TI DSP, MCU 및 Xilinx Zynq FPGA
프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

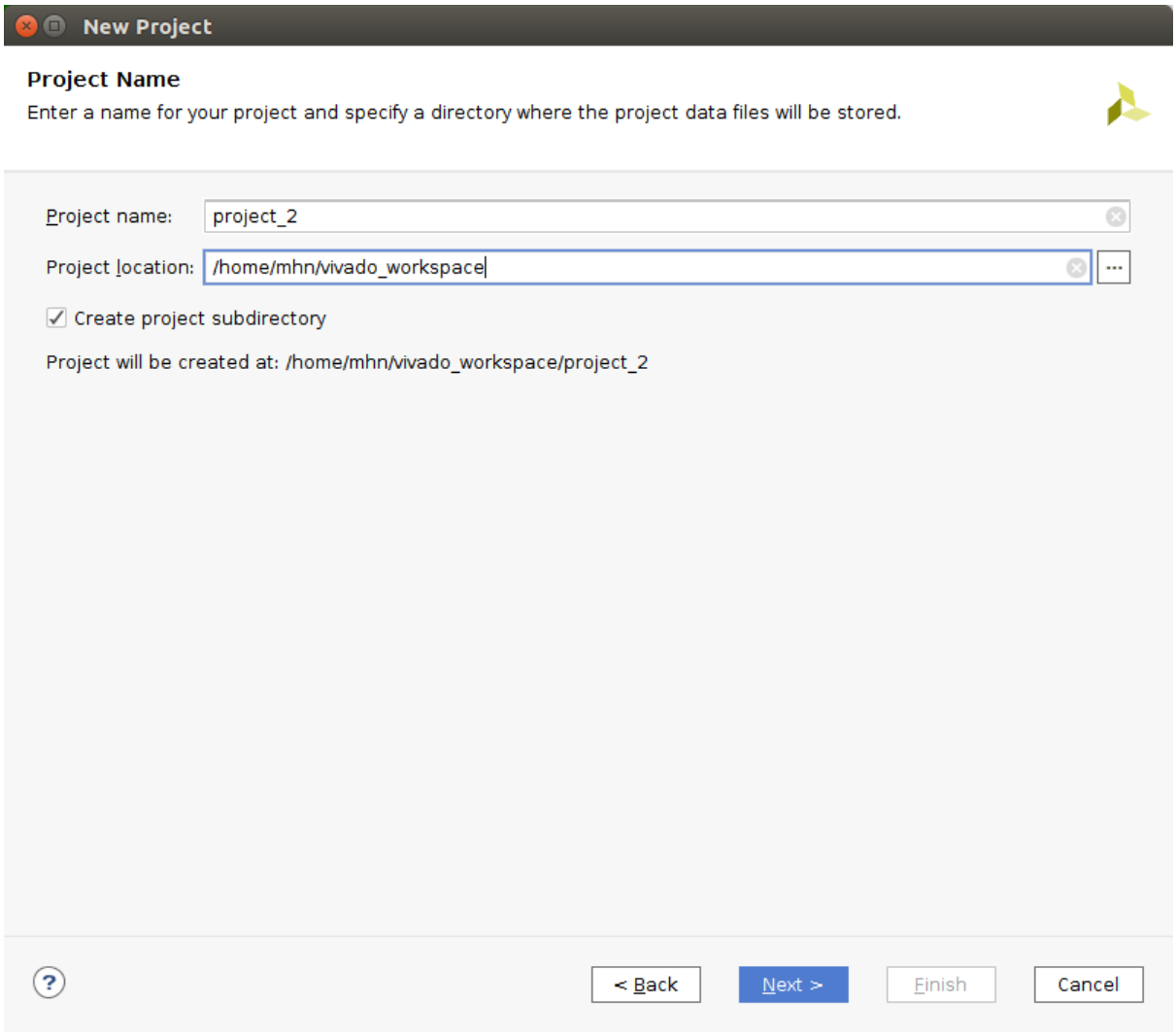
mhn97@naver.com

FPGA

fpga? 각종 트랜지스터를 논리적으로 프로그래밍한 것

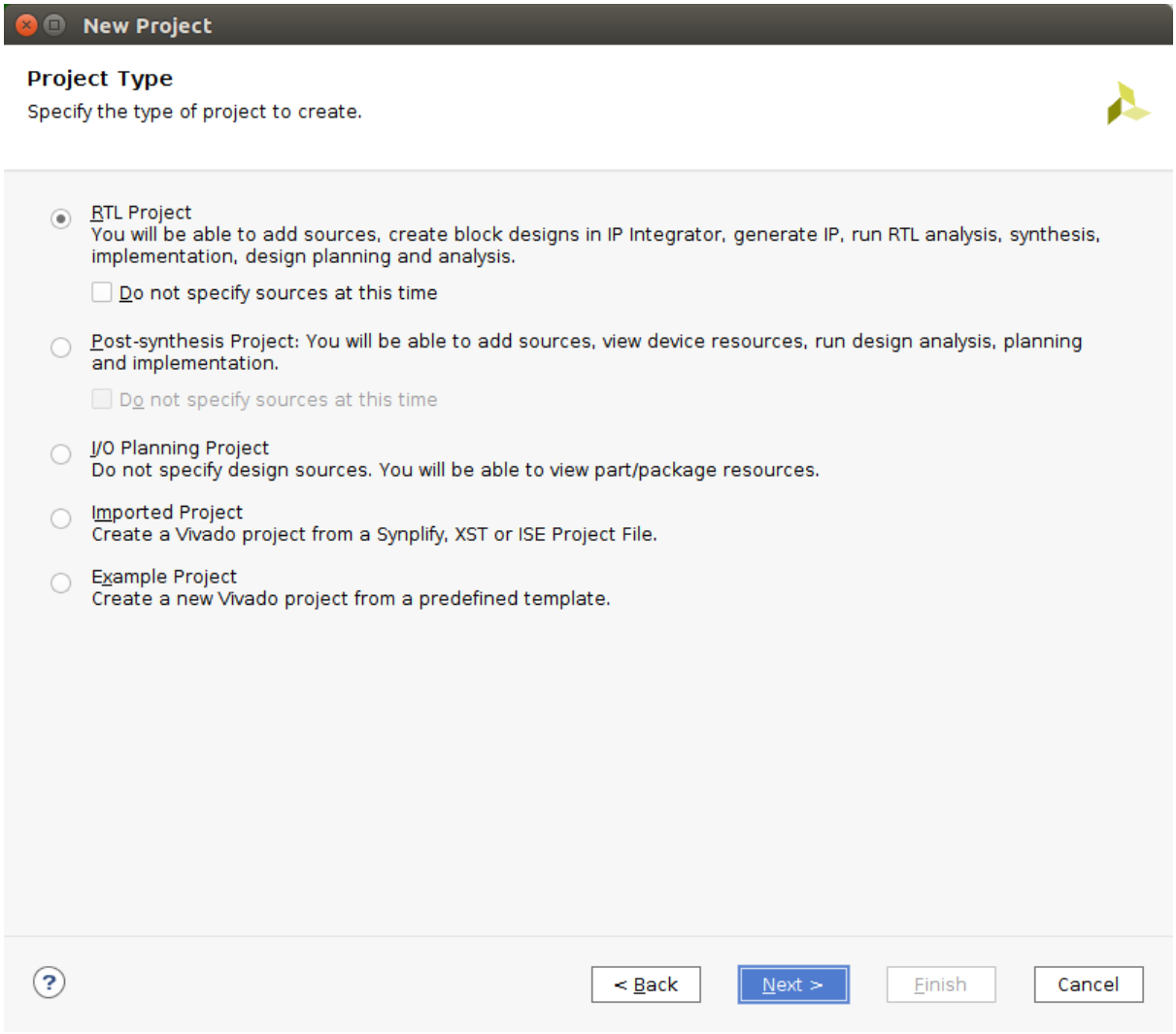
디지털 칩도 결국 내부는 아날로그이기 때문에 복잡해질수록 사람이 하기 힘듦

실습)GPIO회로 프로그래밍

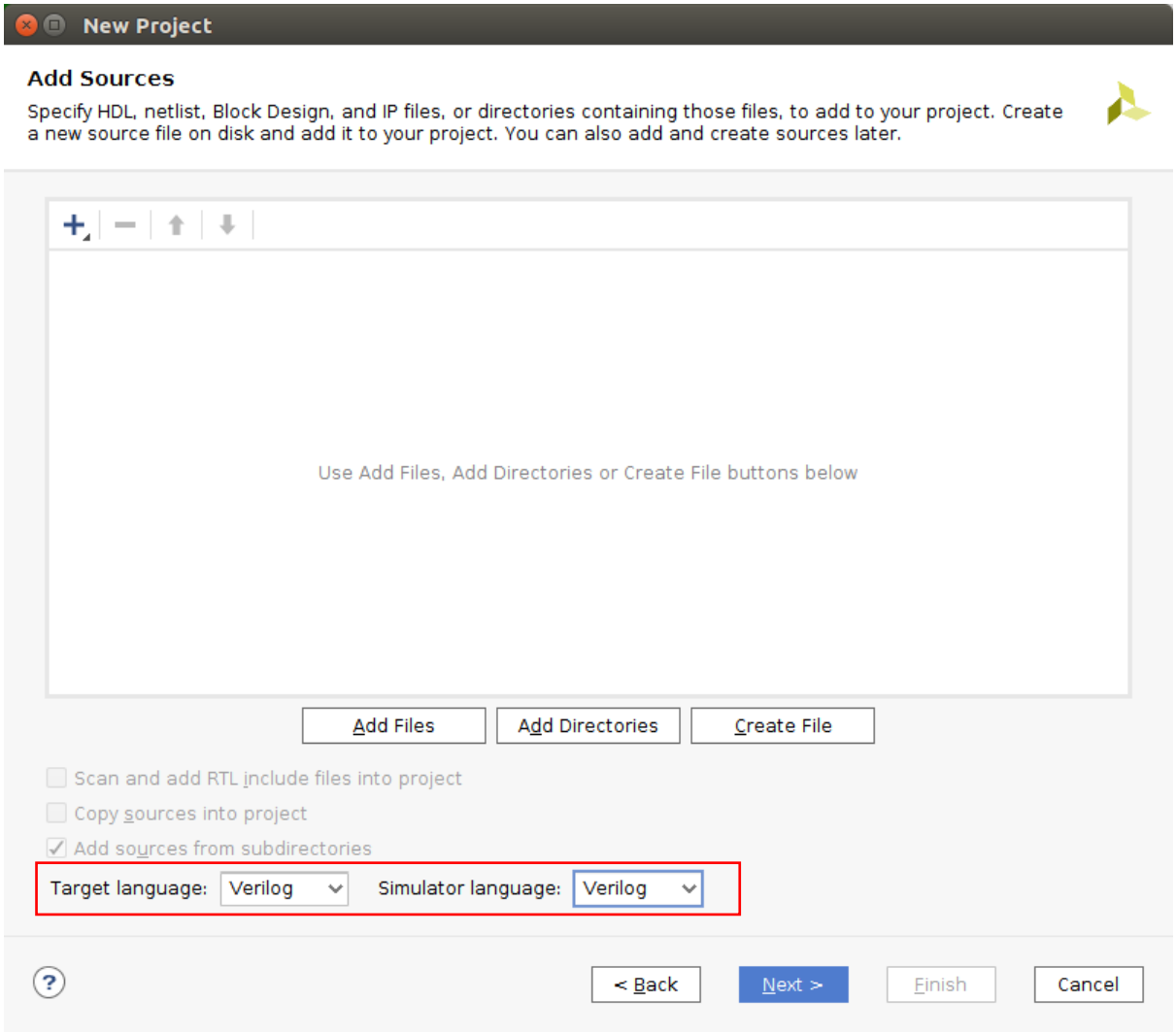


vivado를 실행한 후 프로젝트를 만든다

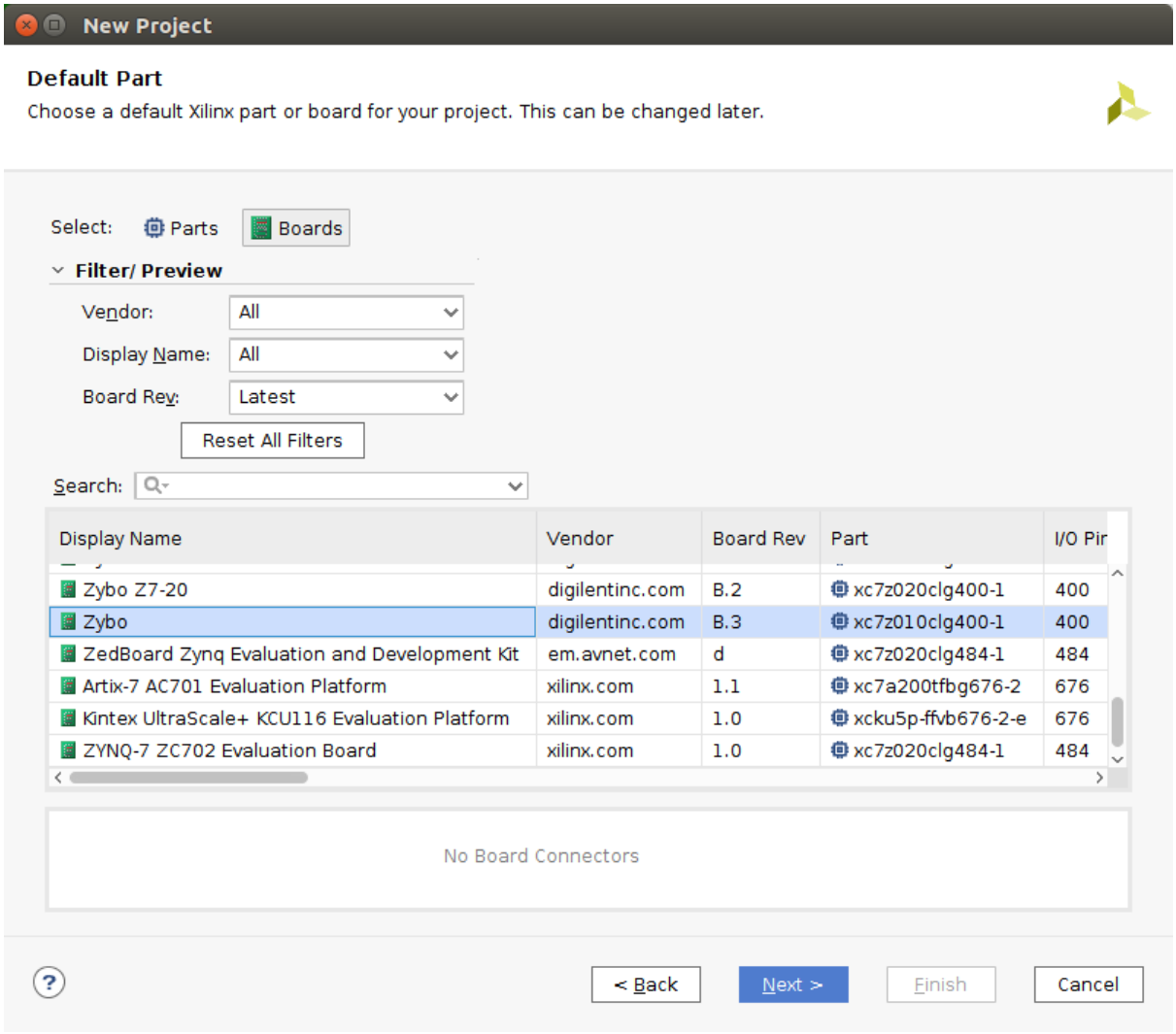
project name과 location을 설정한 후 next



RTL Project로 설정한 후 next



Target language와 Simulator language를 Verilog로 바꿔준 후 Next



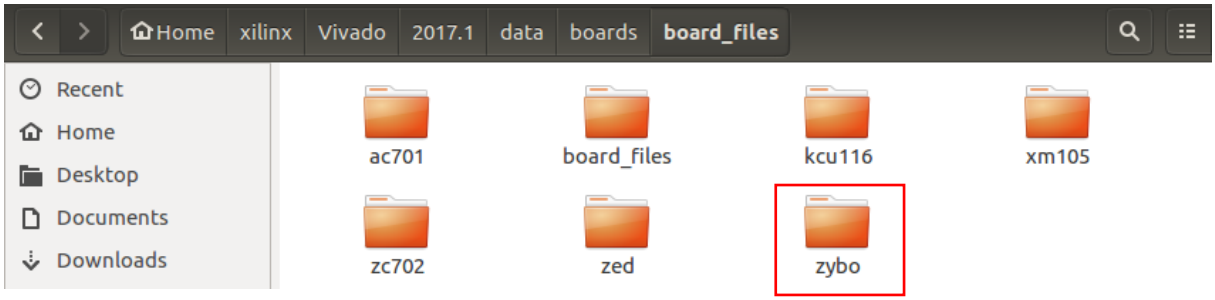
Boards에서 Zybo로 설정

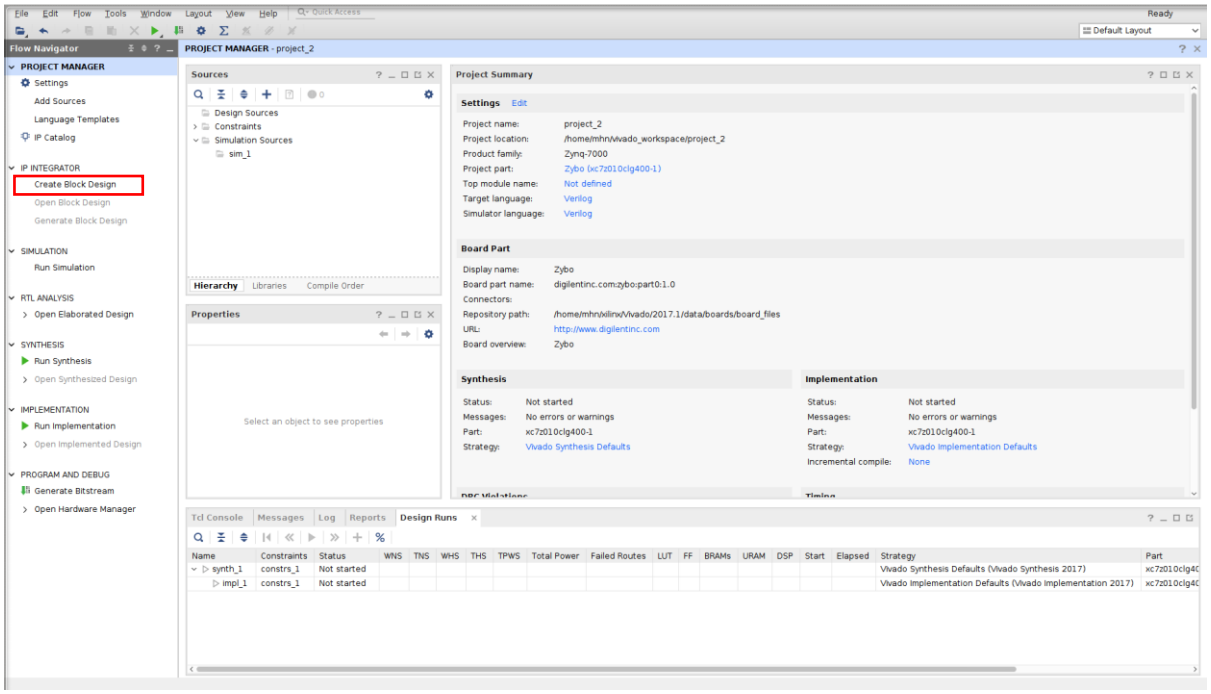
만약 없다면 https://reference.digilentinc.com/reference/software/vivado/board-files 에서 다운

압축을 푼 후 vivado-boards-master/new/board_files폴더 안에 zybo 파일을

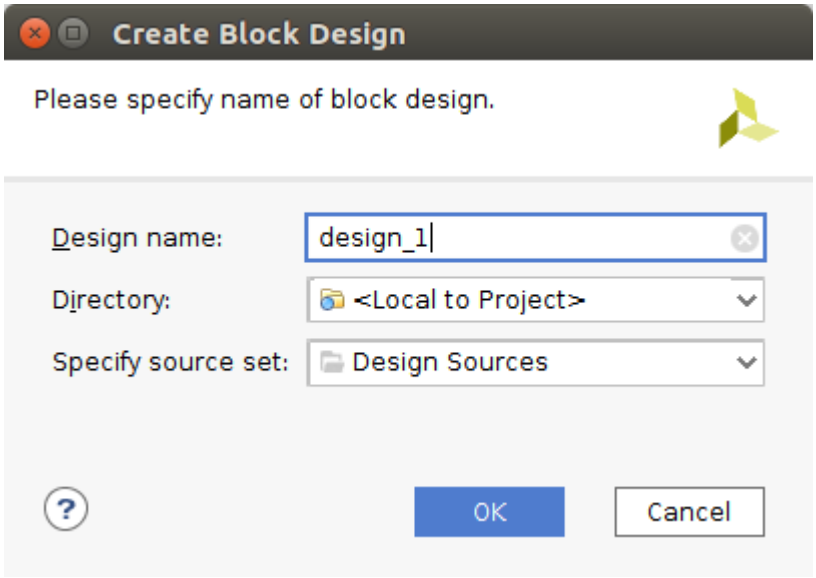
xilinx/Vivado/2017.1/data/boards/board_file 폴더로 이동

```
mhn@mhn-Z20NH-AS51B5U:~/Downloads/vivado-boards-master/new$ sudo cp -r ./board_files/zybo ../../../xilinx/
Vivado/2017.1/data/boards/board_files
```

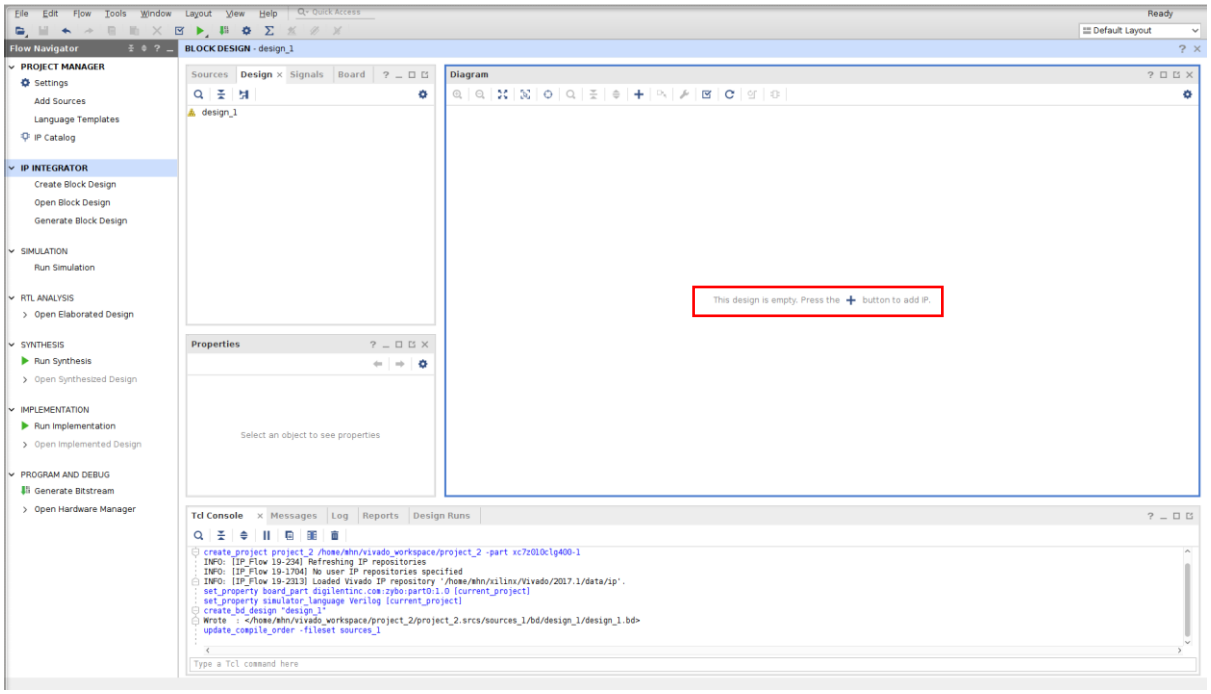




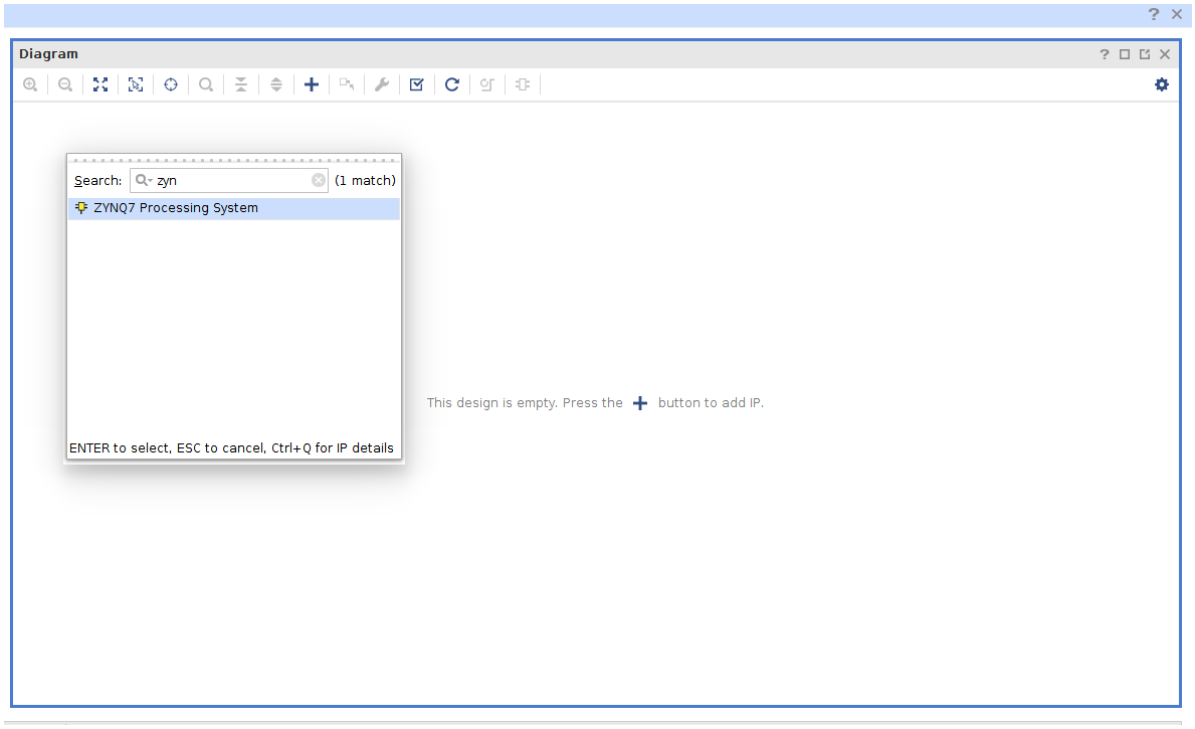
프로젝트 생성 후 왼쪽 Project Manager에서 Create Block Design



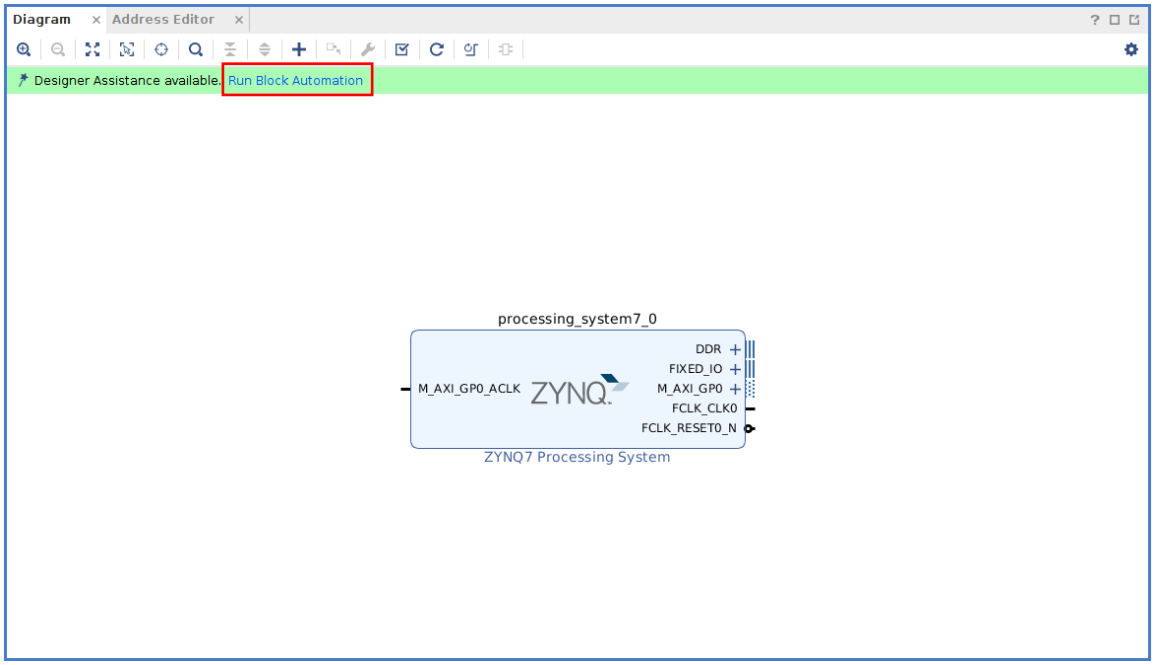
이름과 경로설정 후 ok



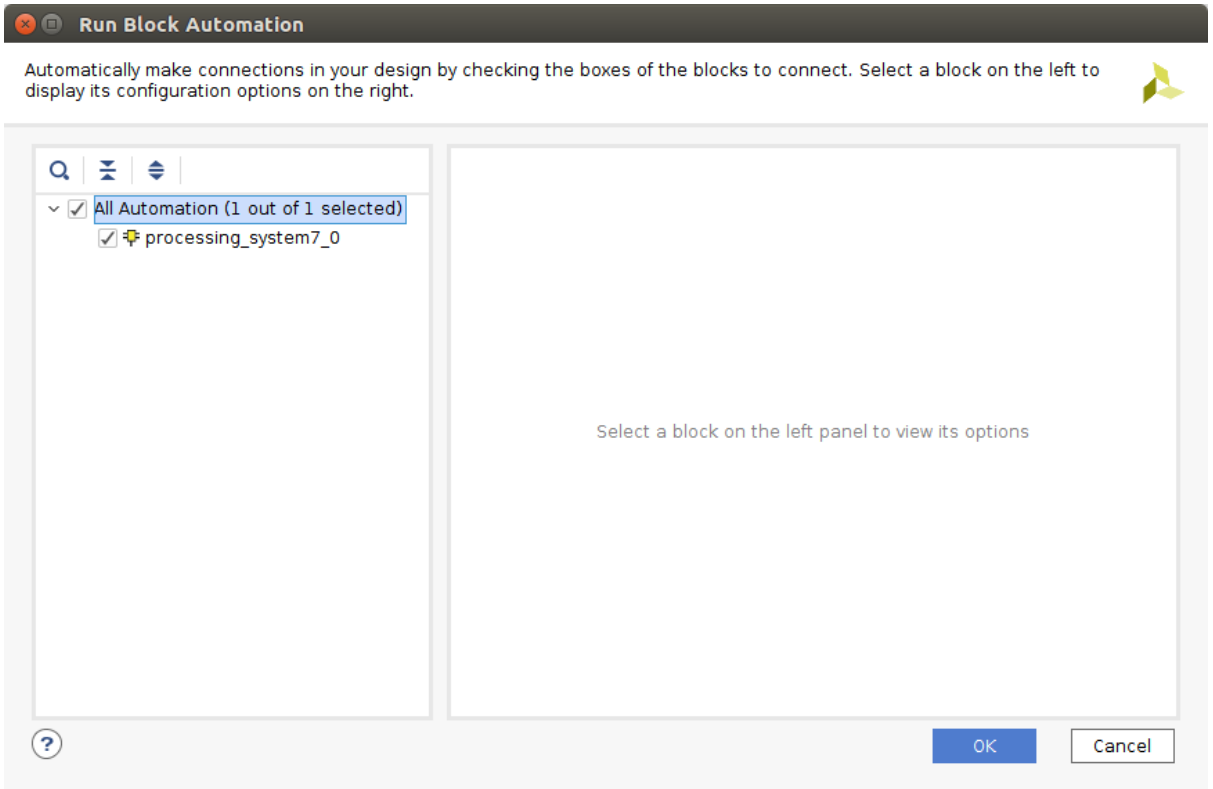
Diagram에서 +를 눌러 추가

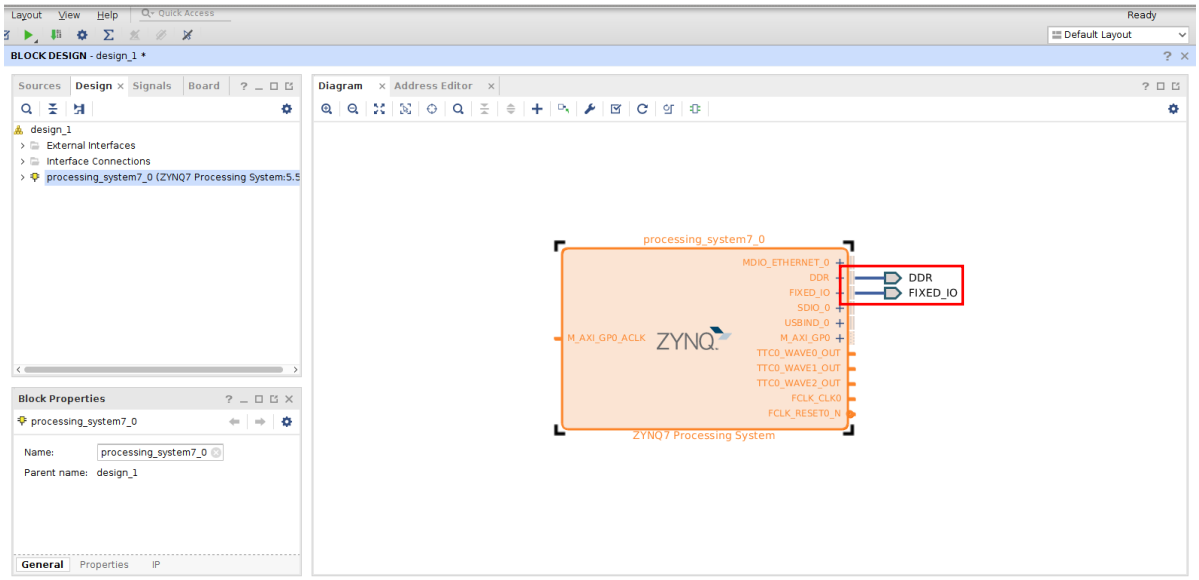


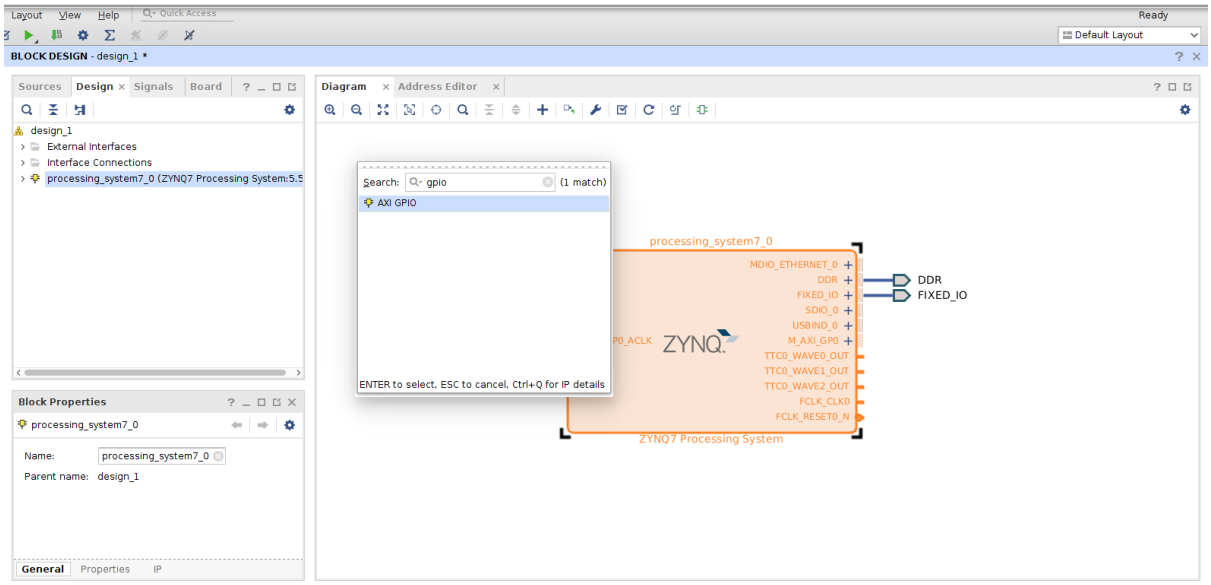
ZYNQ7 선택



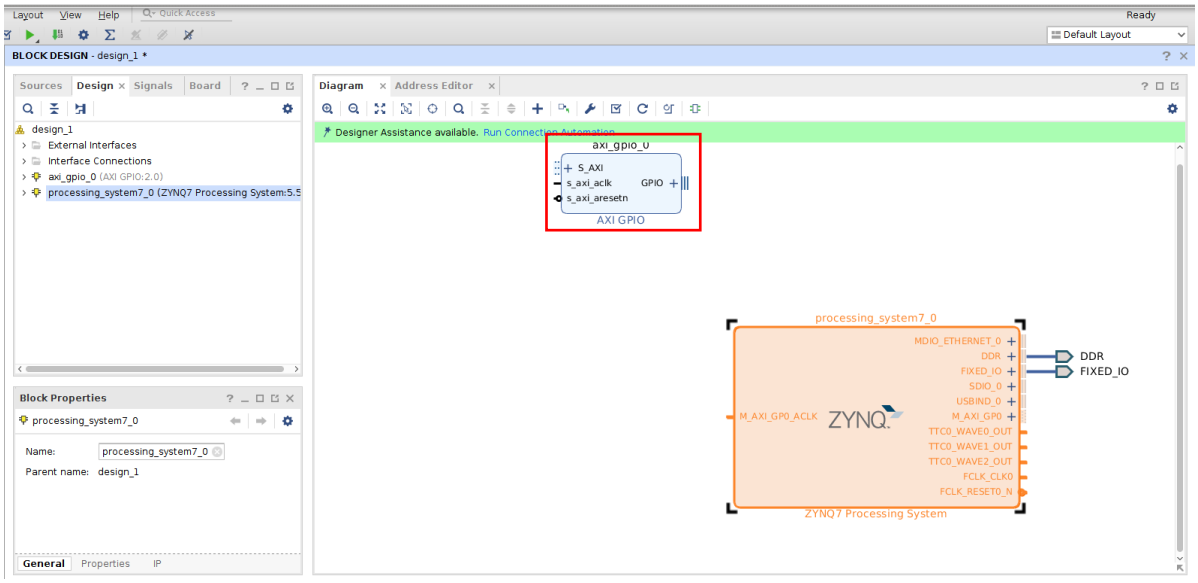
Run Block Automation

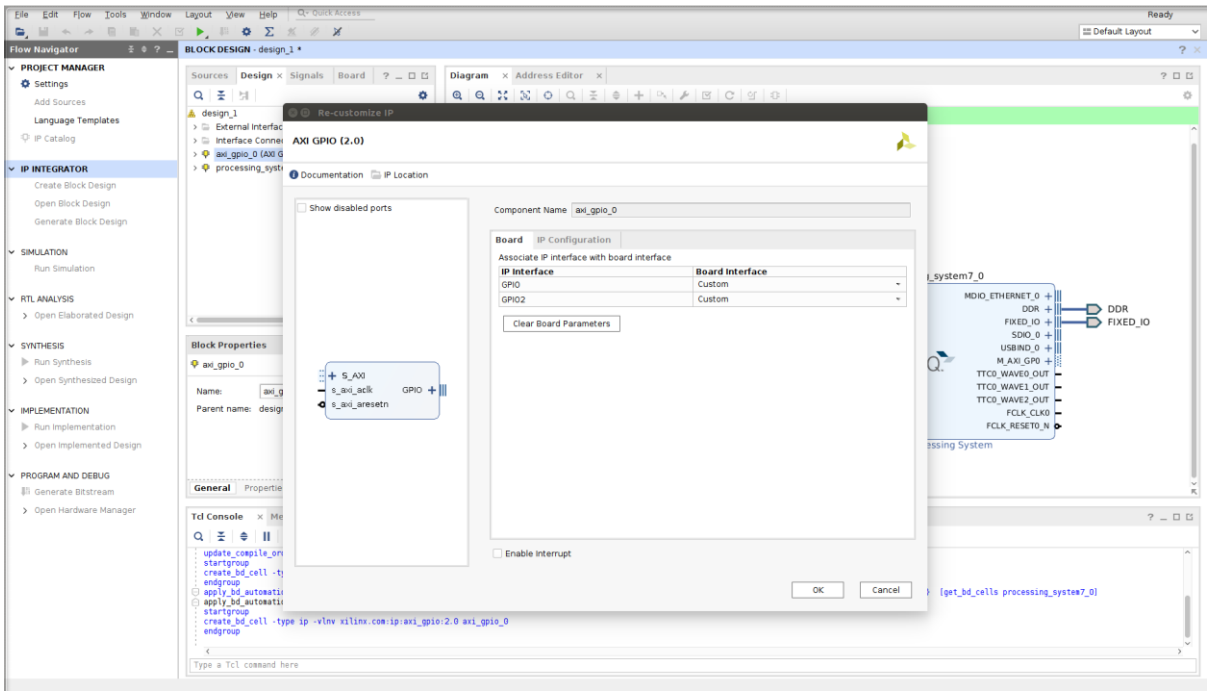




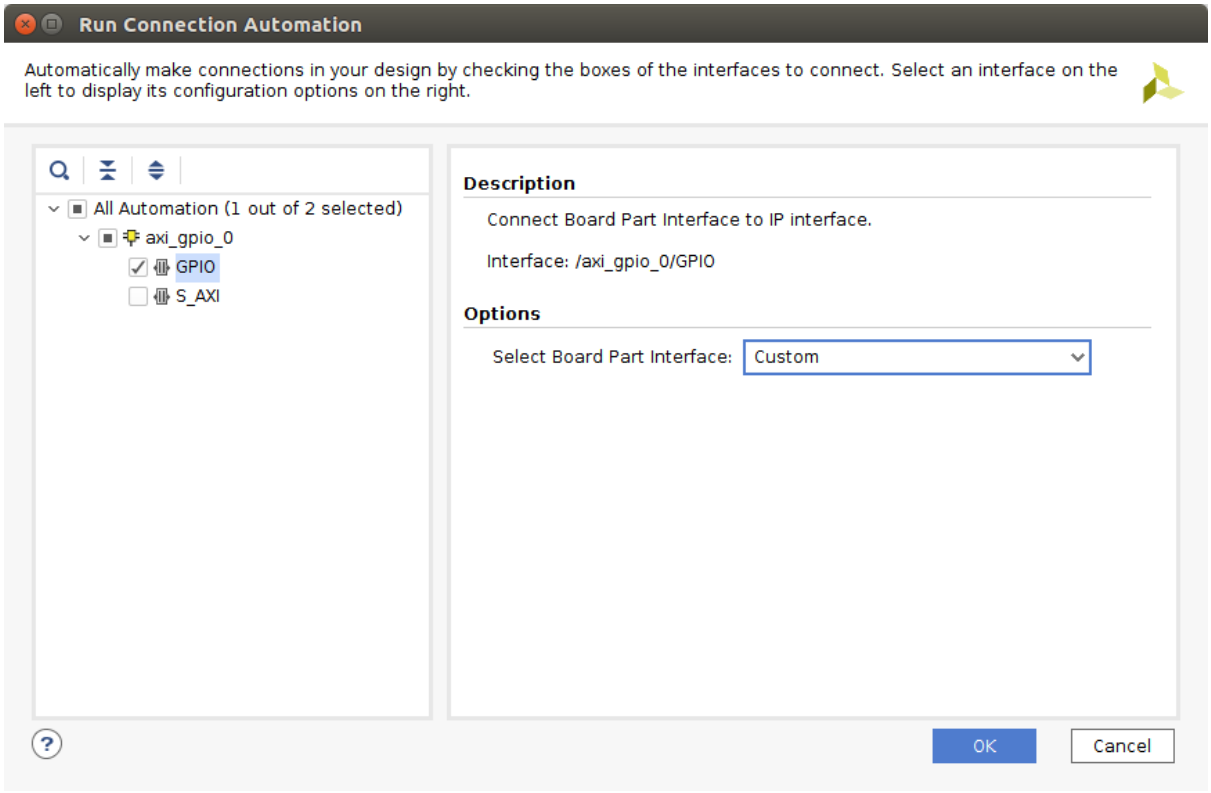


그리고 GPIO를 추가해준다

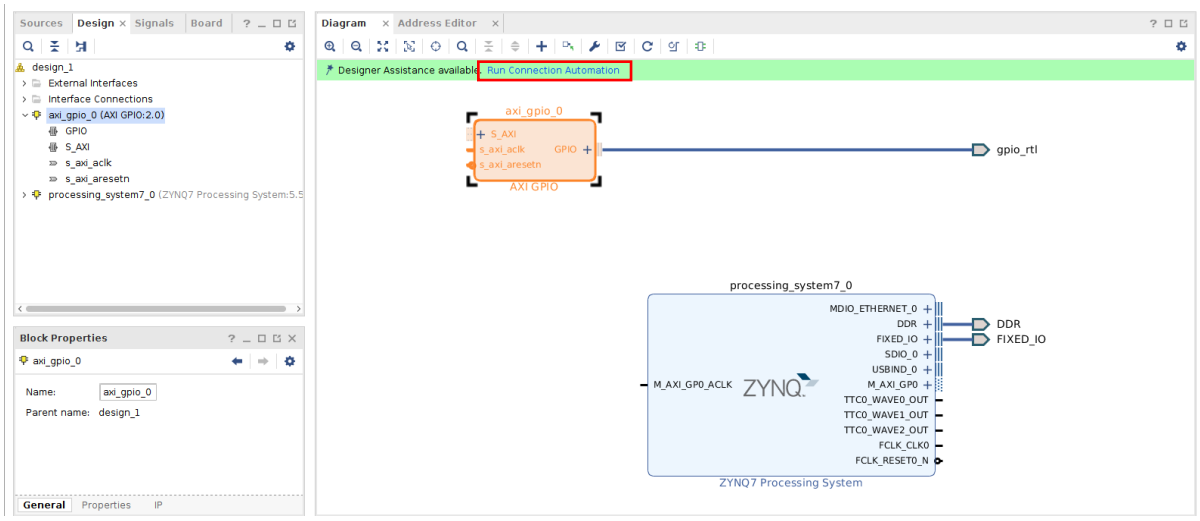




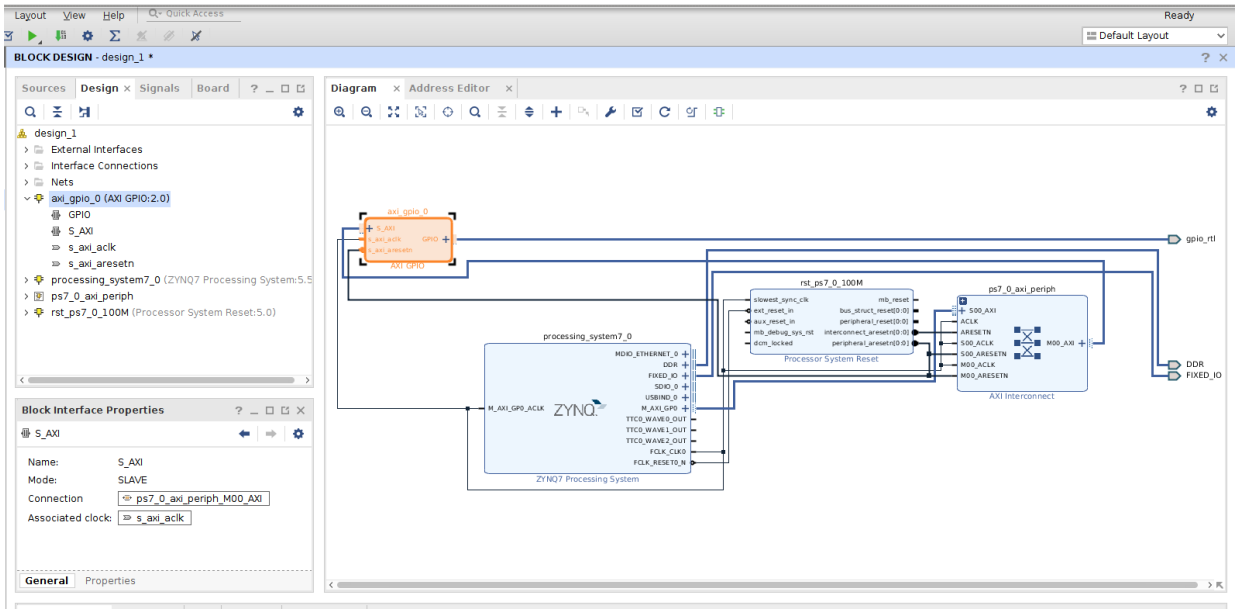
GPIO를 더블클릭하여 Board Interface가 Custom인지 확인한다



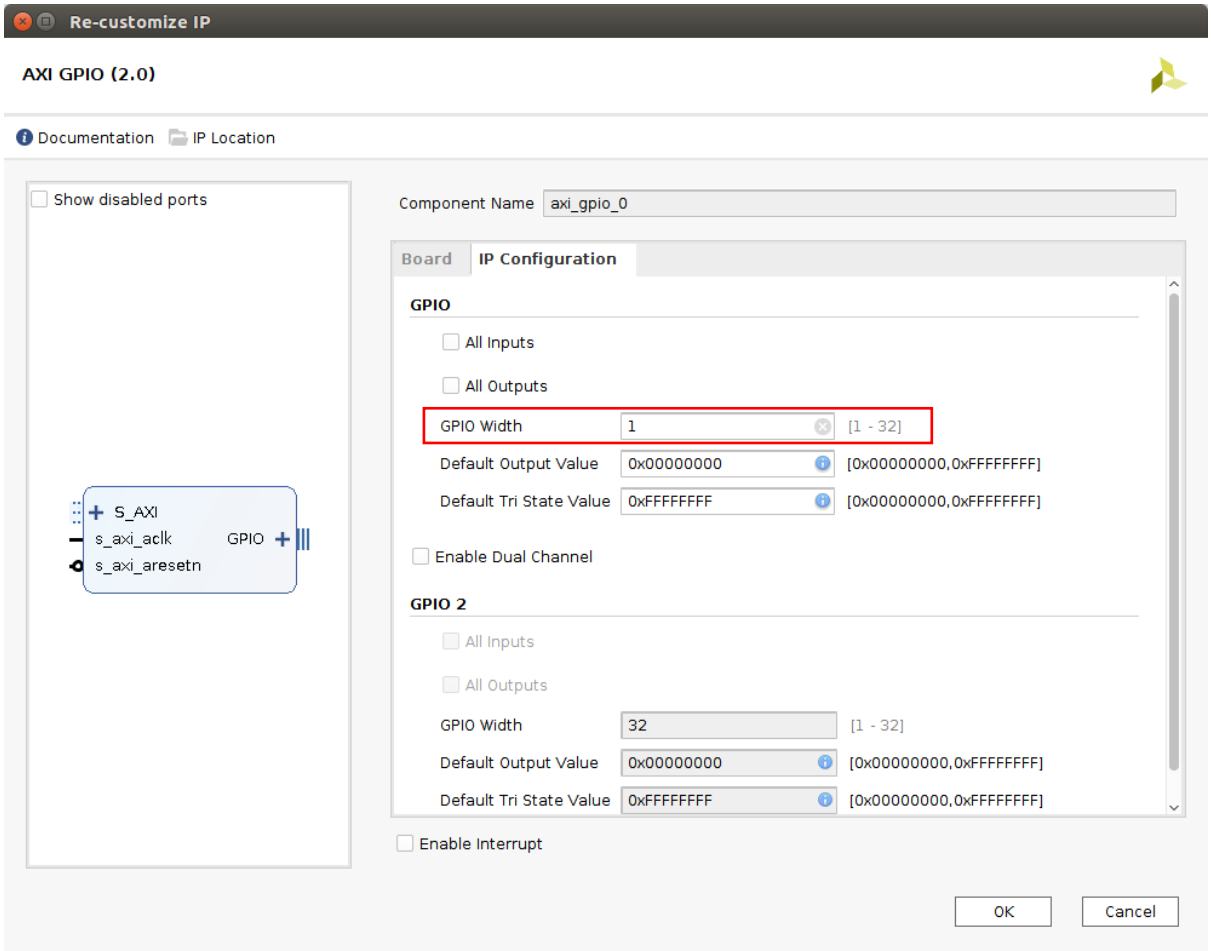
그리고 우측 상단에서 Run Connection Automation을 누른 후 GPIO를 선택하고 Custom으로 바꿔준다.



Run Connection Automation을 다시한번 눌러준다.

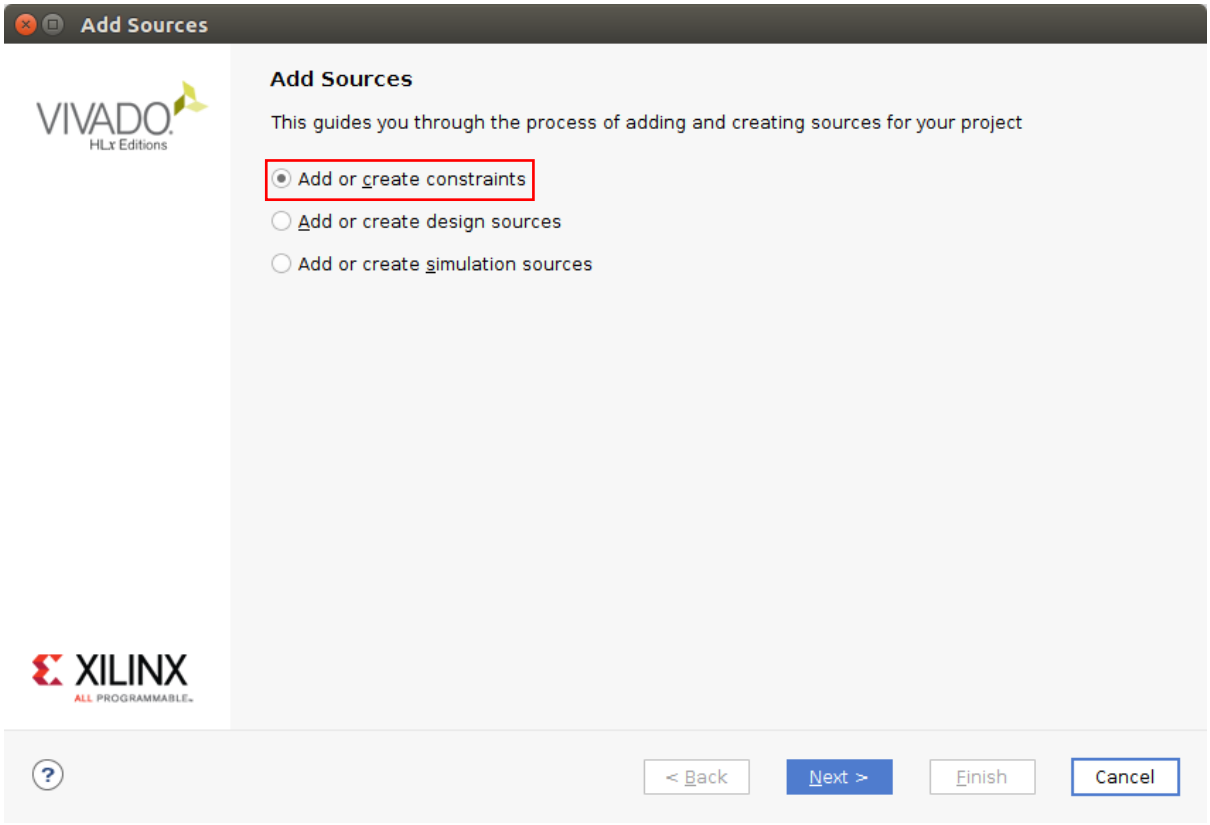


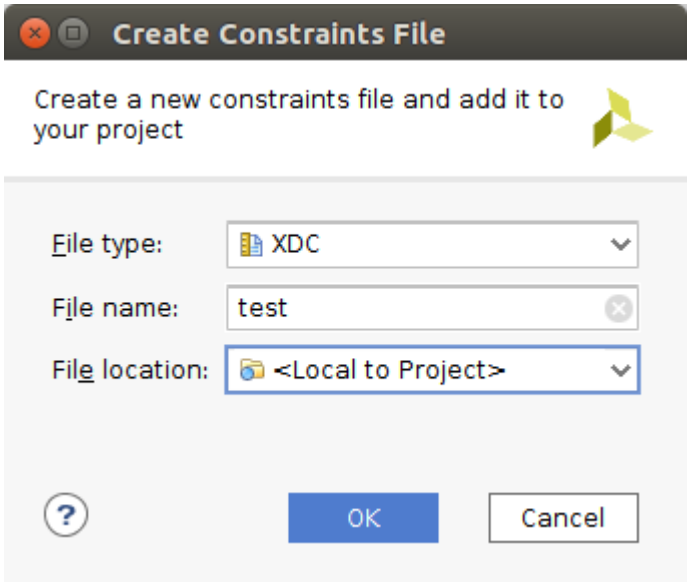
GPIO 더블클릭



IP Configuration에서 GPIO Width를 1로 바꿔준다

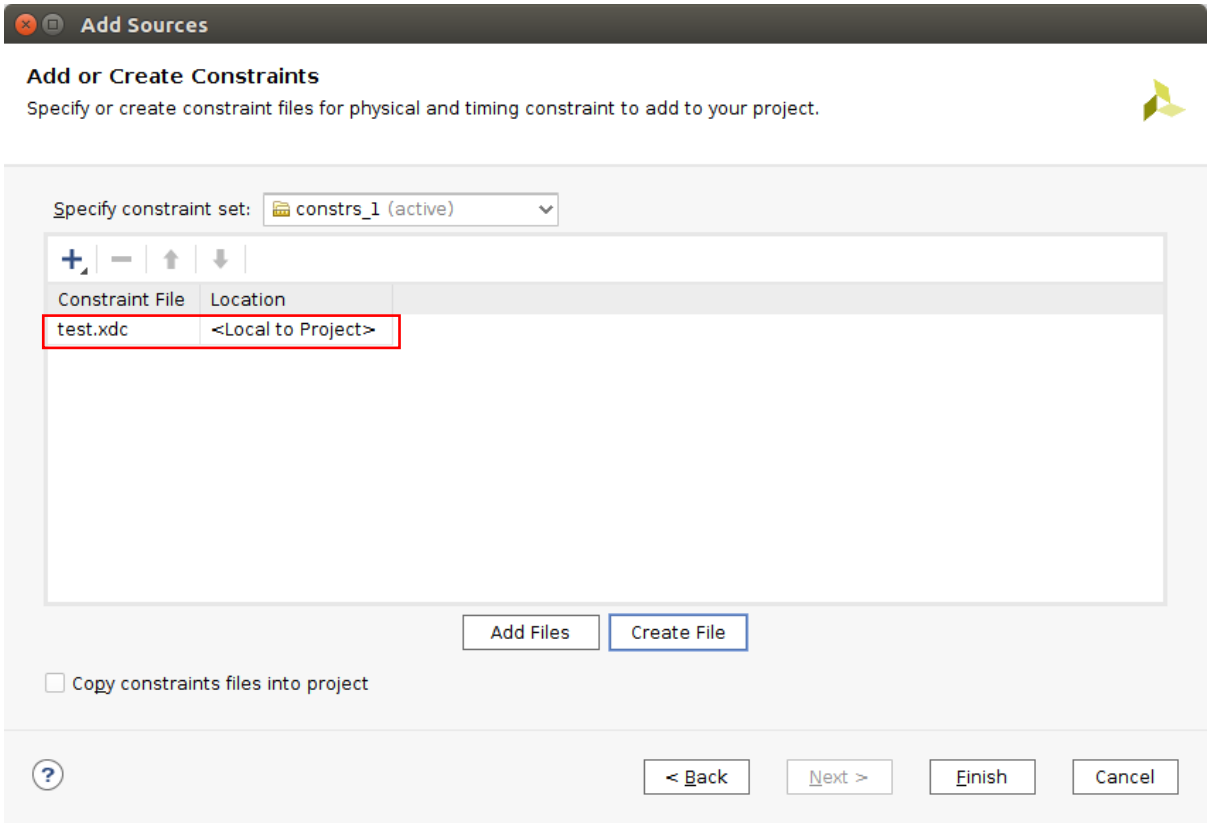
구성이 끝난 후 Project Manager에서 Add Sources를 해준다

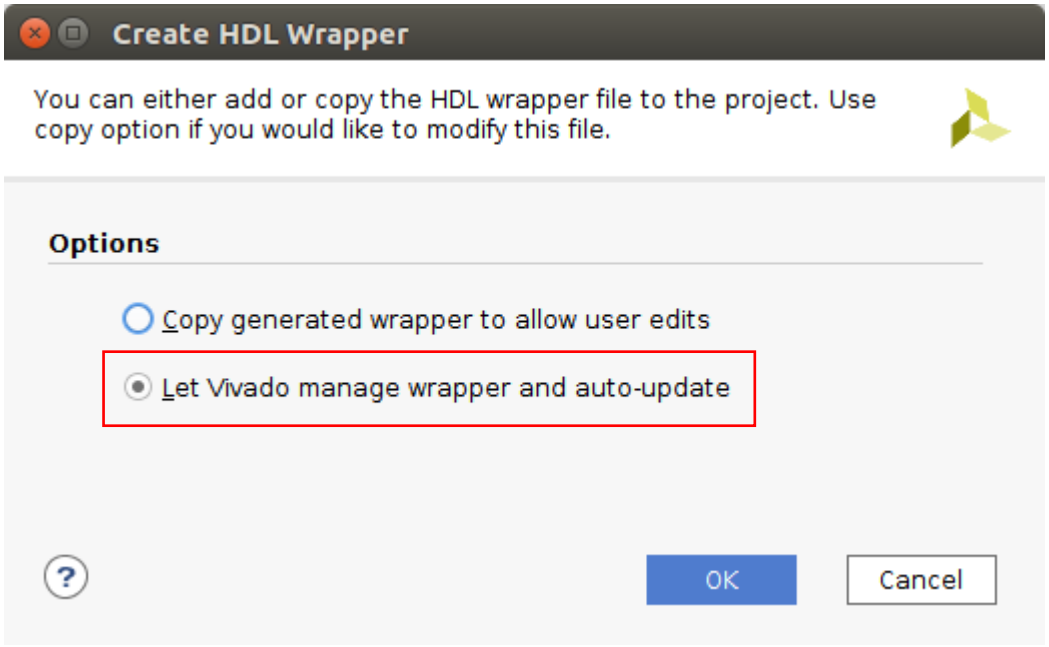


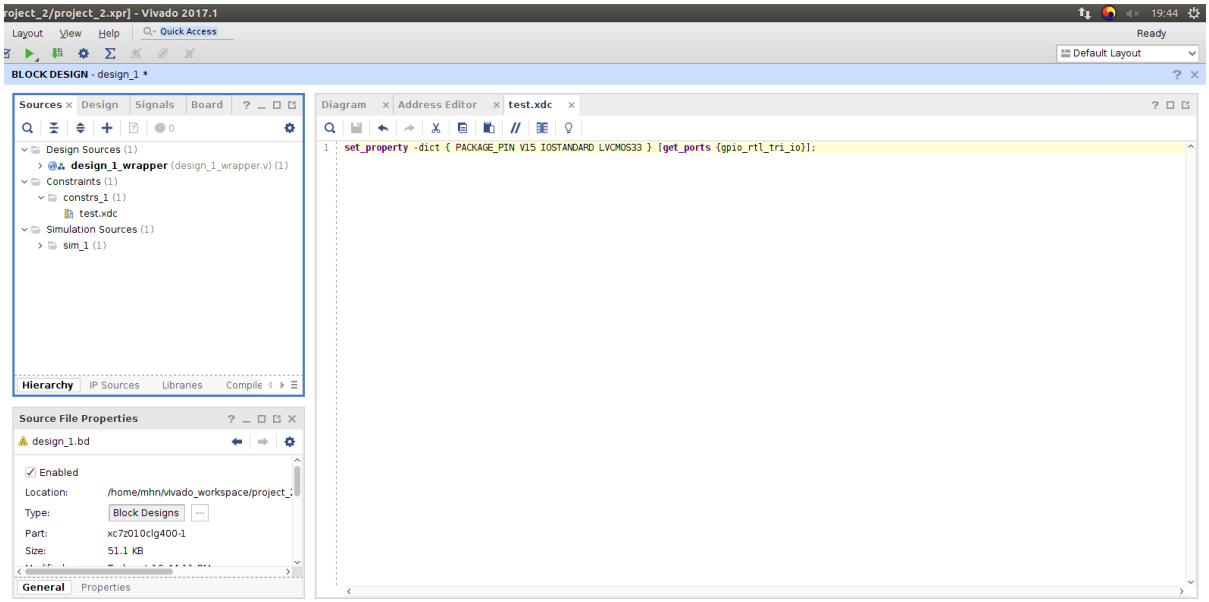


Create File을 누른다

Type은 XDC로 하고 이름과 경로 설정 후 OK


Add Sources
Add or Create Constraints
Specify or create constraint files for physical and timing constraint to add to your project.
Specify constraint set:
constrs_1 (active)
Constraint File
Location
test.xdc
<Local to Project>
Add Files
Create File
Copy constraints files into project
< Back
Next >
Finish
Cancel


Create HDL Wrapper
You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.
Options
Copy generated wrapper to allow user edits
Let Vivado manage wrapper and auto-update
OK
Cancel



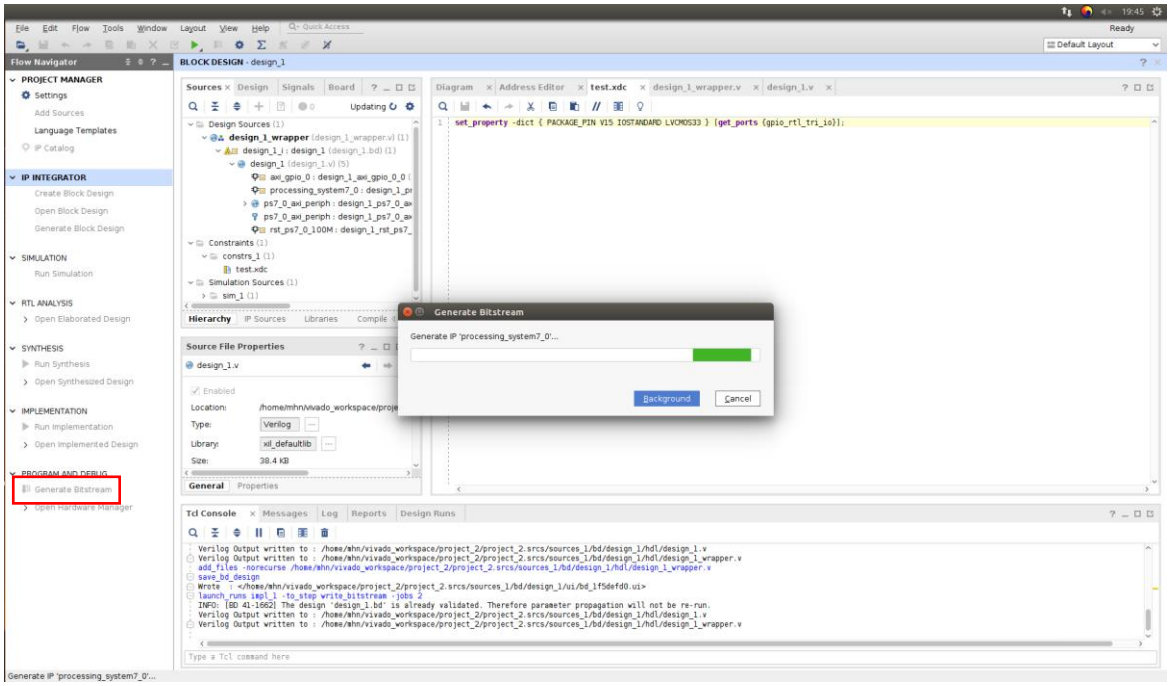
사용할 핀을 설정한다. (데이터시트 참조)

ZYBO™ FPGA Board Reference Manual DIGILENT

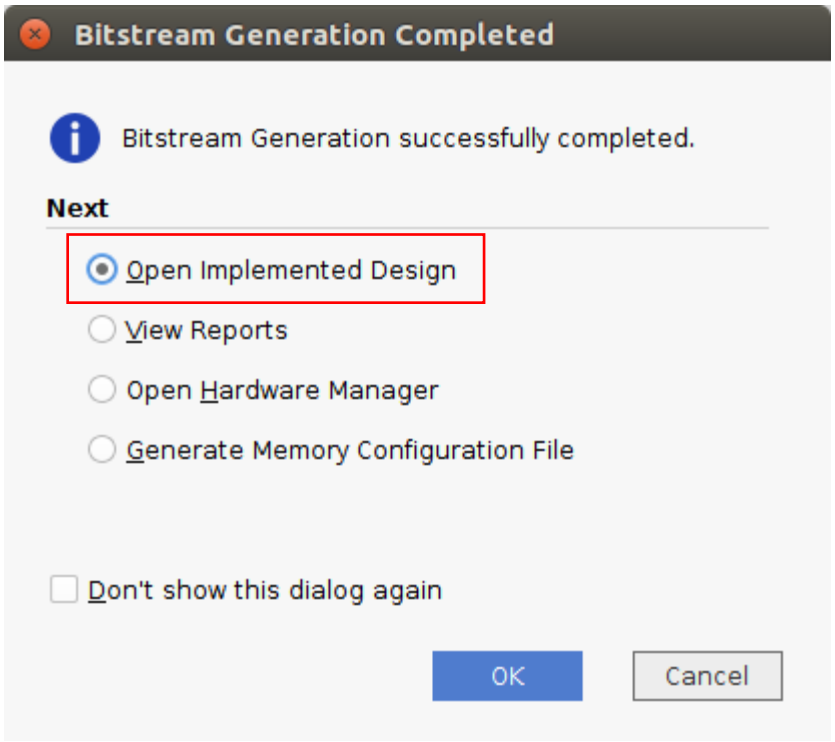
Pmod JA (XADC)	Pmod JB (Hi-Speed)	Pmod JC (Hi-Speed)	Pmod JD (Hi-Speed)	Pmod JE (Std.)	Pmod JF (MIO)
JA1: N15	JB1: T20	JC1: V15	JD1: T14	JE1: V12	JF1: MIO-13
JA2: L14	JB2: U20	JC2: W15	JD2: T15	JE2: W16	JF2: MIO-10
JA3: K16	JB3: V20	JC3: T11	JD3: P14	JE3: J15	JF3: MIO-11
JA4: K14	JB4: W20	JC4: T10	JD4: R14	JE4: H15	JF4: MIO-12
JA7: N16	JB7: Y18	JC7: W14	JD7: U14	JE7: V13	JF7: MIO-0
JA8: L15	JB8: Y19	JC8: Y14	JD8: U15	JE8: U17	JF8: MIO-9
JA9: J16	JB9: W18	JC9: T12	JD9: V17	JE9: T17	JF9: MIO-14
JA10: J14	JB10: W19	JC10: U12	JD10: V18	JE10: Y17	JF10: MIO-15

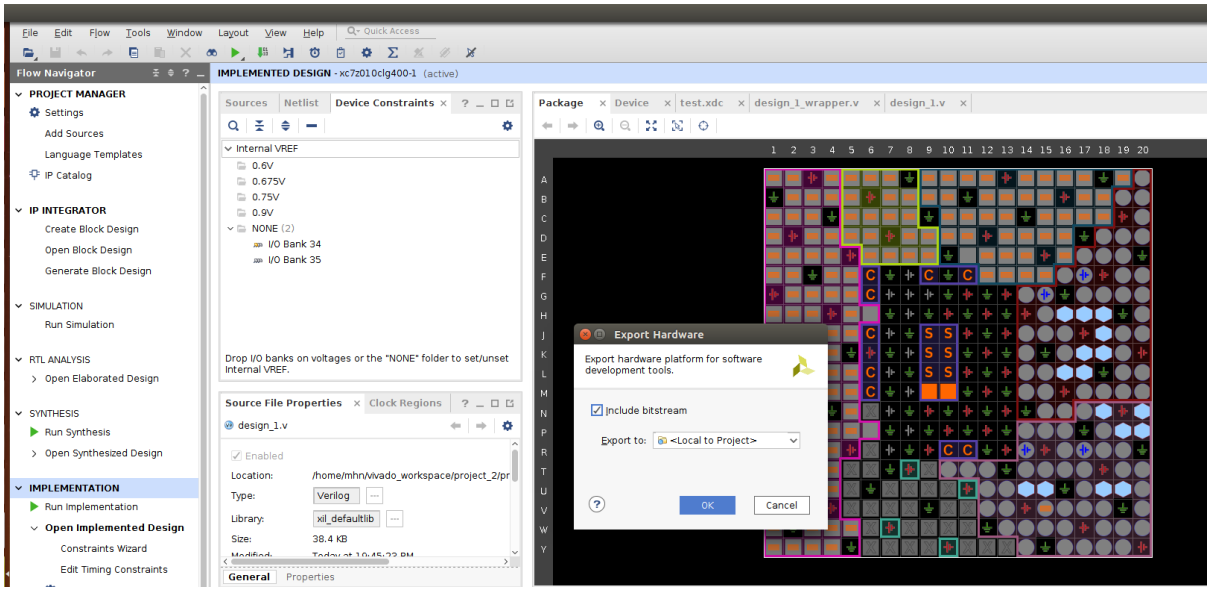
Table 9. Pmod pinout.

그리고 래핑한다

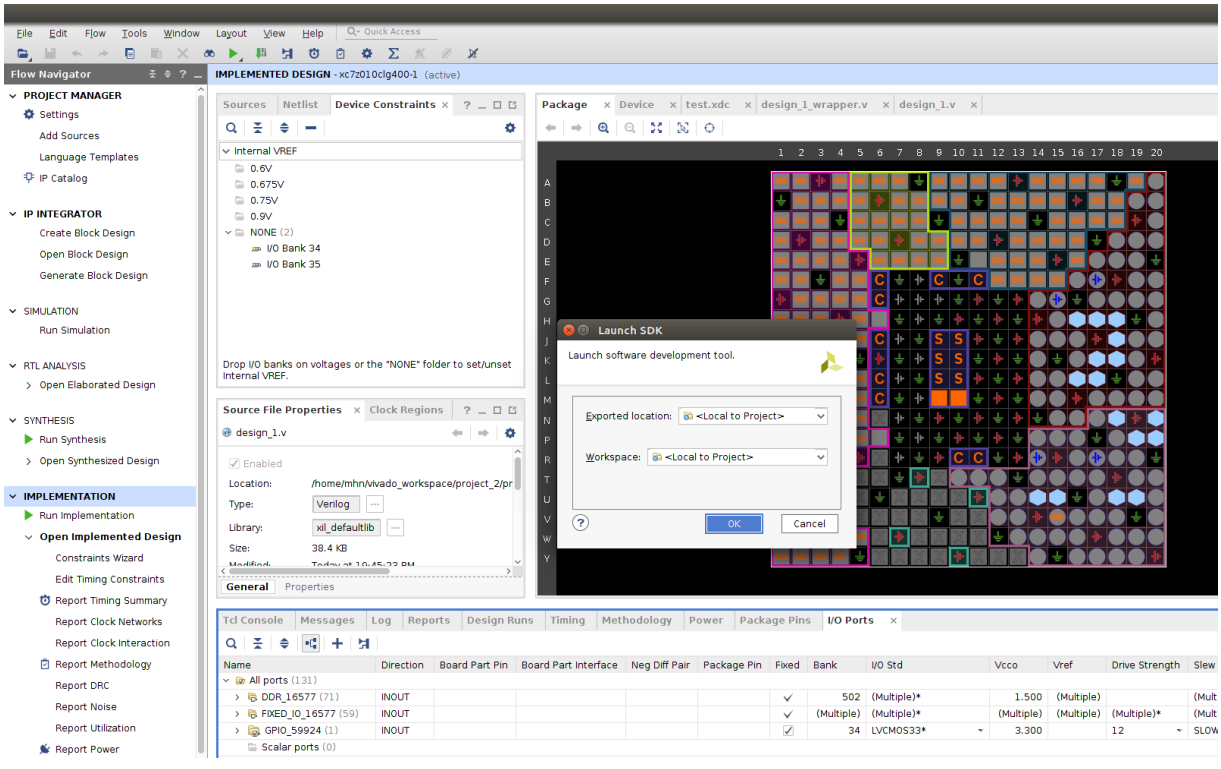


Generate Bitstream

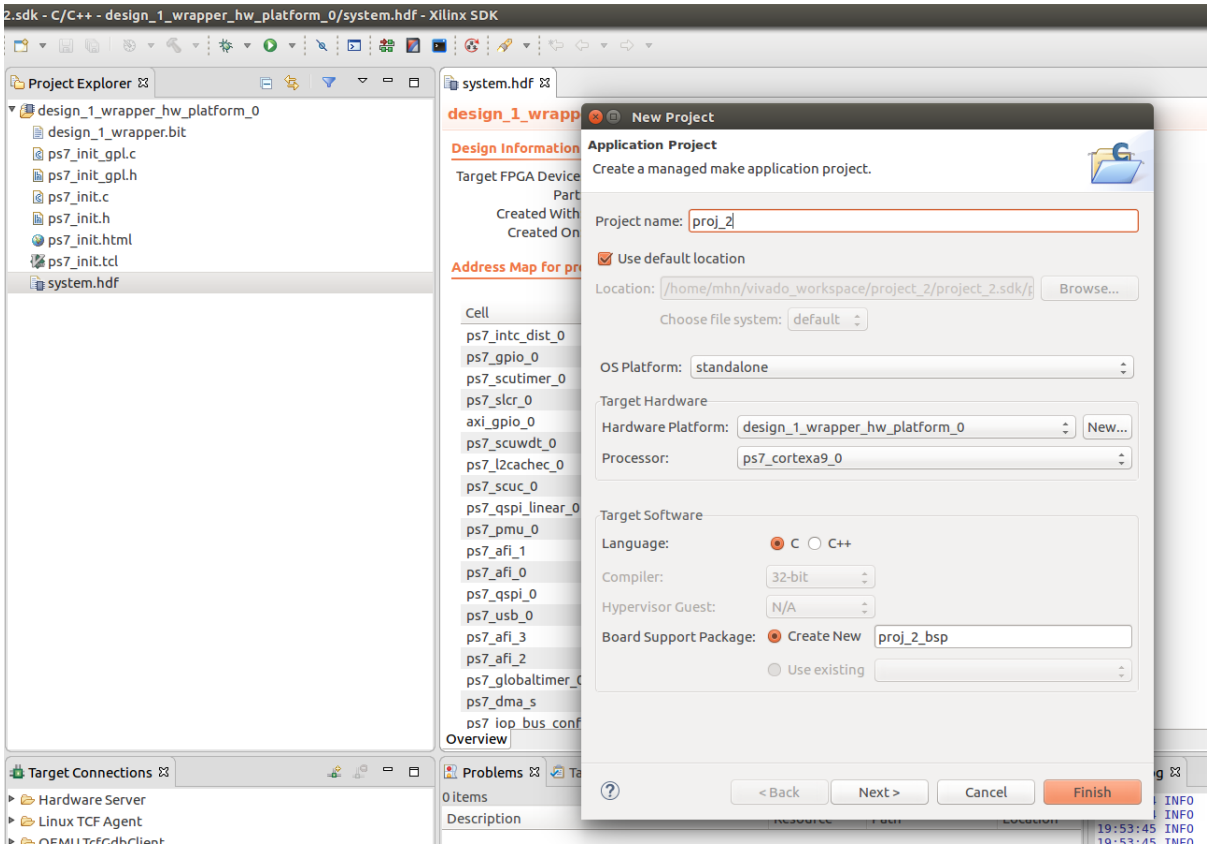




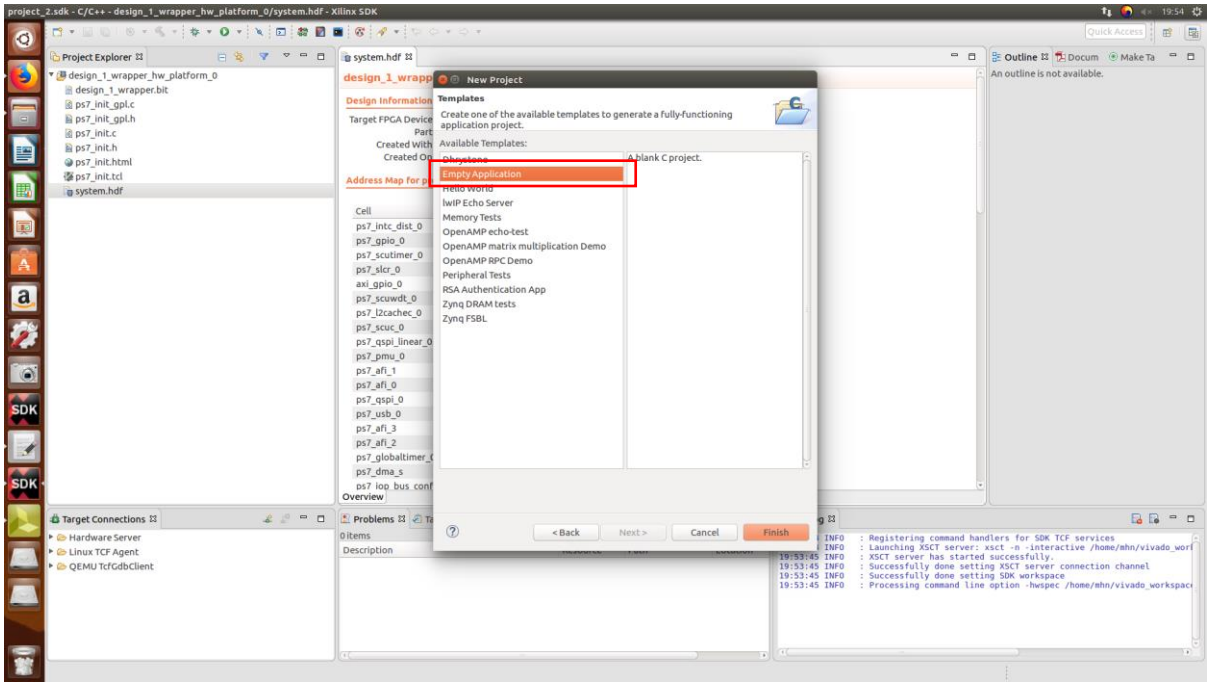
Export Hardware

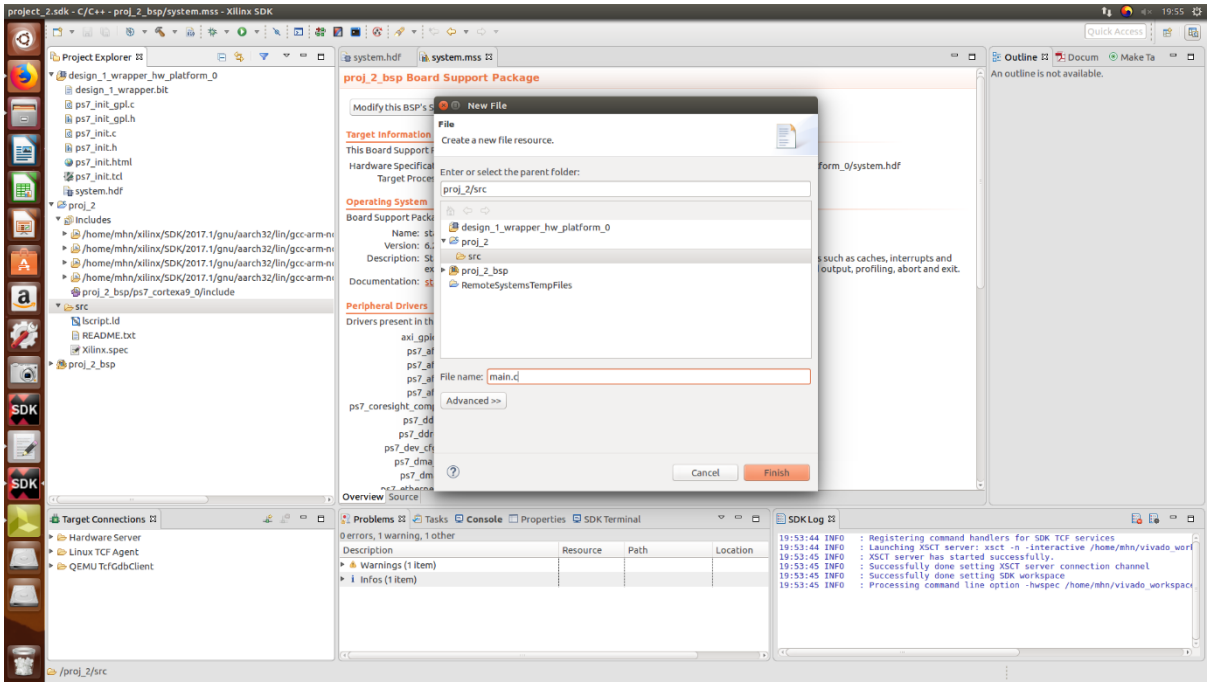


Launch SDK


2.sdk - C/C++ - design_1_wrapper_hw_platform_0/system.hdf - Xilinx SDK
Project Explorer
design_1_wrapper_hw_platform_0
design_1_wrapper.bit
ps7_init_gpl.c
ps7_init_gpl.h
ps7_init.c
ps7_init.h
ps7_init.html
ps7_init.tcl
system.hdf
system.hdf
New Project
Application Project
Create a managed make application project.
Project name: proj_2
Use default location
Location: /home/mhn/vivado_workspace/project_2/project_2.sdk/
Browse...
Choose file system: default
OS Platform: standalone
Target Hardware
Hardware Platform: design_1_wrapper_hw_platform_0
New...
Processor: ps7_cortexa9_0
Target Software
Language: C C++
Compiler: 32-bit
Hypervisor Guest: N/A
Board Support Package: Create New proj_2_bsp
Use existing
< Back
Next >
Cancel
Finish
Target Connections
Hardware Server
Linux TCF Agent
Problems
0 items
Description

프로젝트를 만든다


project_2.sdk - C/C++ - design_1_wrapper_hw_platform_0/system.hdf - Xilinx SDK
New Project
Templates
Create one of the available templates to generate a fully-functioning application project.
Available Templates:
Empty Application
Hello World
lwIP Echo Server
Memory Tests
OpenAMP echo-test
OpenAMP matrix multiplication Demo
OpenAMP RPC Demo
Peripheral Tests
RSA Authentication App
Zynq DRAM tests
Zynq FSBL
A blank C project.
< Back
Next >
Cancel
Finish



프로젝트 생성 후 src폴더에서 main.c 파일을 생성한 후 코딩