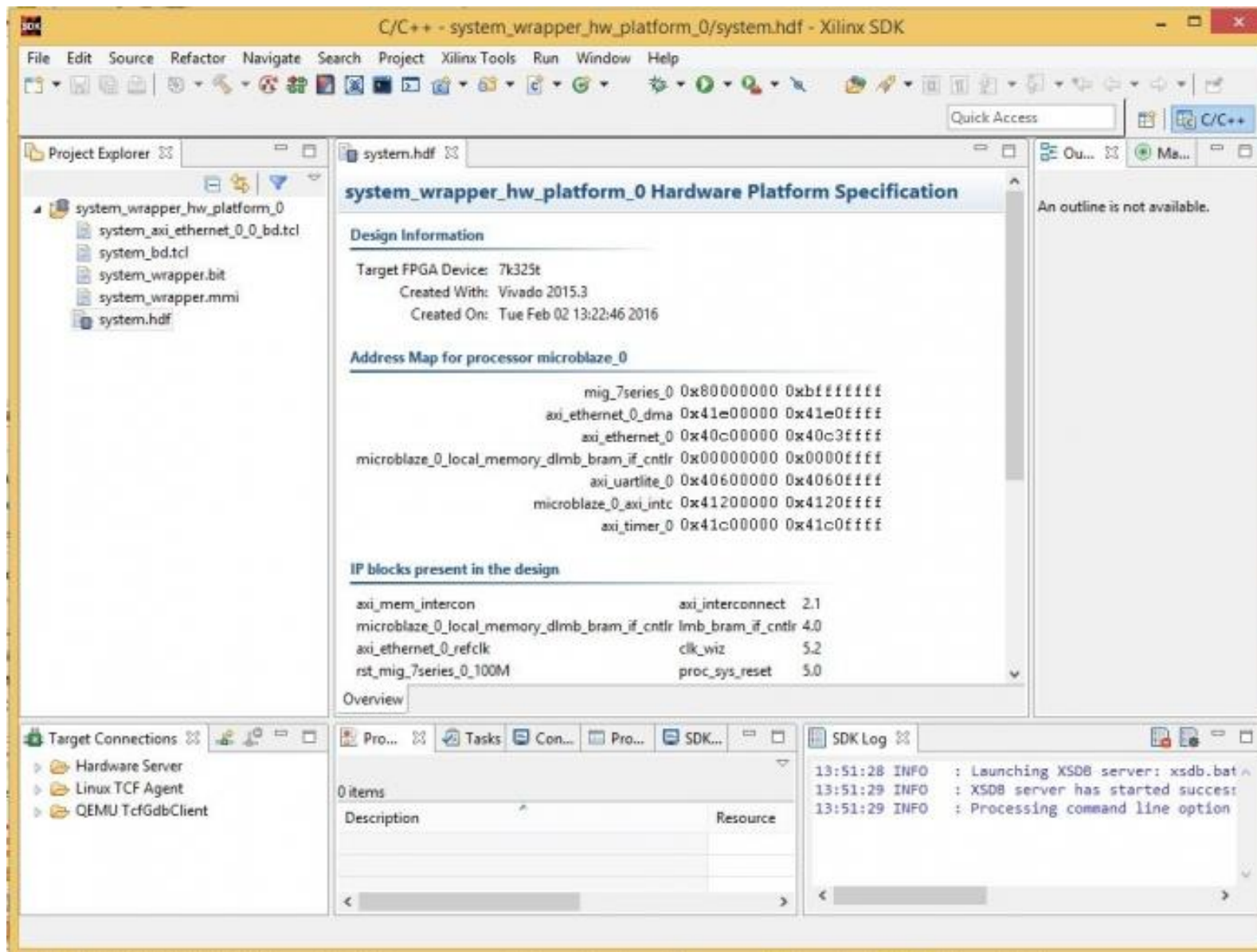
system 은 Vivado 에서 만든 Block Design 의 이름이다.

이 HW 플랫폼에는 모든 HW Design 정의, 추가 된 IP 인터페이스, 외부 출력 신호 정보 및 로컬 메모리 주소 정보가 있다.

Pmod IP 장치 전용 드라이버는 /drivers 아래의 HW 플랫폼에 있는 해당 폴더에서 찾을 수 있다.

이 드라이버를 편집하려면 libsrc 의 보드 지원 패키지에 있는 버전을 사용하도록 한다.

드라이버를 수정하는 경우 HW 를 변경하면 이러한 변경 사항과 BSP 소스 재생성을 모두 덮어 쓴다.



11. Create a New Application Project in SDK

11.1) New dropdown 화살표를 클릭하고 Application Project 를 선택한다.

New Project

Application Project
Create a managed make application project.

Project name: Pmods

☒ Use default location

Location: C:\Tommywork\Vivado\15_4\b\b.sdk\Pmods Browse...

Choose file system: default

OS Platform: standalone

Target Hardware

Hardware Platform: design_1_wrapper_hw_platform_0 New...

Processor: microblaze_0

Target Software

Language: ☒ C ☐ C++

Compiler: 32-bit

Board Support Package: ☒ Create New Pmods_bsp ☐ Use existing pmodgps_bsp

? < Back Next > Finish Cancel

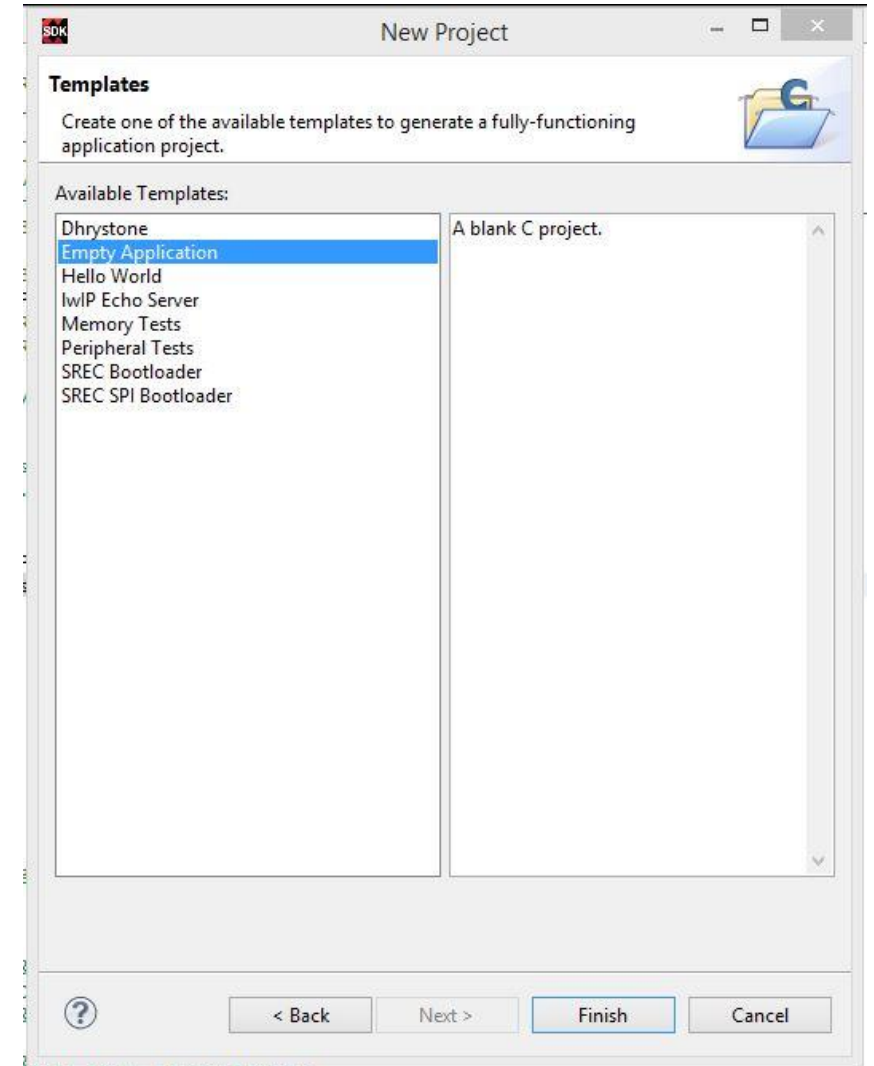
프로젝트에 공백이 없는 이름을 지정하고 Next 를 클릭한다.

11.2) 템플릿 목록에서 Empty Application 을 선택하고 OK 를 클릭한다.

프로젝트 탐색기 패널에 두 개의 새 폴더가 표시된다.

- 모든 바이너리, *.c 및 *.h 파일을 포함하는 응용 프로그램 프로젝트
- 프로젝트에 포함 될 수 있는 드라이버 소스 파일이 포함 된 프로젝트 전용 보드 지원 패키지

우리의 중요 작업 소스 폴더에는 "Iscrip.ld" 라는 중요한 파일이 들어 있다.
이것은 Xilinx 자동 생성 링커 스크립트 파일이며 다른 메모리 영역의 크기뿐만 아니라 Block Design 의 다른 IP 구성 요소에 대한 메모리 주소에 관한 정보를 포함한다.

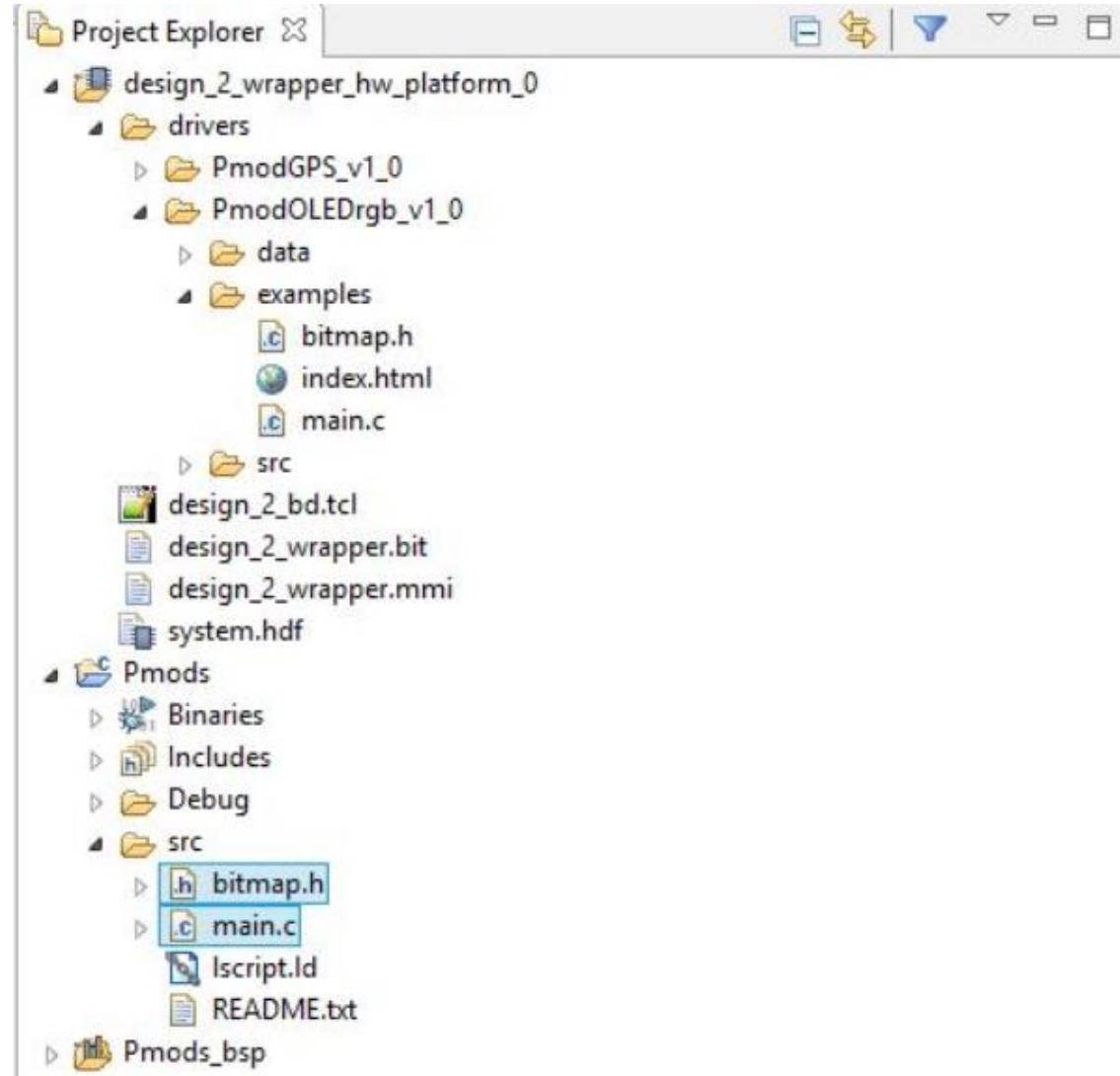


12. Import the Example Project

design_1_wrapper_platform_0 폴더를 확장한다.

drivers 폴더에는 Design 에 사용중인 Pmod 목록이 있다.

Pmod.../examples 폴더를 펼치고 여기에 있는 모든 파일을 project_name/src 폴더로 복사한다.



13. Program the FPGA with the Bit File

13.1) Board 가 커져 있고 UART 및 프로그래밍을 위해 micro USB 케이블로 host PC 에 연결되어 있는지 확인한다.

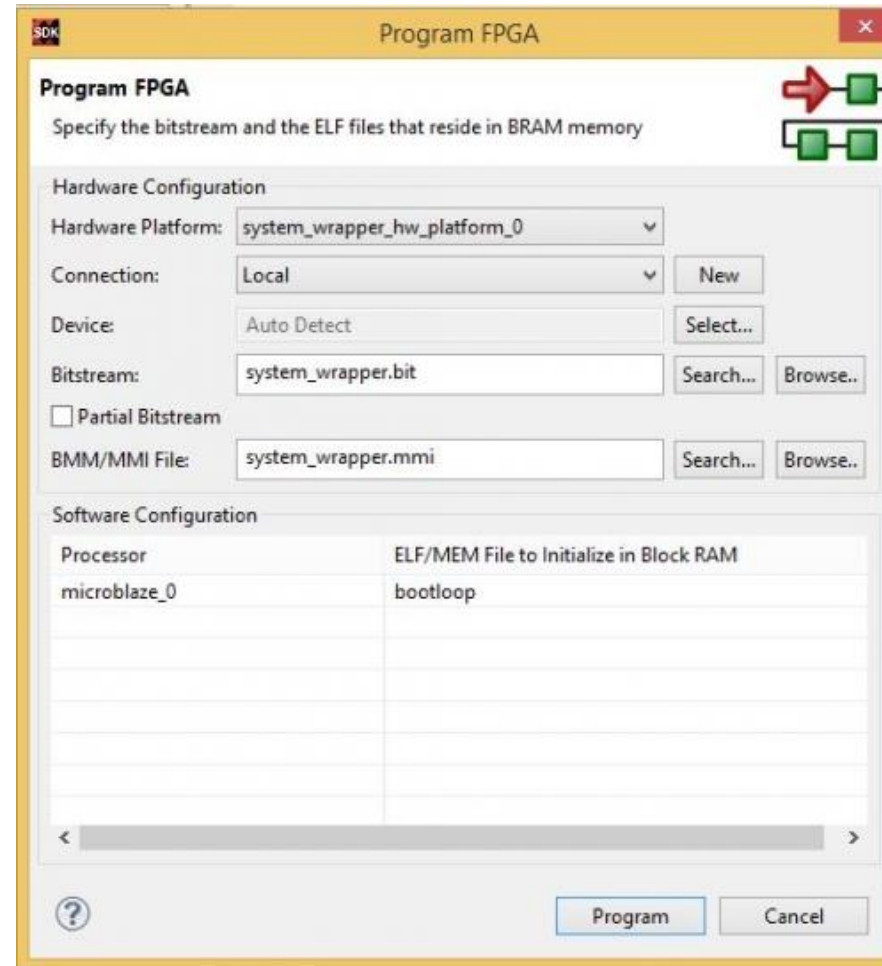
일부 Board 에서는 하나의 PROG/UART 포트를 연결하면 되고

다른 포트에서는 일반적으로 UART 및 PROG 또는 JTAG 라는 두 개의 다른 포트에 PC 를 연결해야한다.

상단 툴바에서 프로그램 FPGA 버튼을 클릭하라.

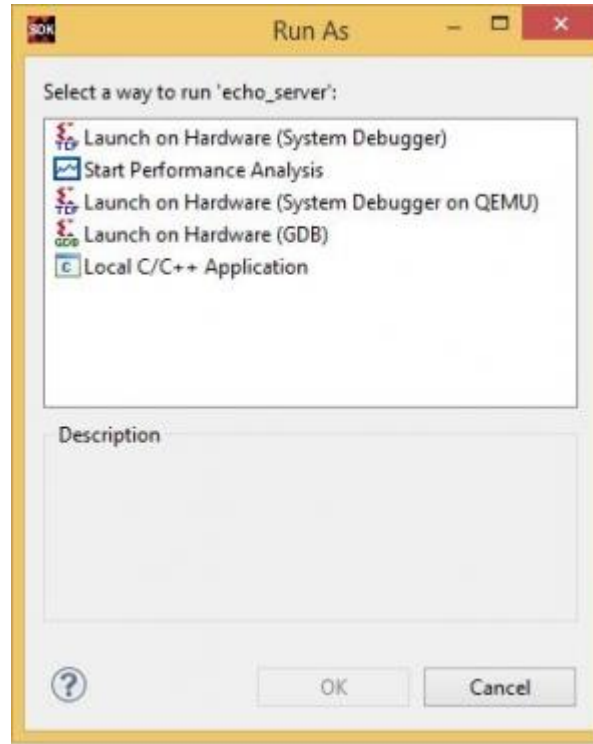
일부 Board 는 별도의 전원에 연결해야 할 수도 있다.

13.2) Program 을 클릭하여 HW Design 으로 FPGA 를 프로그래밍한다.



14. Program the Microblaze/ZYNQ Processor

- 14.1) 대부분의 데모에서는 데모에 의해 출력된 메시지를 읽기 위해 PC 의 직렬 터미널을 사용해야한다.
터미널에 대한 설정은 Board 에 따라 다르지만 일반적으로 Board 속도 115200 또는 9600, 8 비트 데이터, 패리티 비트 및 정지 비트 1 개를 사용해야한다.
Zynq 프로젝트는 115200 baud 를 사용하고 MicroBlaze 프로젝트의 전송 속도는 Vivado 의 Uartile IP 구성에 따라 달라진다.
- 14.2) 애플리케이션 프로젝트를 선택하고 Run As ... 버튼을 클릭한다.
Launch on Hardware(System Debugger) 를 선택하고 OK 를 클릭한다.



- 14.3) Xilinx SDK 는 Microblaze/Zynq 프로세서의 main.c 에서 시작하는 프로그램을 실행한다.
예제 main 파일 최상위에 있는 주석 헤더를 검토하여 데모가 수행하는 작업에 대한 자세한 정보와 추가 설치 요구 사항을 확인하라.

References

1. <https://reference.digilentinc.com/learn/programmable-logic/tutorials/pmod-ips/start>