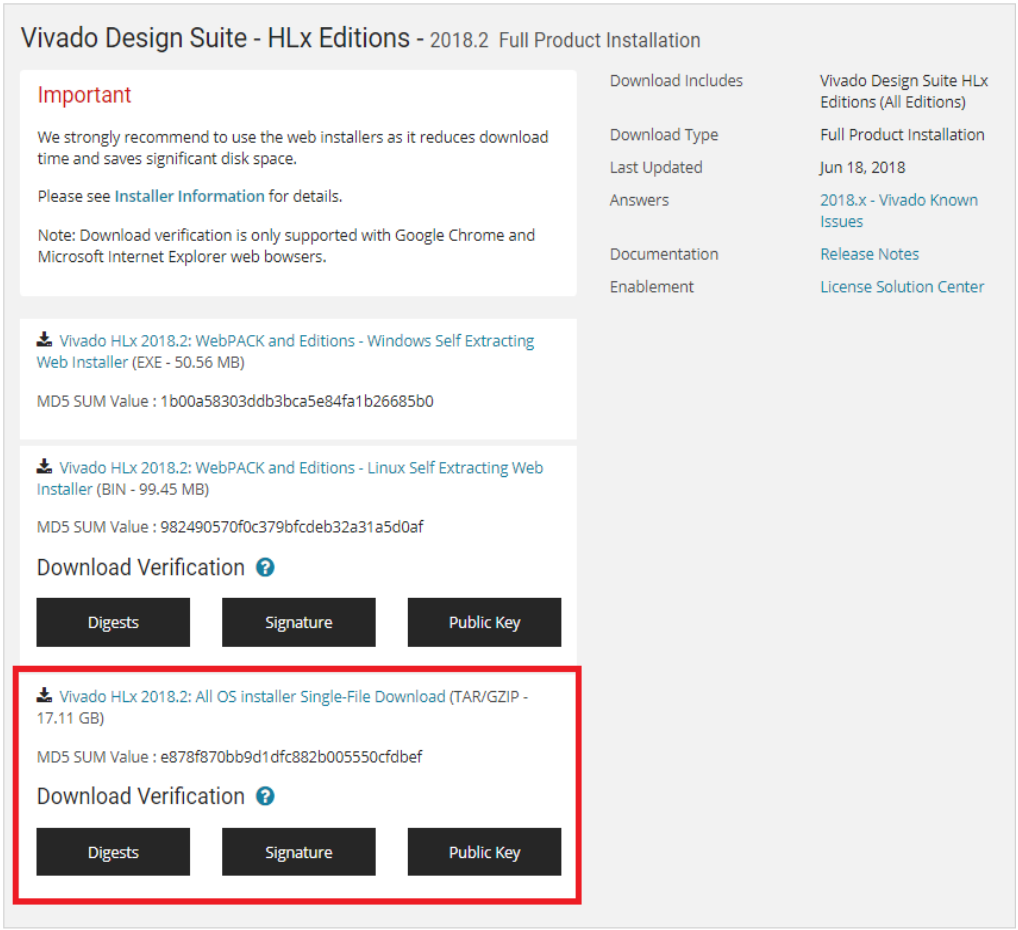
EasySoC-Z7010 FPGA/SoC 설계 플랫폼

2. Xilinx Vivado를 이용한 EasySoC-Z7010 실습

Vivado 설치 및 실행하기

- Vivado 다운받기
  - Xilinx 홈페이지(www.Xilinx.com)에서 회원가입 후 Vivado Design Suite – HLx Editions 설치 파일을 다운로드 받는다.

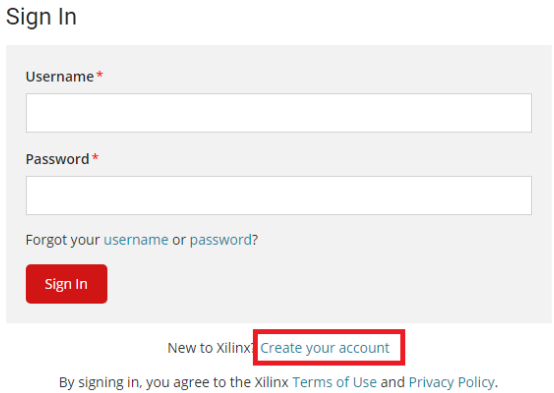


HUINS

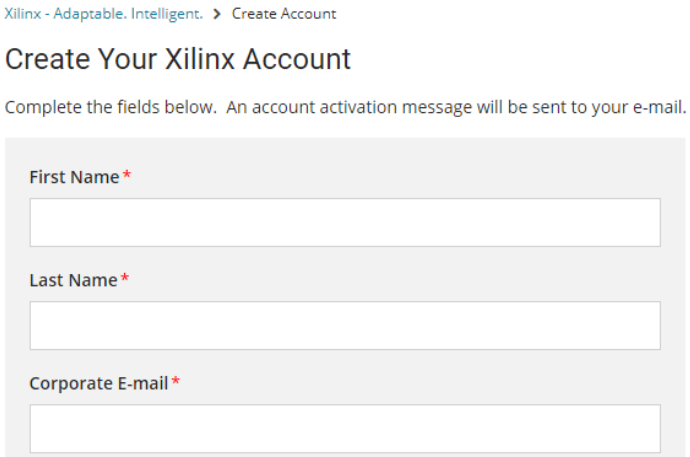
Vivado 설치 및 실행하기

- Vivado 다운받기

✓ 계정이 없을 경우, Xilinx에 가입을 위해 Create Account 항목을 클릭한다.



✓ 아이디, 패스워드, 이메일 주소 등을 입력해 계정을 생성한다.

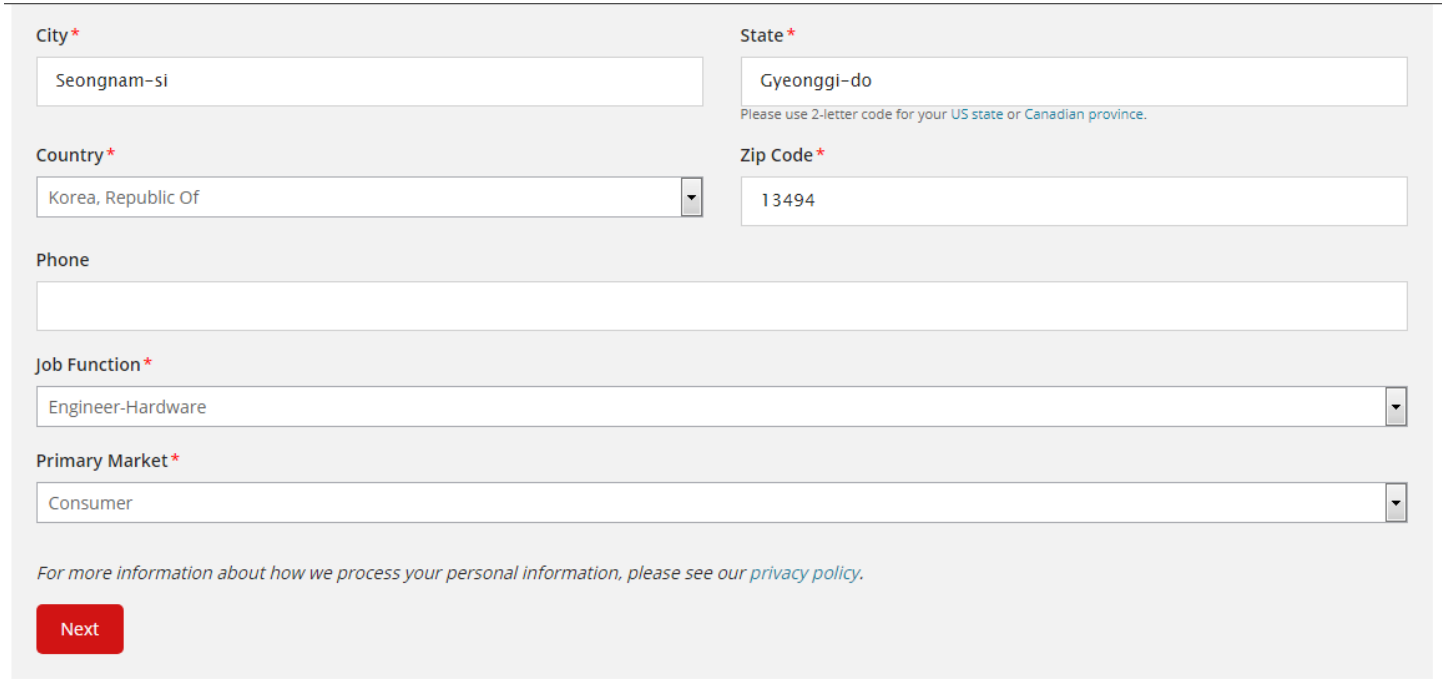


HUINS

Vivado 설치 및 실행하기

- Vivado 다운받기

  ✓ 이전의 다운로드 페이지를 다시 들어간 후 해당 버전의 파일을 다운로드 후 아래의 프로필 확인 페이지에서 아래 next 버튼을 클릭해 다운 받는다.

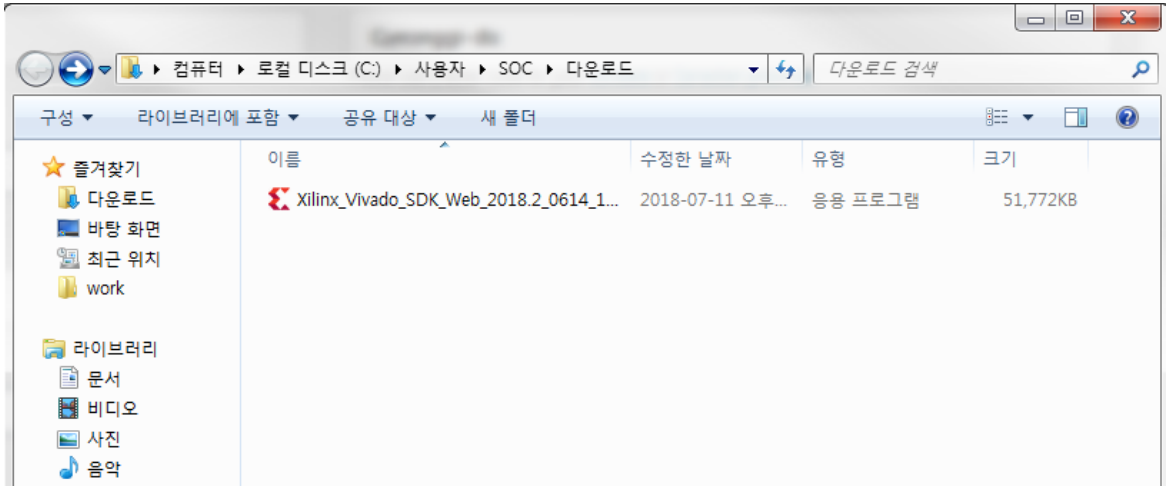


HUINS

Vivado 설치 및 실행하기

- Vivado 설치하기

  ✓ 다운로드 경로에서 다운 받은 파일을 찾은 뒤 실행하여 설치를 진행한다.

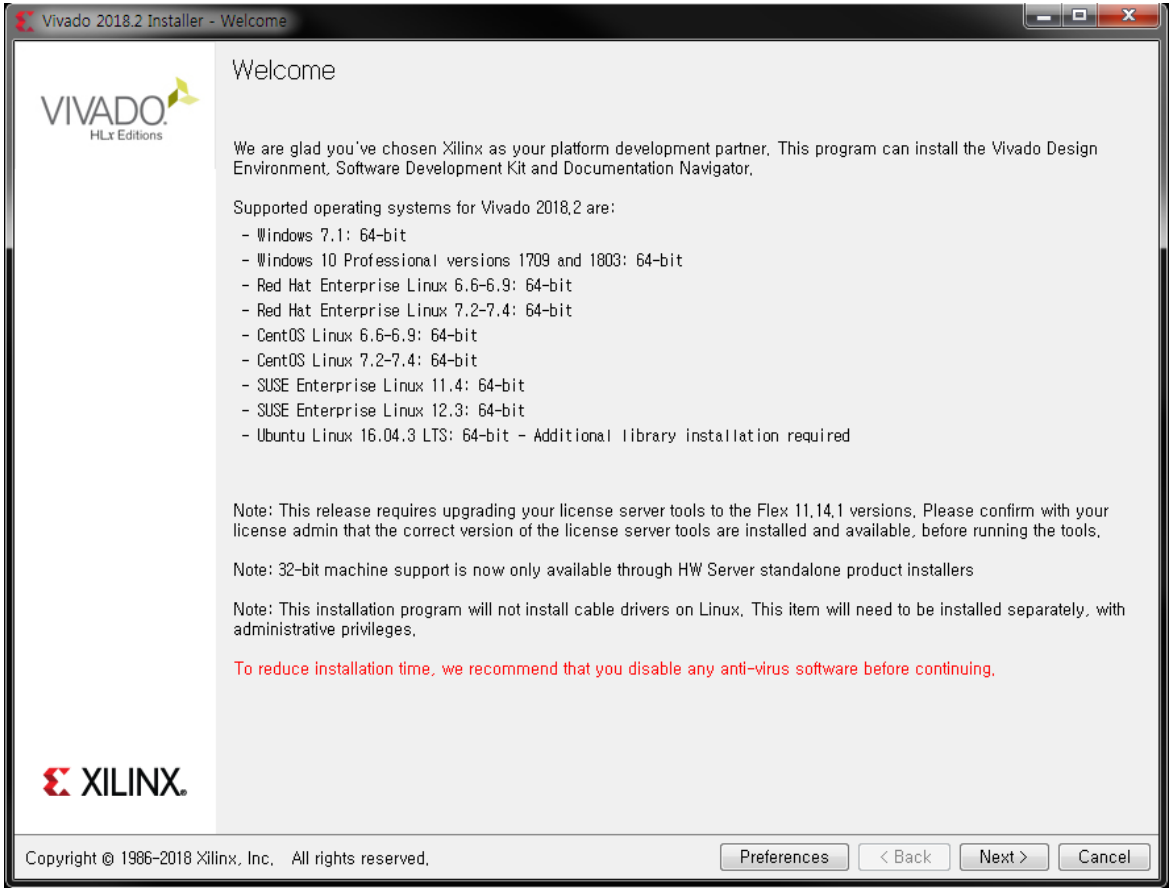


HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기

  ✓ 설치를 시작하는 대화상자에서 Next 버튼을 클릭하여 다음 단계를 진행한다.

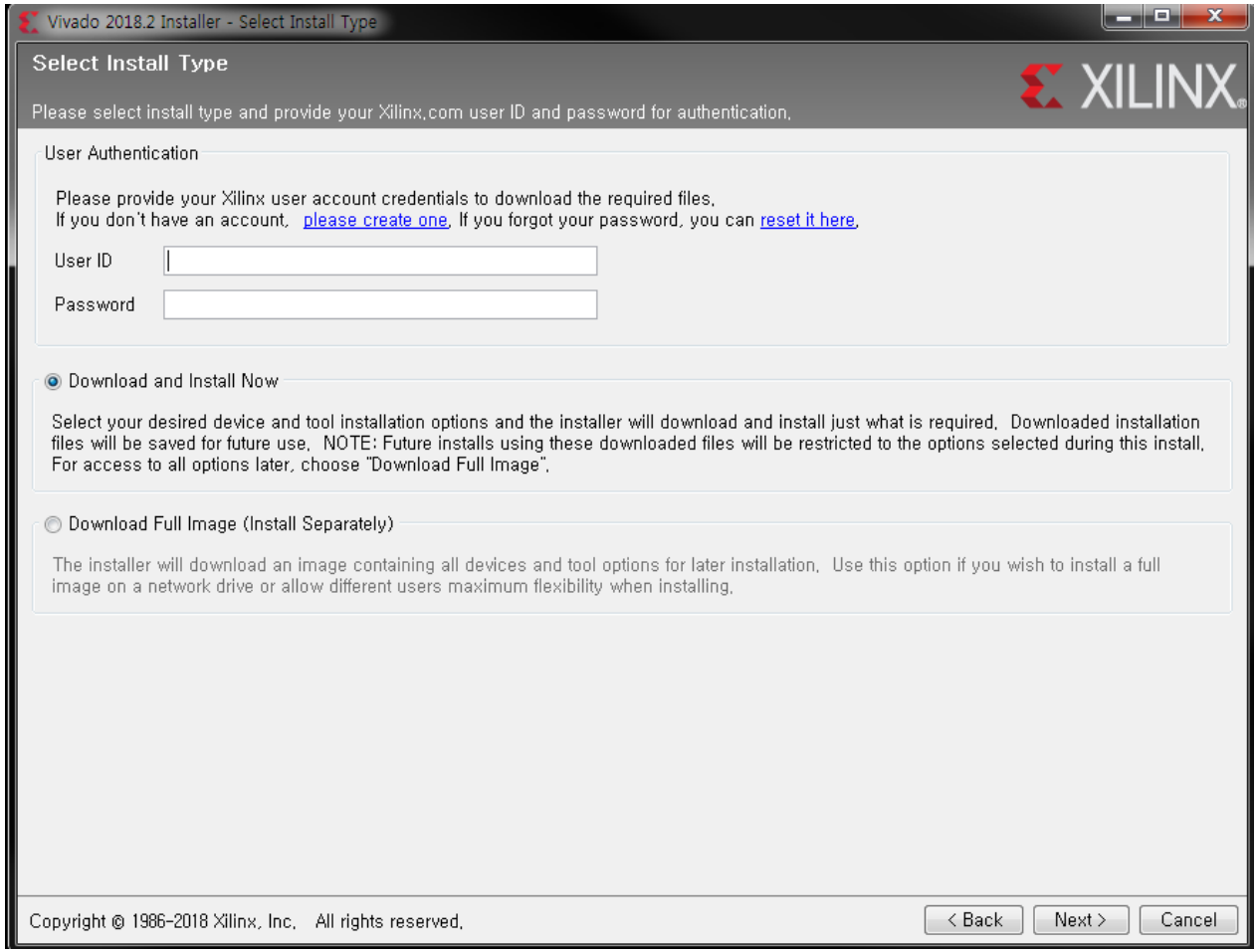


HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기

✓ User ID와 Password 창에 기입 후 Next 버튼을 클릭한다.

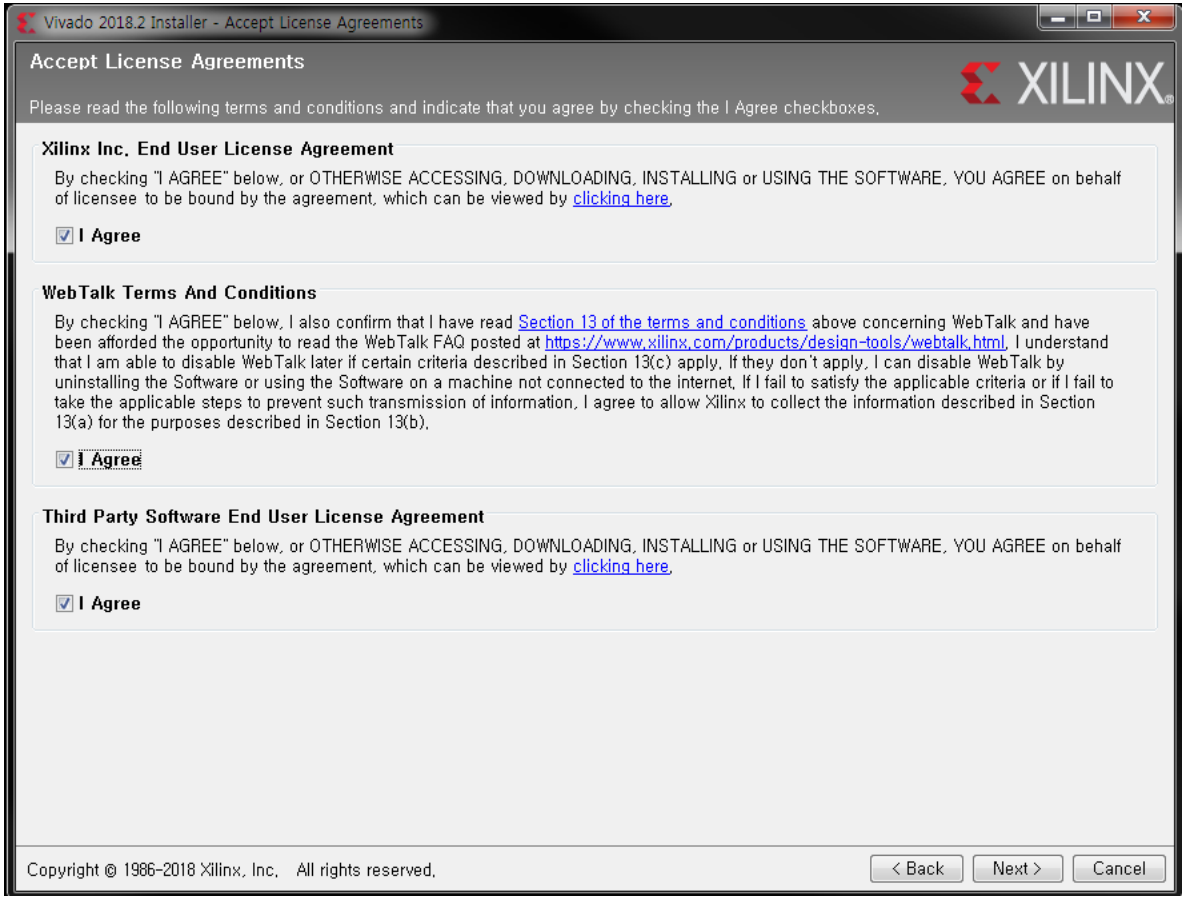


HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기

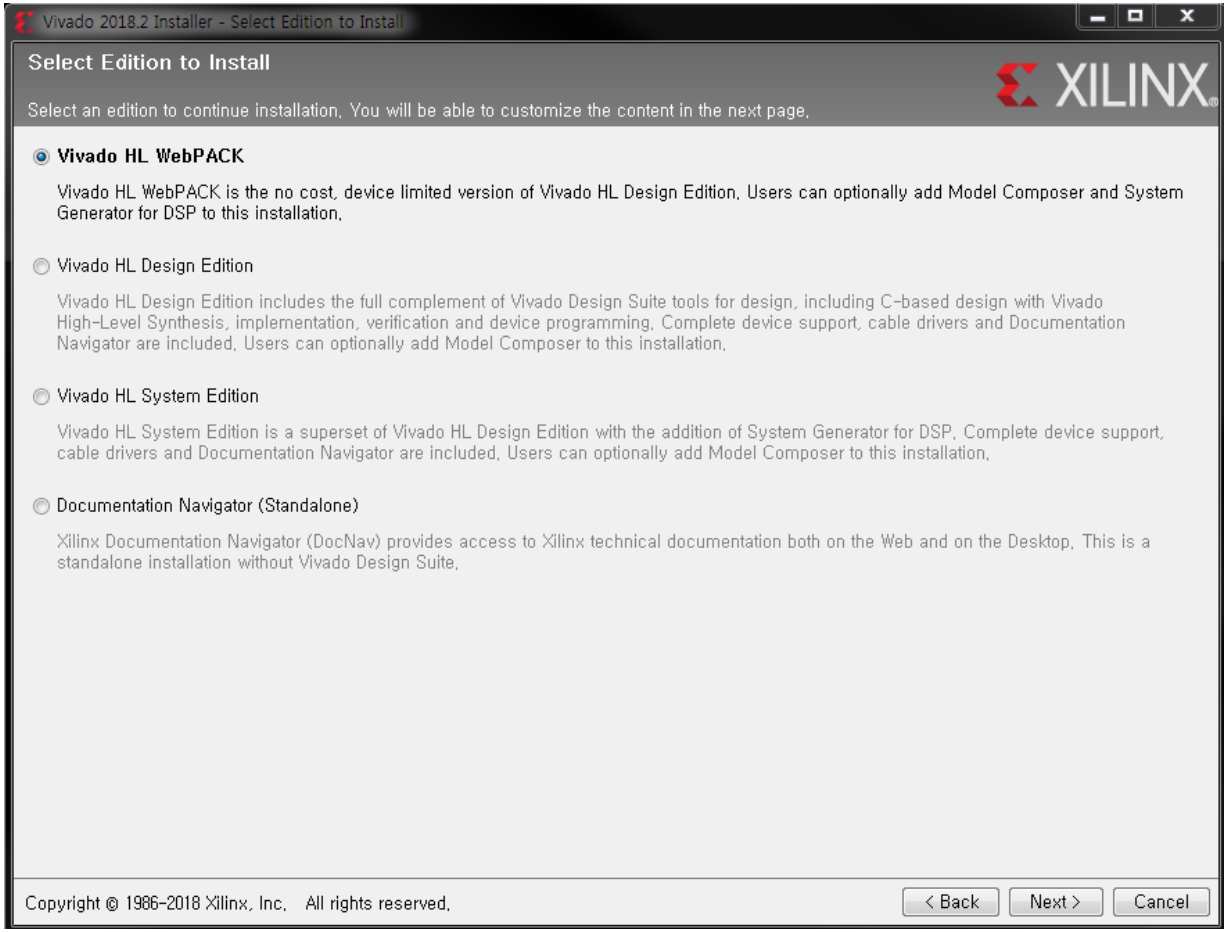
  ✓ License 관련 항목들에 대해 체크를 한 뒤 Next 버튼을 클릭한다.



HUINS

Vivado 설치 및 실행하기

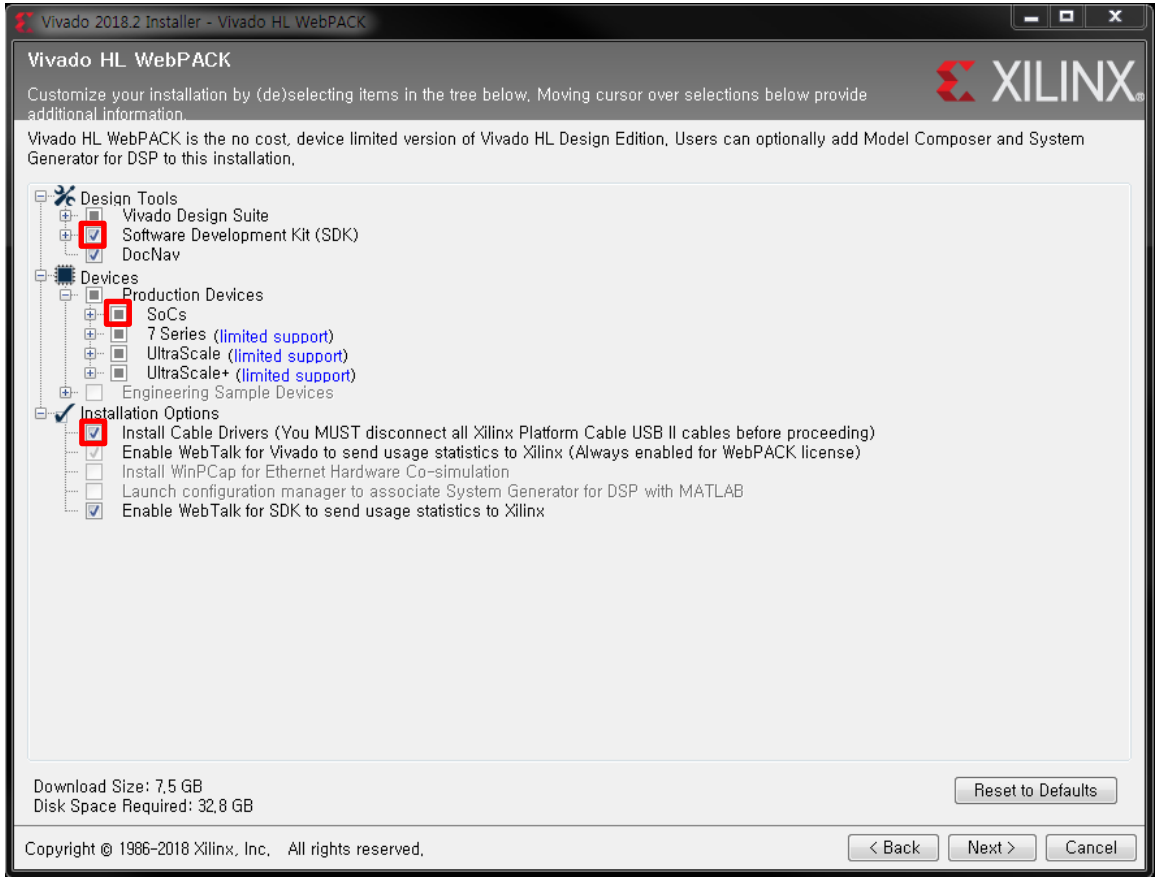
- Vivado WebPACK 설치하기
  - ✓ Vivado HL WebPACK을 선택하고 Next 버튼을 클릭한다.



HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기
  - ✓ 디자인 도구의 SDK와 디바이스의 SoCs, Cable Drivers를 모두 체크하여 설치를 진행한다.

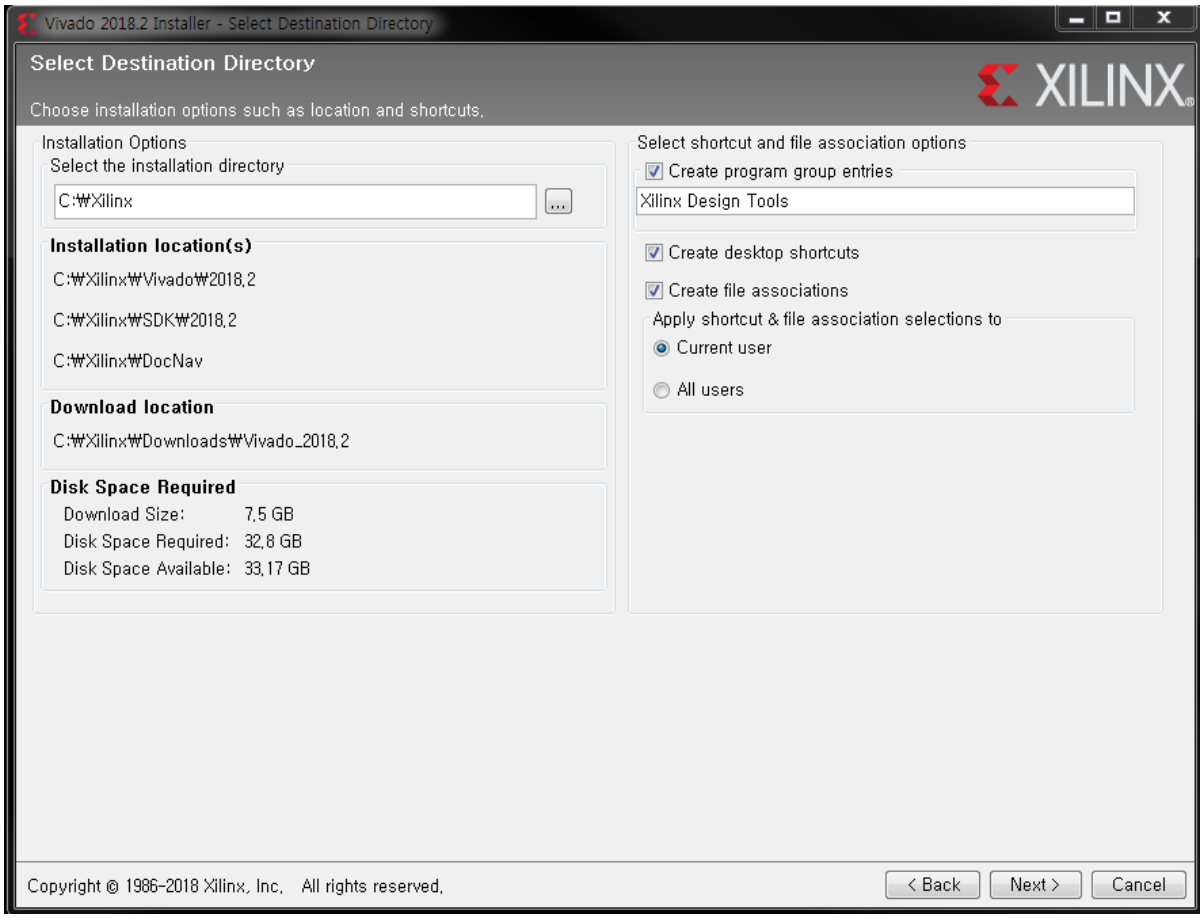


HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기

  ✓ Vivado HL WebPACK를 설치할 경로를 지정한다.



HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기

  ✓ 지정한 항목들을 확인하고 설치를 실행한다.

HUINS

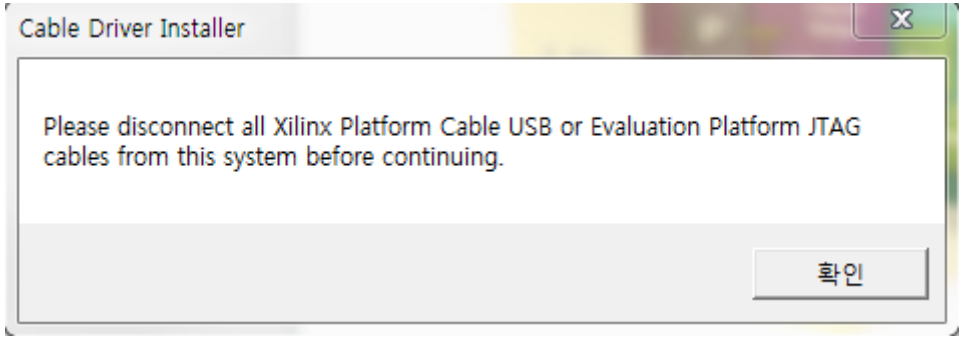
Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기
  - ✓ 설정에 맞춰 설치가 시작된다.

HUINS

Vivado 설치 및 실행하기

- Vivado WebPACK 설치하기
  - ✓ 설치 중 케이블 드라이버 설치 시 Xilinx 보드와 연결된 JTAG 또는 USB 케이블을 제거한 뒤 USB드라이버를 설치한다.

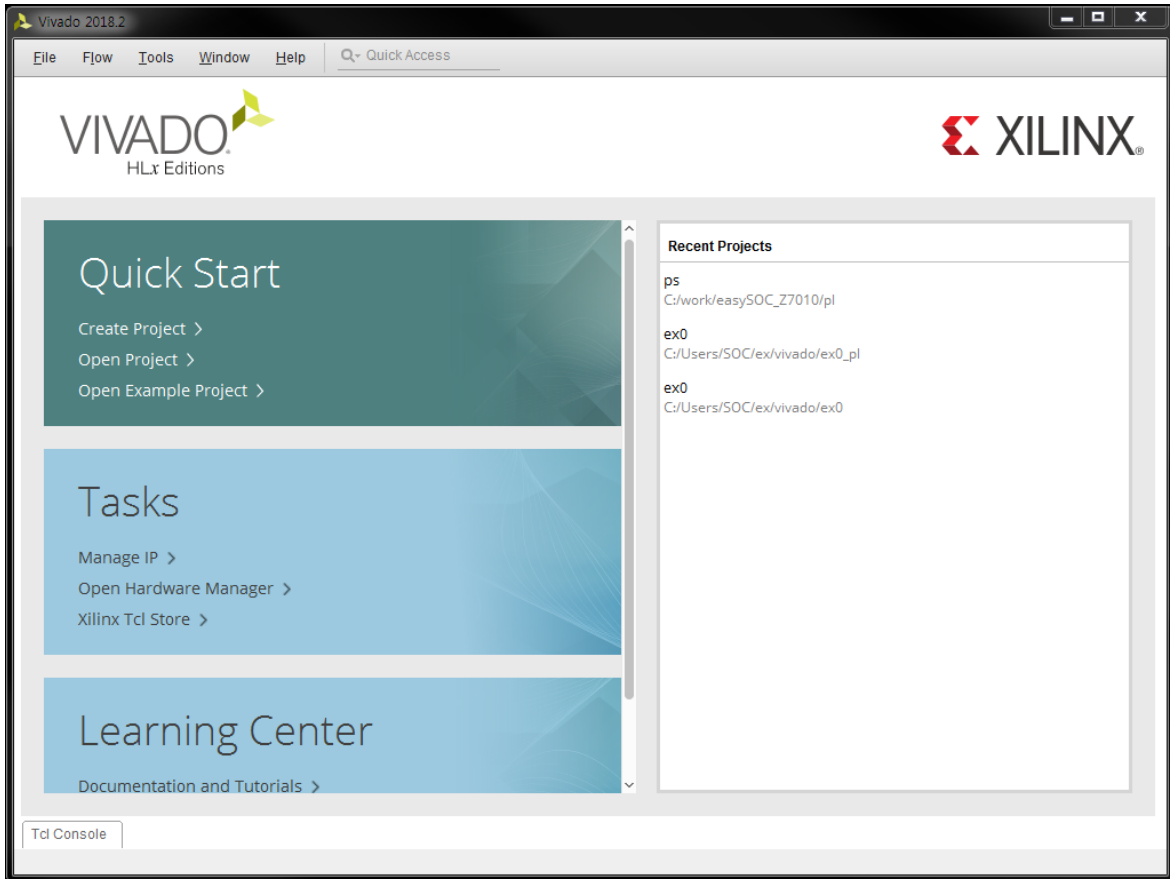


HUINS

Vivado 설치 및 실행하기

- Vivado 실행하기

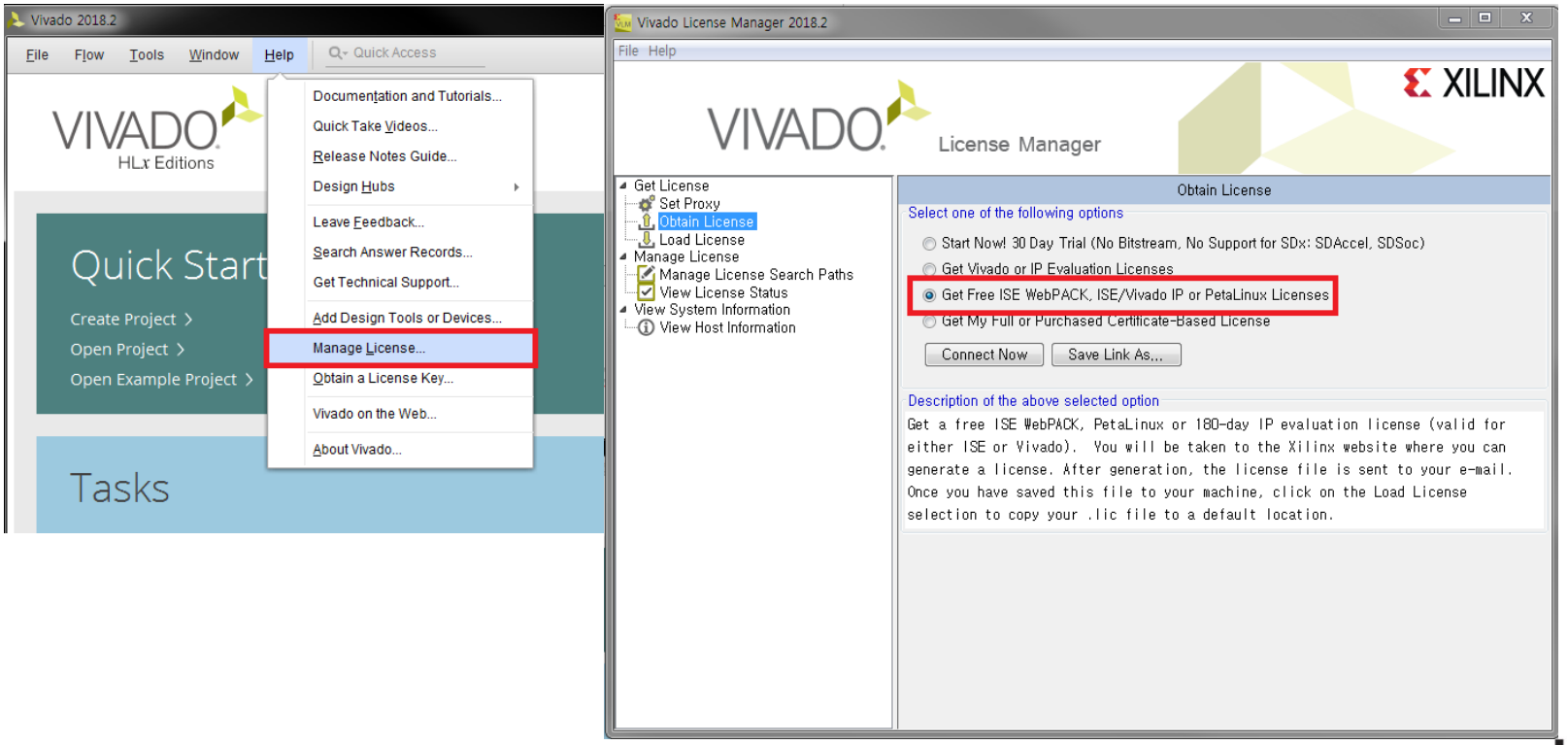
  ✓ 바탕화면에 생긴 Vivado 2018.2 실행파일 또는 시작메뉴에서 Vivado 디자인 툴을 클릭해 정상적으로 실행됨을 확인할 수 있다.



HUINS

Vivado 설치 및 실행하기

- License 인증
  - ✓ Vivado Design Suite에서 Help – Manage License를 들어가 라이선스 매니저를 실행한다.
  - ✓ Get Free ISE WebPACK, ISE/Vivado IP or Petalinux Licenses를 선택하여 Connect Now 버튼을 클릭한다.

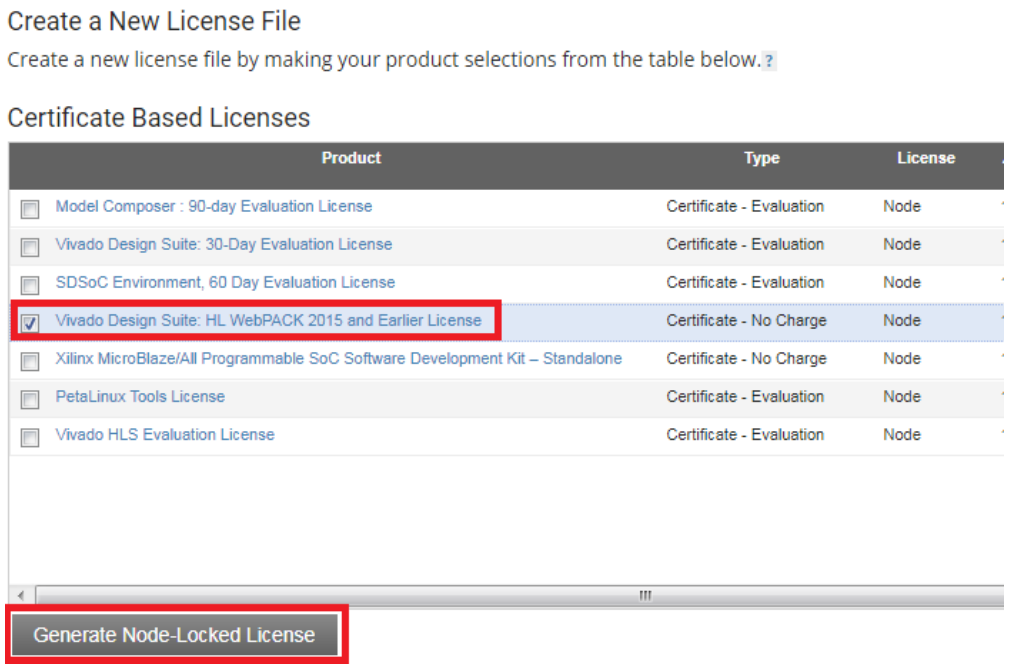


HUINS

Vivado 설치 및 실행하기

- License 인증

  ✓ Vivado Design Suite: HL WebPACK 2015 and Earlier Licenses를 선택하여 Node-Locked License를 생성한다.

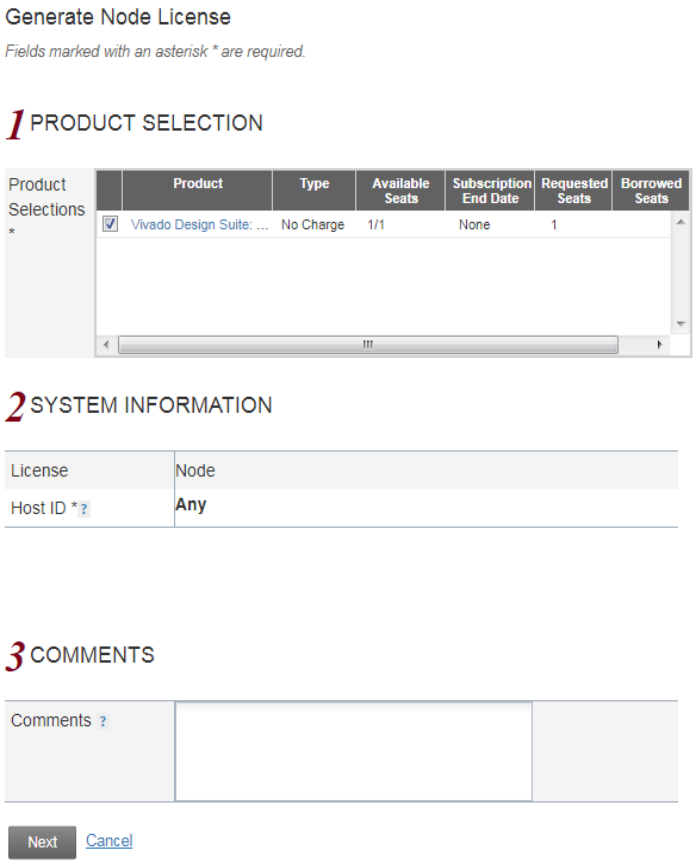


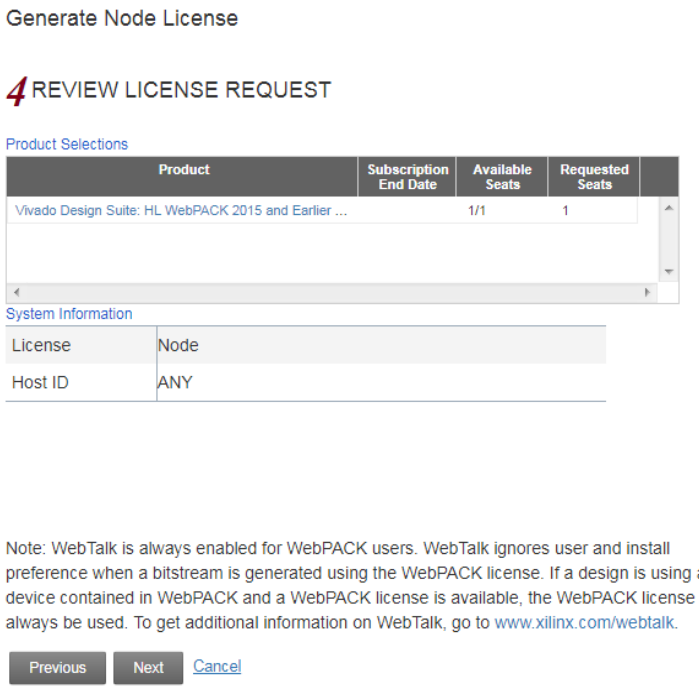
HUINS

Vivado 설치 및 실행하기

- License 인증

  ✓ Next 버튼을 클릭해 라이선스를 생성한다.



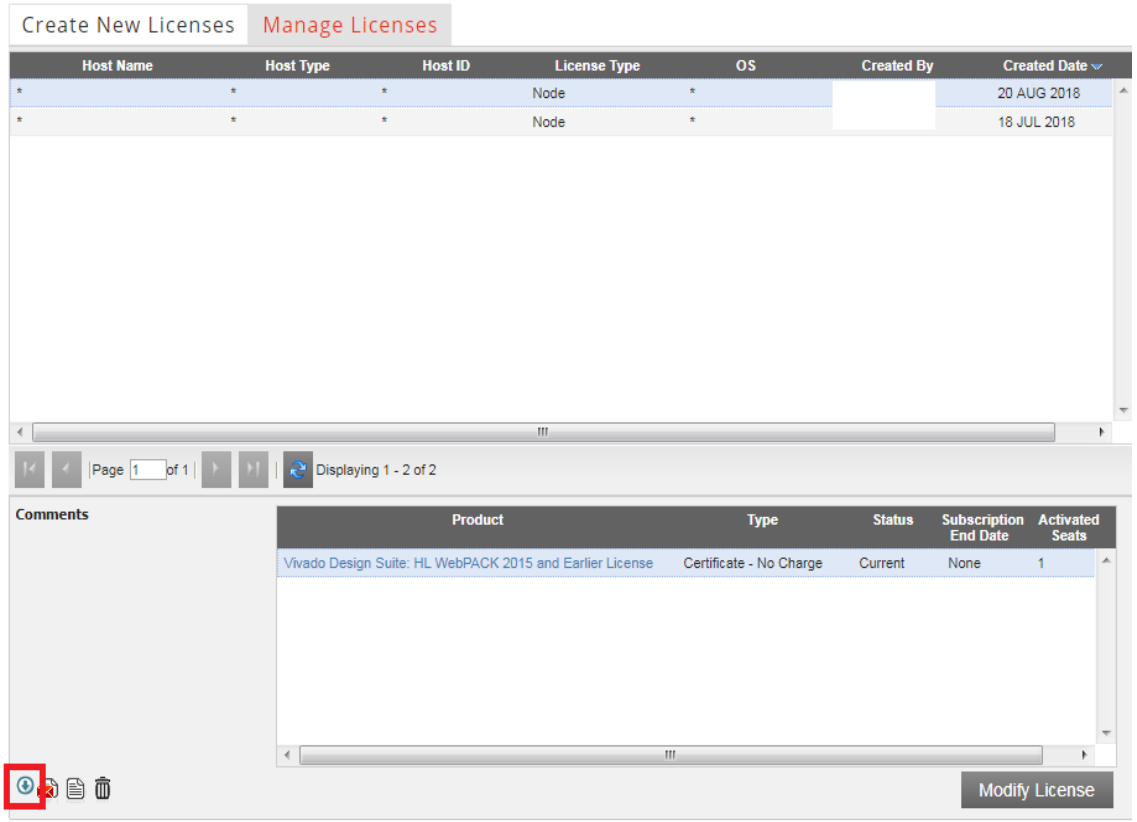


HUINS

Vivado 설치 및 실행하기

- License 인증

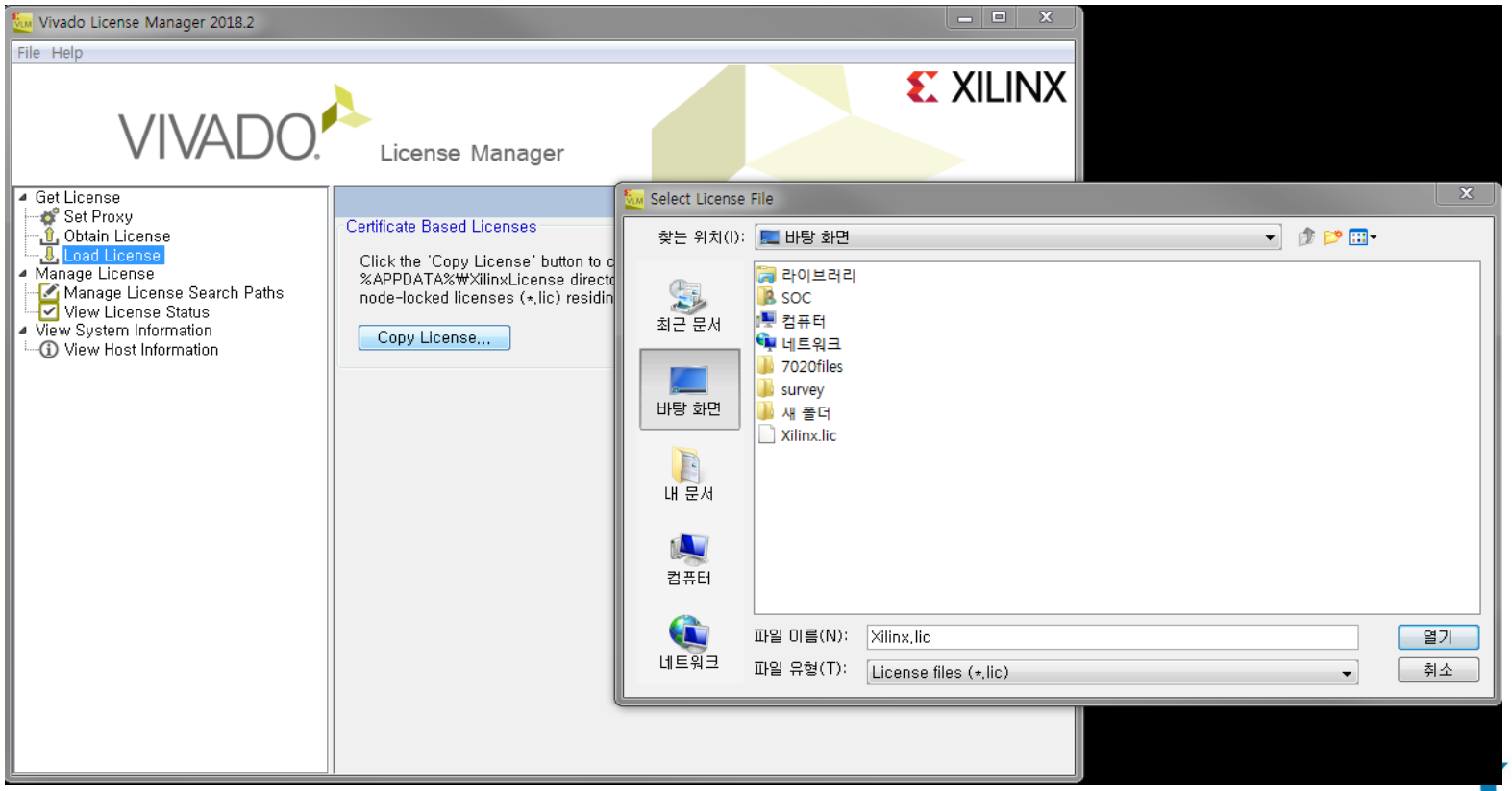
  ✓ 생성된 라이선스 파일은 Xilinx 계정에 등록된 email로 발송되며, 또는 그림과 같이 Manage Licenses 탭에서 해당 라이선스를 선택하여 파일로 다운로드 받을 수 있다.



HUINS

Vivado 설치 및 실행하기

- License 인증
  - ✓ Vivado License Manager에서 Load License탭에서 다운받은 라이선스를 등록한다.



HUINS

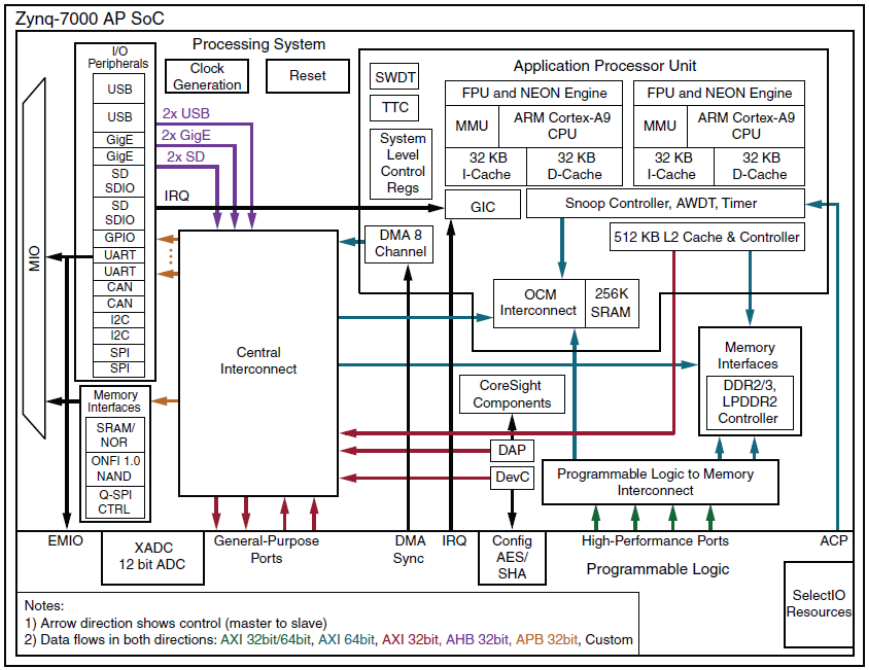
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Micro USB를 통한 Xilinx USB Platform 동작 제어
  - 호스트 컴퓨터가 EasySoC-Z7010을 제어할 때 필수적인 장치.
  - 해당 플랫폼의 경우 micro USB 연결을 통해 Xilinx USB platform 동작 수행.
- Xilinx USB Platform Cable 주요 역할
  - Zynq Processing System7(이하 PS)을 제어
    - ✓ 동작시킬 F/W를 DDR3 메모리에 올리는 역할.
    - ✓ DDR3 메모리에 올라가 있는 F/W를 PS가 동작시키도록 함.
    - ✓ F/W가 동작할 때, 디버깅 기능 제공.
  - Zynq Programmable Logic(이하 PL)을 제어
    - ✓ 합성 된 HDL 하드웨어를 PL에 다운로드 하는 역할.
    - ✓ PL에 다운로드 된 HDL 하드웨어 내부 신호를 probing 하는 역할.
  - Vivado Hardware Co-Simulation
    - ✓ PS는 Zynq 칩에서 동작, PS에 연결된 User HDL 코드는 vivado에서 시뮬레이션 할 때, PS에서 신호를 받거나 PS에 신호를 입력하는 역할을 수행.

HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

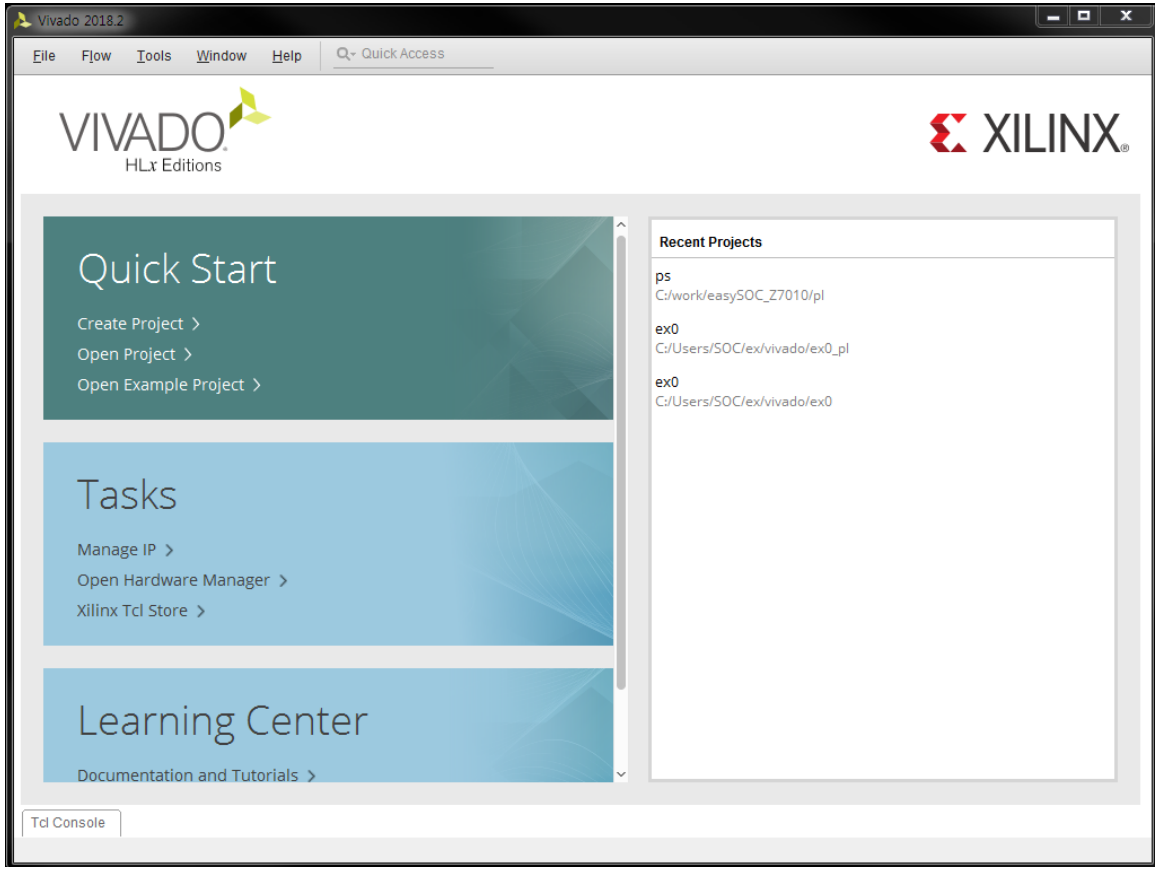
- Processing System 소개
  - Processing System
    - ✓ Xilinx 사에서 제작한 임베디드 시스템.
    - ✓ Zynq 칩에 포함되어 programmable logic과 연결되어 동작함.
  - Processing System 특징
    - ✓ ARM dual Cortex-A9 프로세서가 포함.
    - ✓ 임베디드 시스템을 구성하는데 필요한 IP들이 다수 포함.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

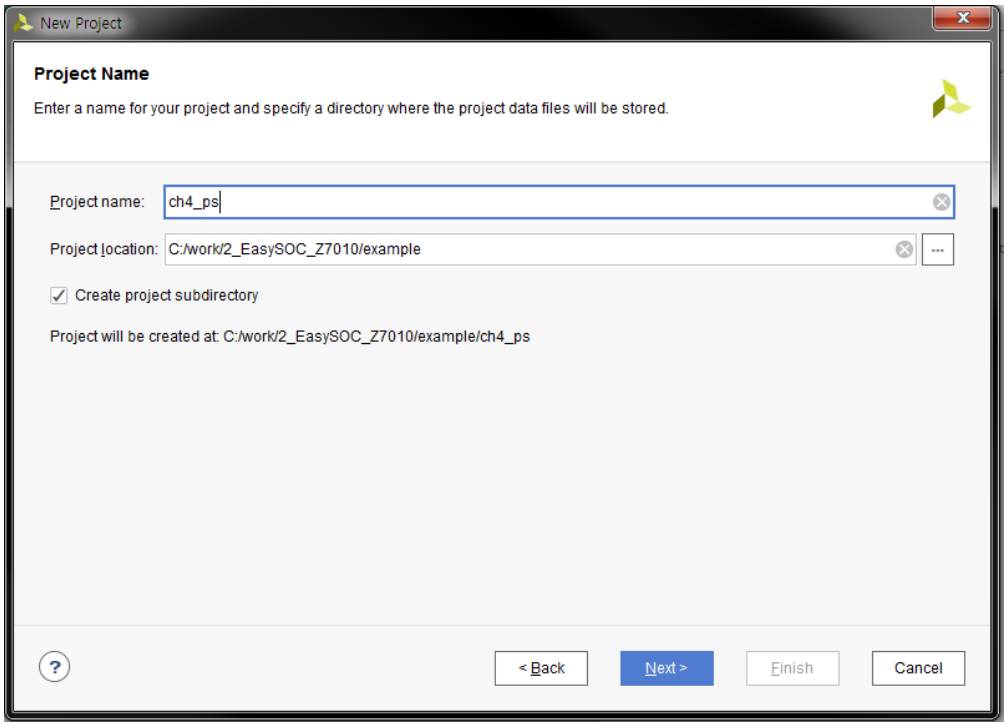
- Processing System을 이용한 실습
  - Project 생성
    - ✓ Vivado 실행 후 “Create New Project” 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

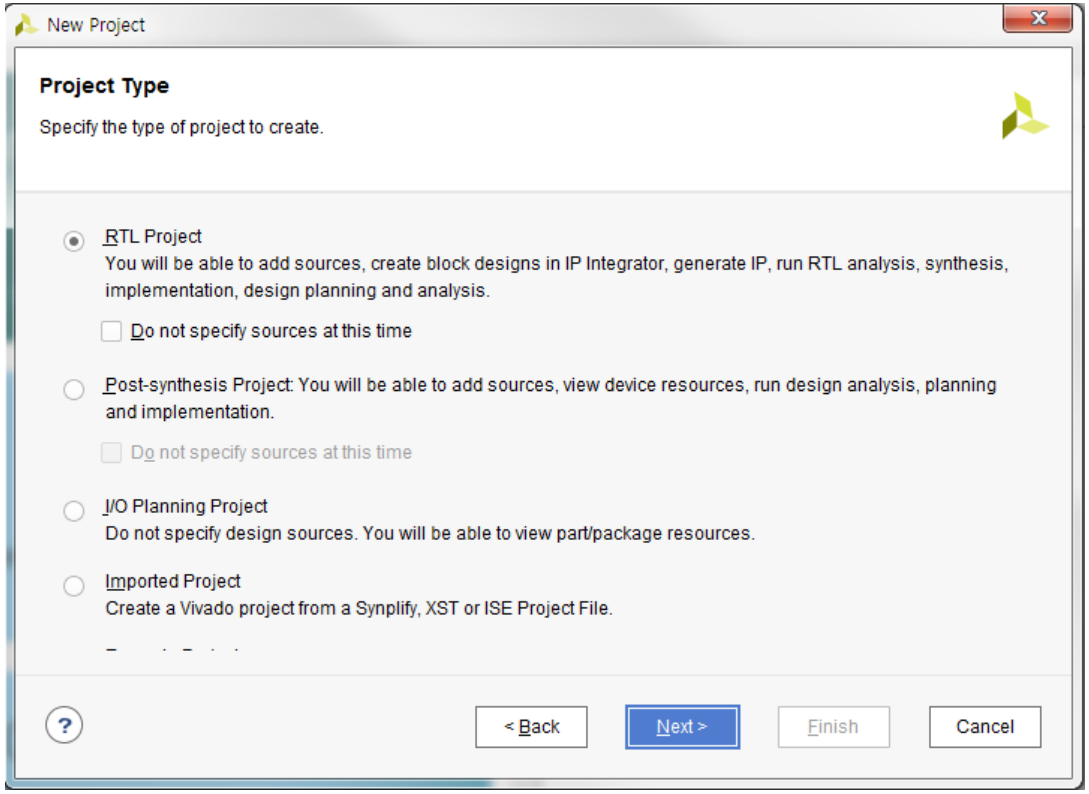
- Processing System을 이용한 실습
  - Project 생성
    - ✓ Project가 생성될 폴더와 이름을 입력한다. "Create project subdirectory" 항목 체크 후 Next 버튼 클릭한다.
    - ✓ "Create project subdirectory" 항목 체크 시, "Project location"에 입력 된 폴더 아래 "Project Name"에 입력된 이름으로 폴더가 생성되고, 그 폴더에 Project 파일들이 위치하게 된다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

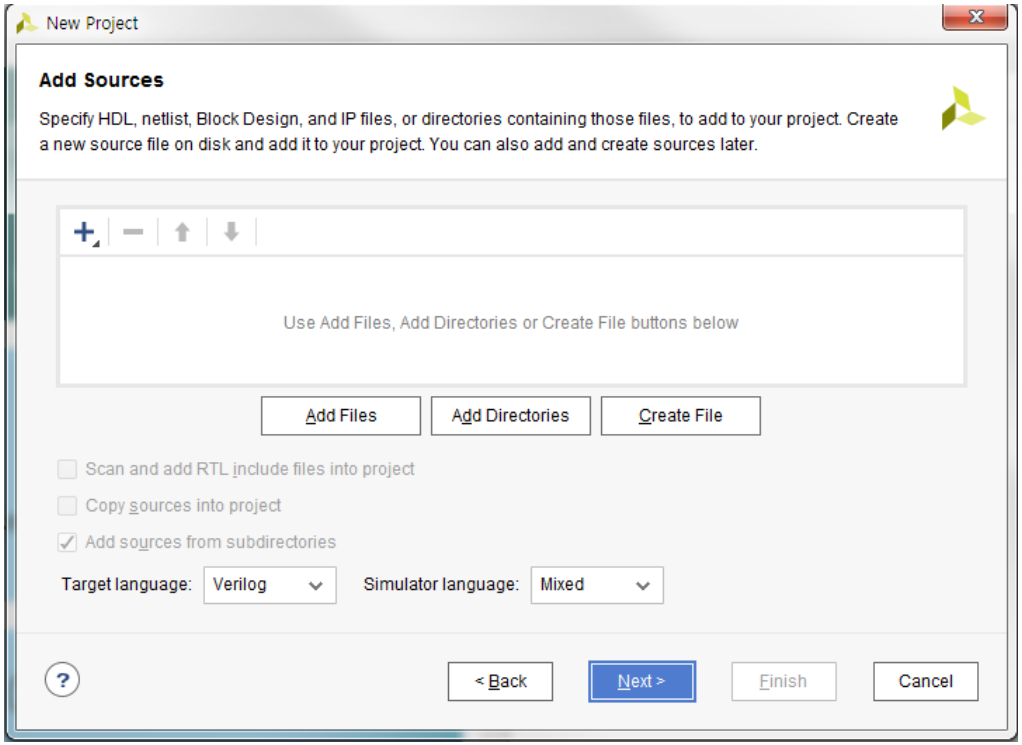
- Processing System을 이용한 실습
  - Project 생성
    - ✓ Project Type을 RTL Project로 선택하고 Next 버튼을 클릭한다.
    - ✓ RTL Project는 HDL Code를 생성하거나 수정한 후 synthesis와 implement를 할 수 있다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

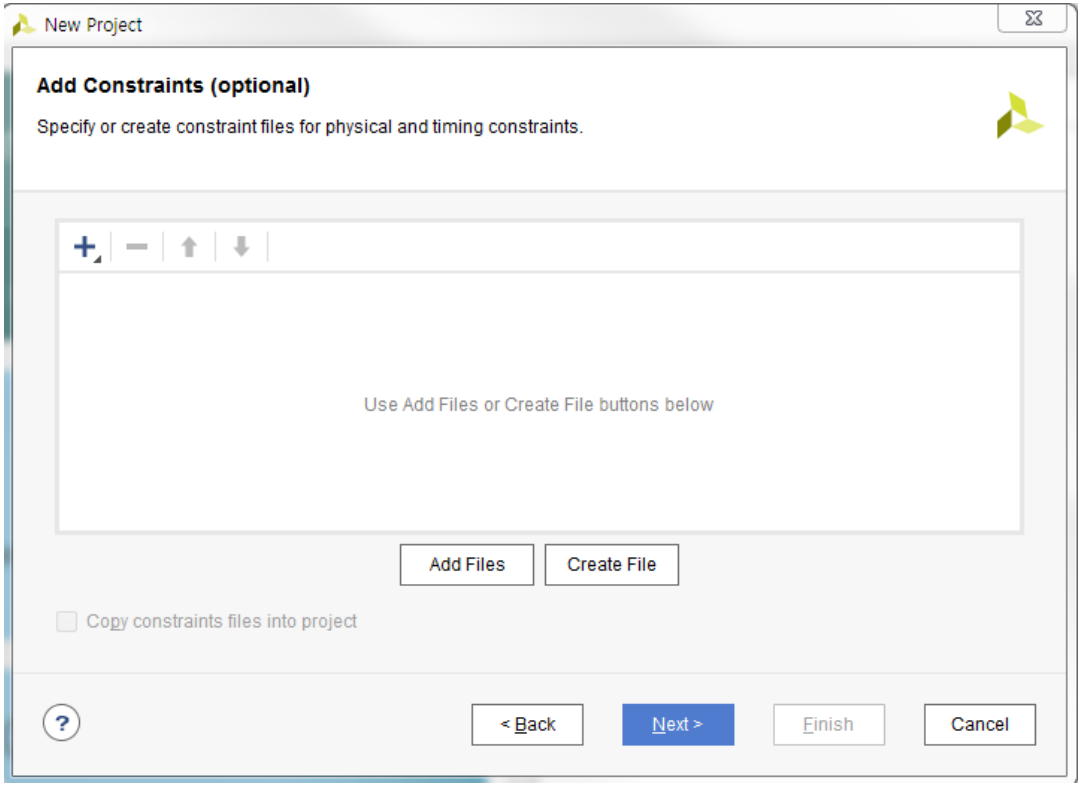
- Processing System을 이용한 실습
  - Project 생성
    - ✓ HDL code를 생성하거나 추가하는 단계이다.
    - ✓ Target Language 항목을 Verilog로 하고 Next 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

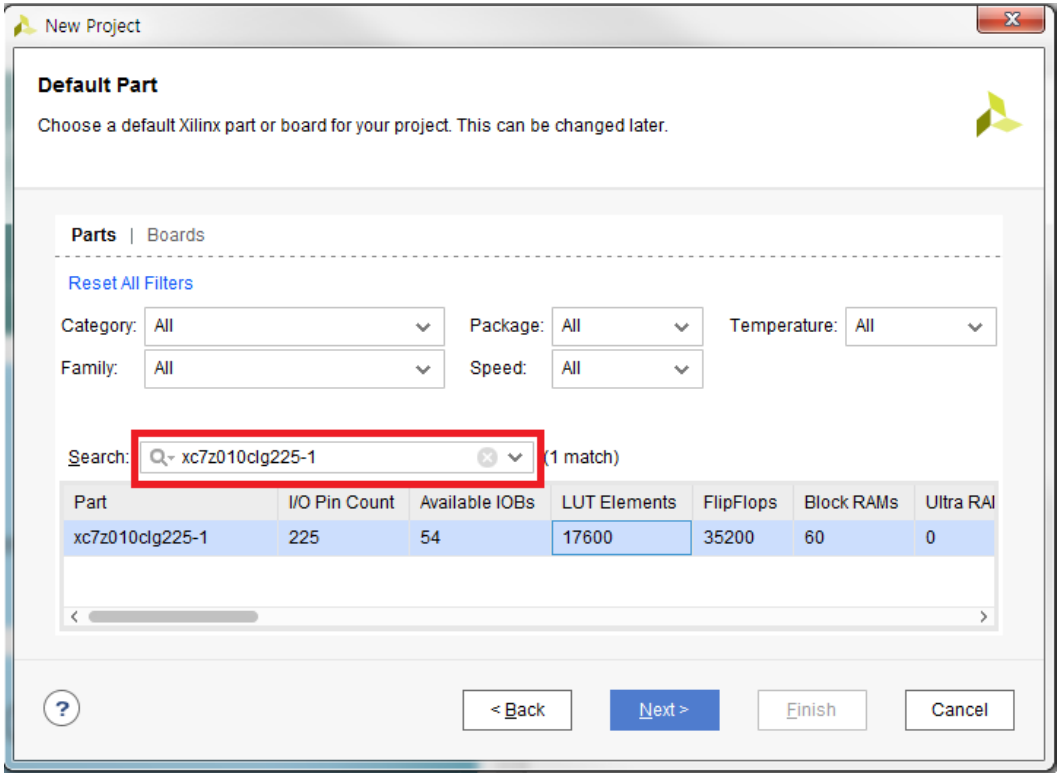
- Processing System을 이용한 실습
  - Project 생성
    - ✓ Constraint 파일(.xdc)을 생성 또는 추가하는 단계이며, Next 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

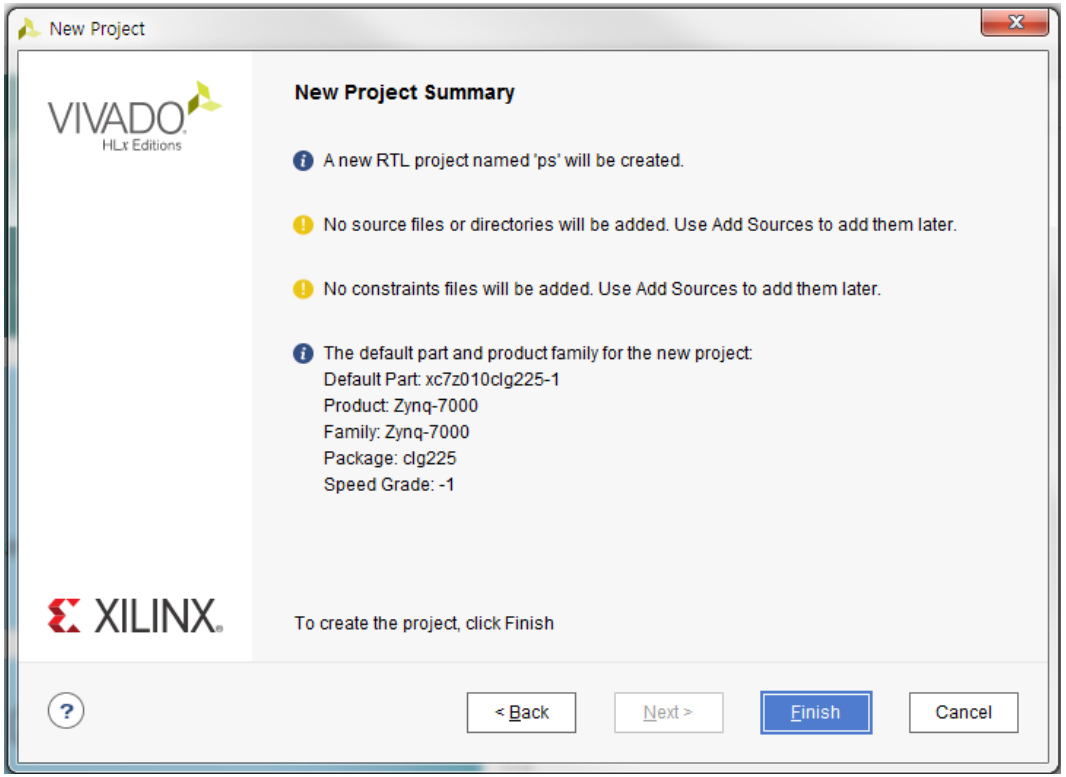
- Processing System을 이용한 실습
  - Project 생성
    - ✓ Xilinx FPGA 또는 Zynq를 선택하는 단계이다.
    - ✓ Search 입력란에 xc7z010clg225-1를 입력한 후 해당 파트를 선택한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

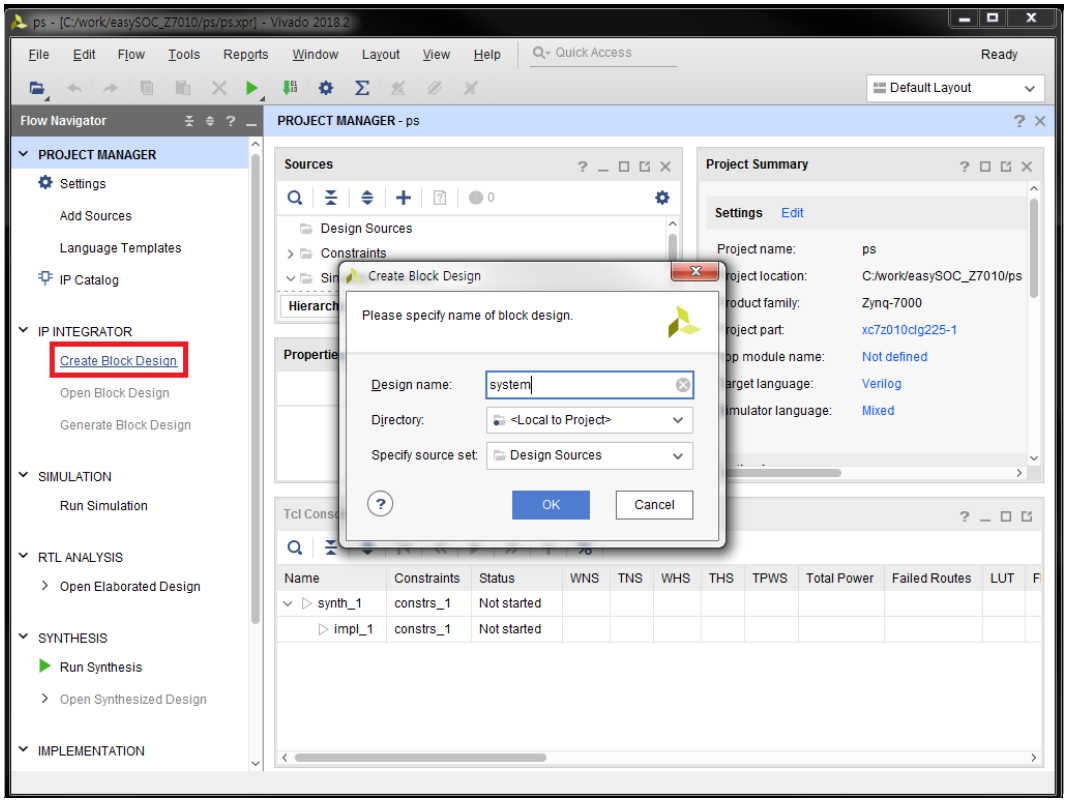
- Processing System을 이용한 실습
  - Project 생성
    - ✓ 설정한 부분들에 대해서 최종적으로 확인 후 Finish 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

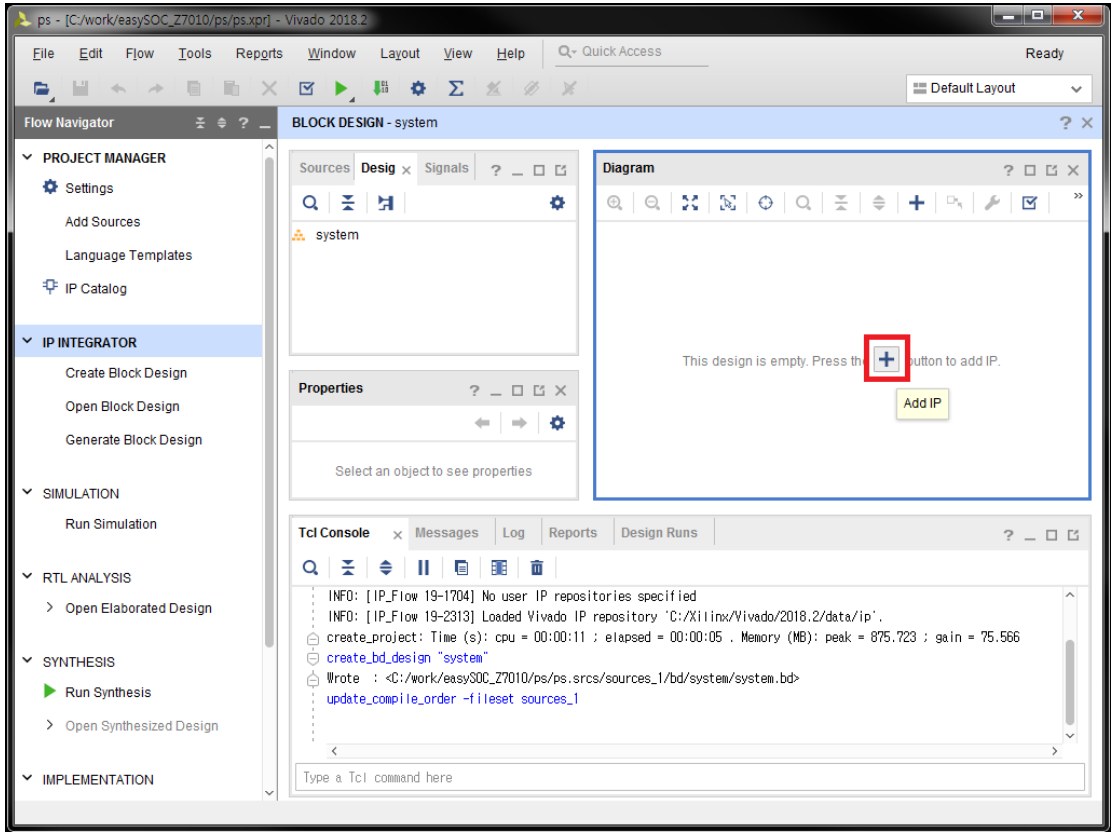
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Flow Navigator 창에서 Create Block Design 을 클릭한다.
      - – Xilinx IP(*.xco, *.xci)인 Processing System(Cortex-A9)을 추가하기 위함.
    - ✓ 생성하고자 하는 Block의 이름을 "system"으로 지정하고 OK 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

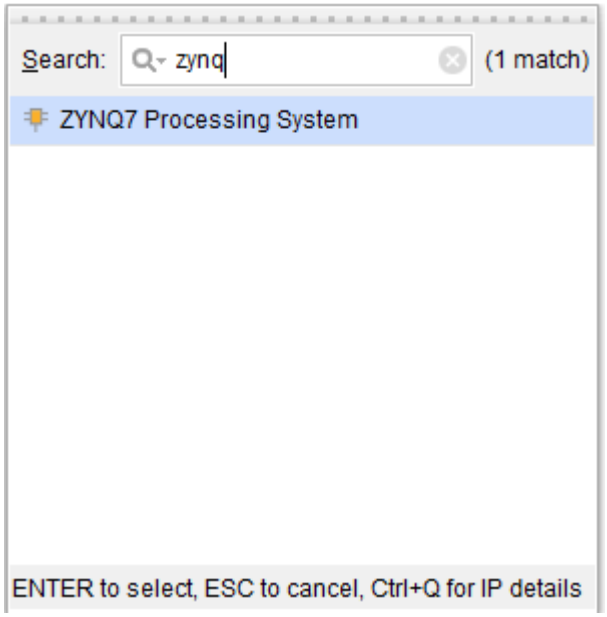
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ 생성 된 Diagram 서브 창에서 "Add IP"아이콘을 클릭한다.
    - ✓ Diagram 서브 창이 나오지 않을 경우 Design Hierarchy / Source 서브 창에서 system 항목을 더블클릭 하여 Diagram 서브 창을 띄운다.



HUINS

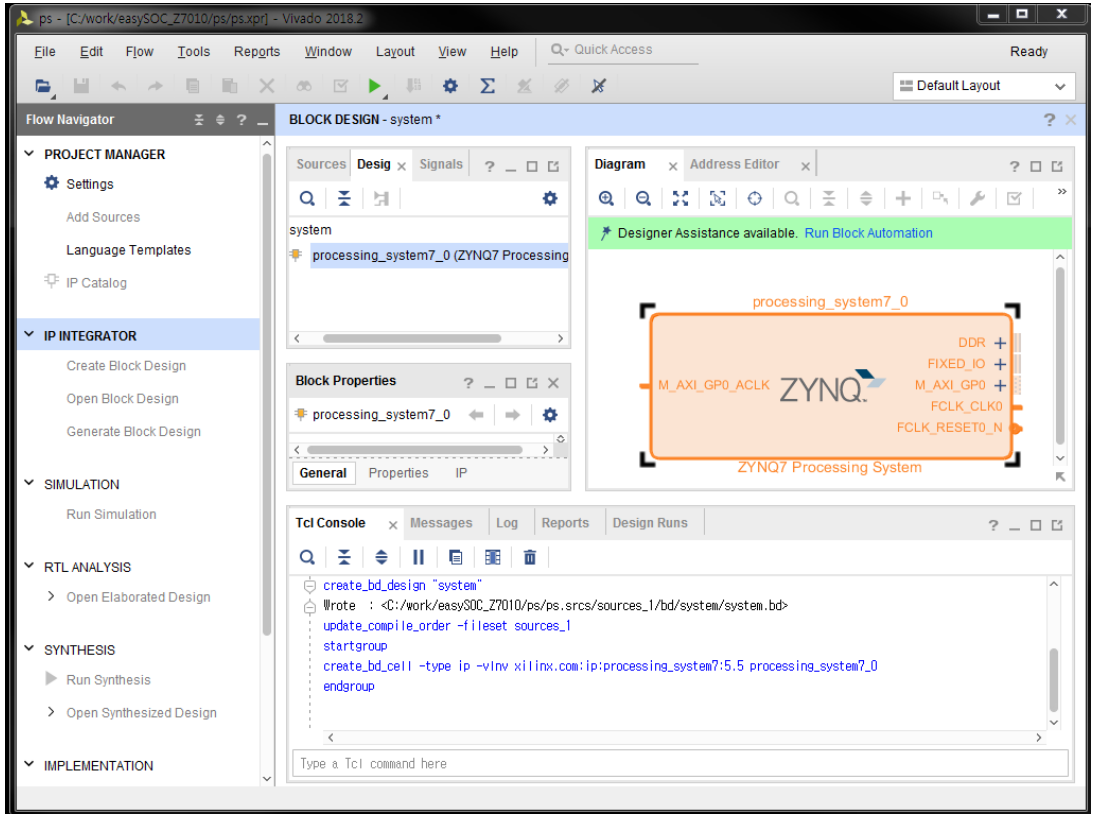
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Xilinx IP를 추가하는 대화상자의 Search 입력란에 "zynq"를 입력하여 ZYNQ7 Processing System 항목을 선택하여 추가한다.



Processing System을 포함한 프로젝트 생성 및 동작시키기

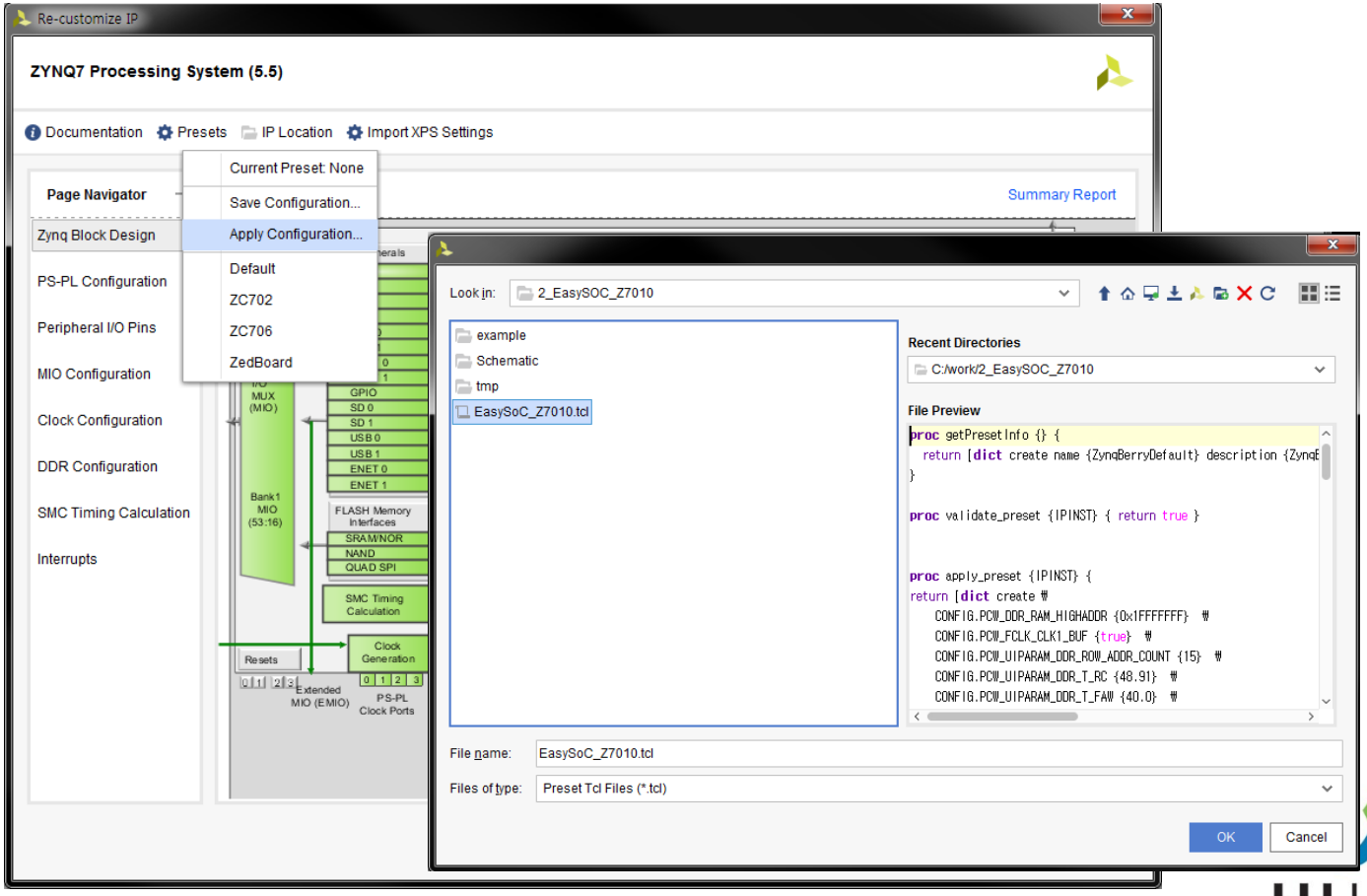
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Zynq Processing System 영역을 설정하기 위해, Diagram 서브 창에서 생성된 ZYNQ7 Processing System 블록을 더블클릭 한다.



HUINS

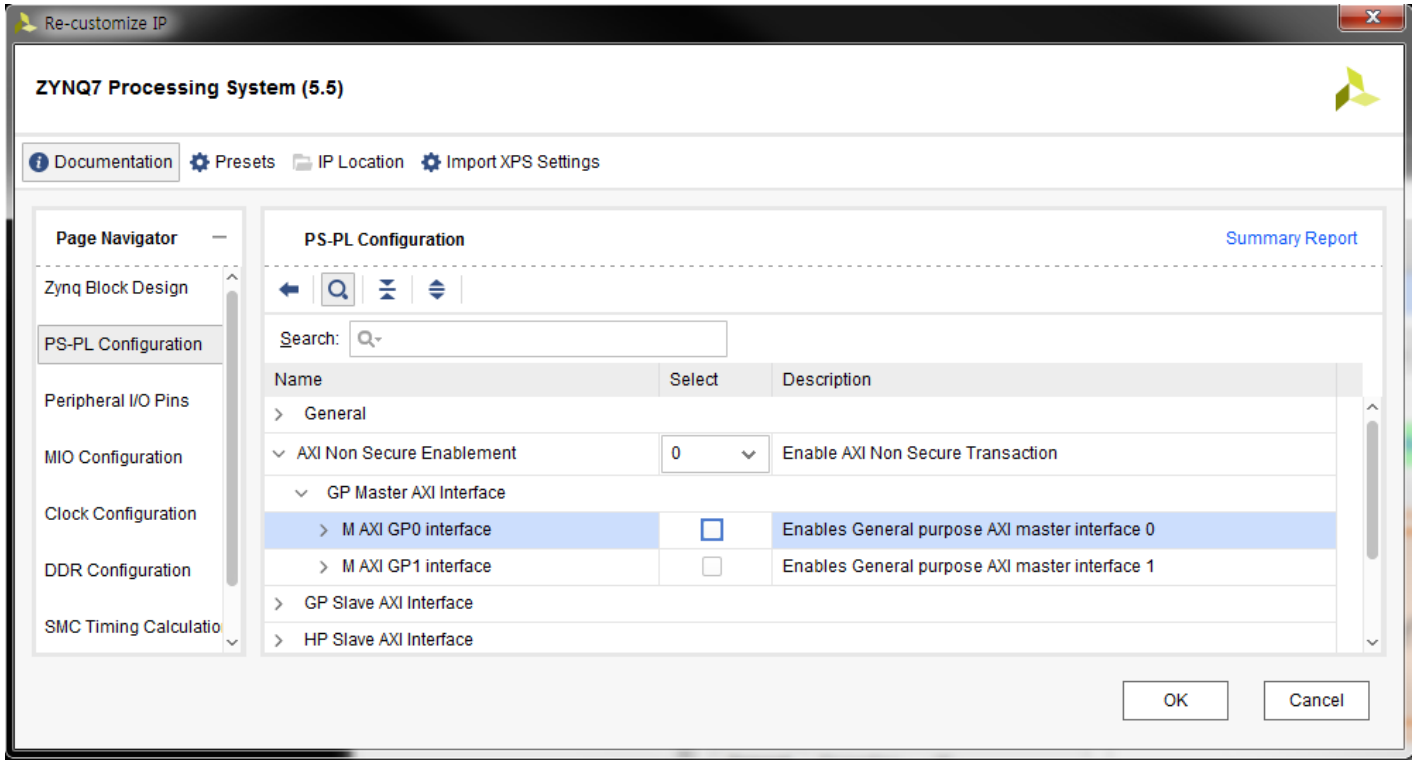
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Processing System 추가
  - Presets – Apply Configuration에 EasySoC_Z7010.tcl 파일을 적용한다.



Processing System을 포함한 프로젝트 생성 및 동작시키기

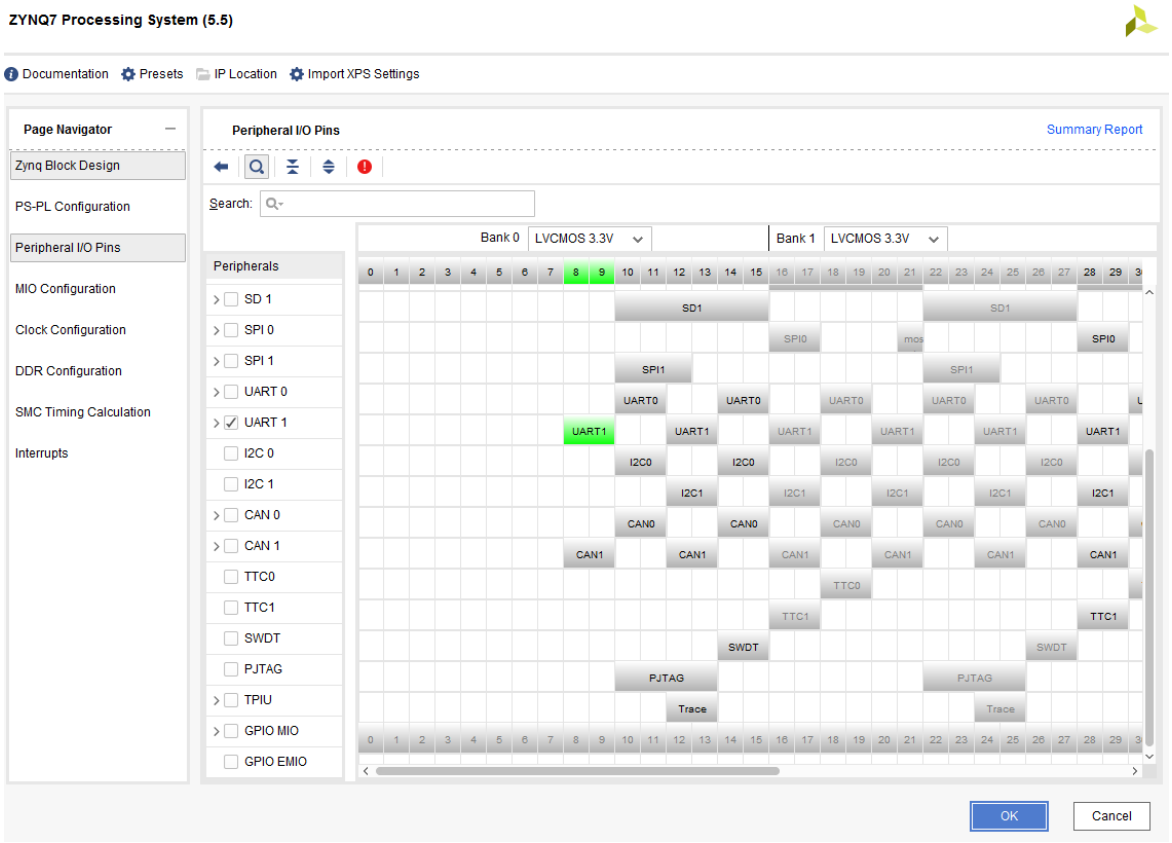
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ 이번 예제에서는 AXI 주변장치를 사용하지 않으므로 Page Navigator에 있는 PS-PL Configuration 항목에서 M AXI GP0 Interface를 체크 해제 한다.



HUINS

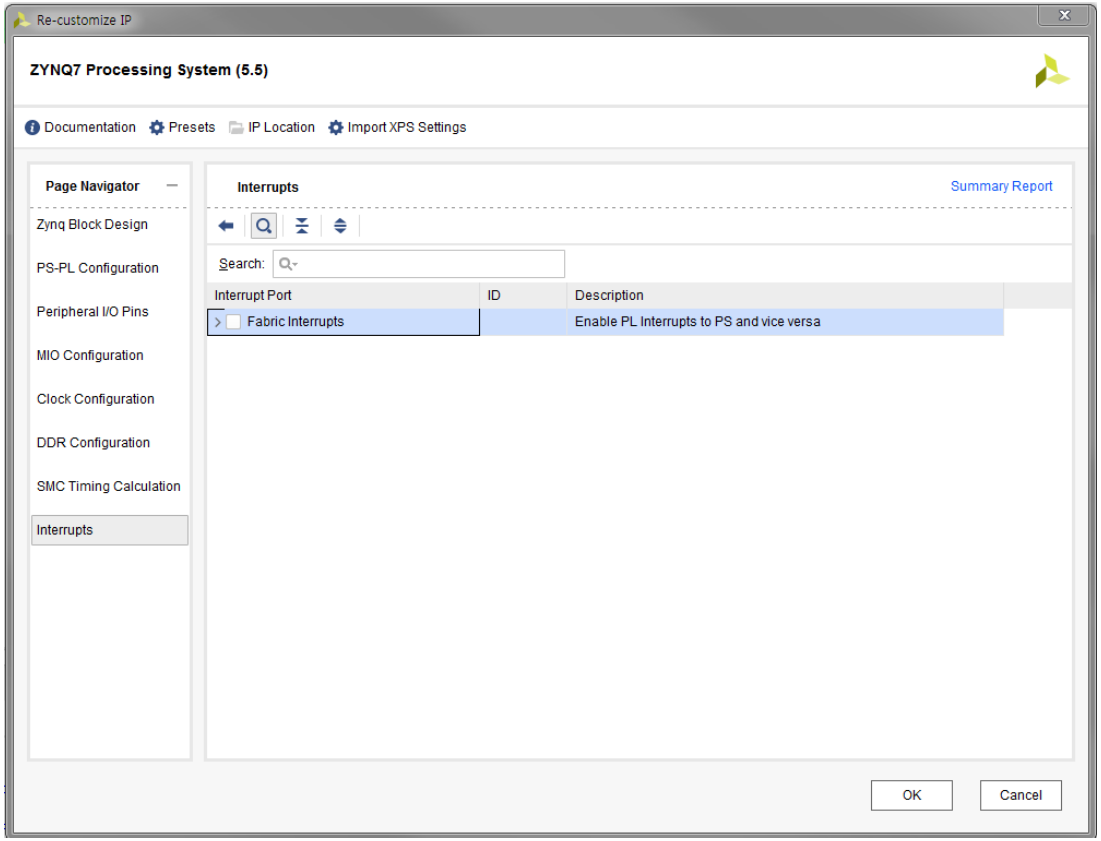
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ 본 예제에서는 MIO를 UART만 사용하므로 Page Navigator에 있는 Peripheral I/O Pins에서 UART 1(MIO 8, 9) 항목만 체크하고 모두 풀어준다.



Processing System을 포함한 프로젝트 생성 및 동작시키기

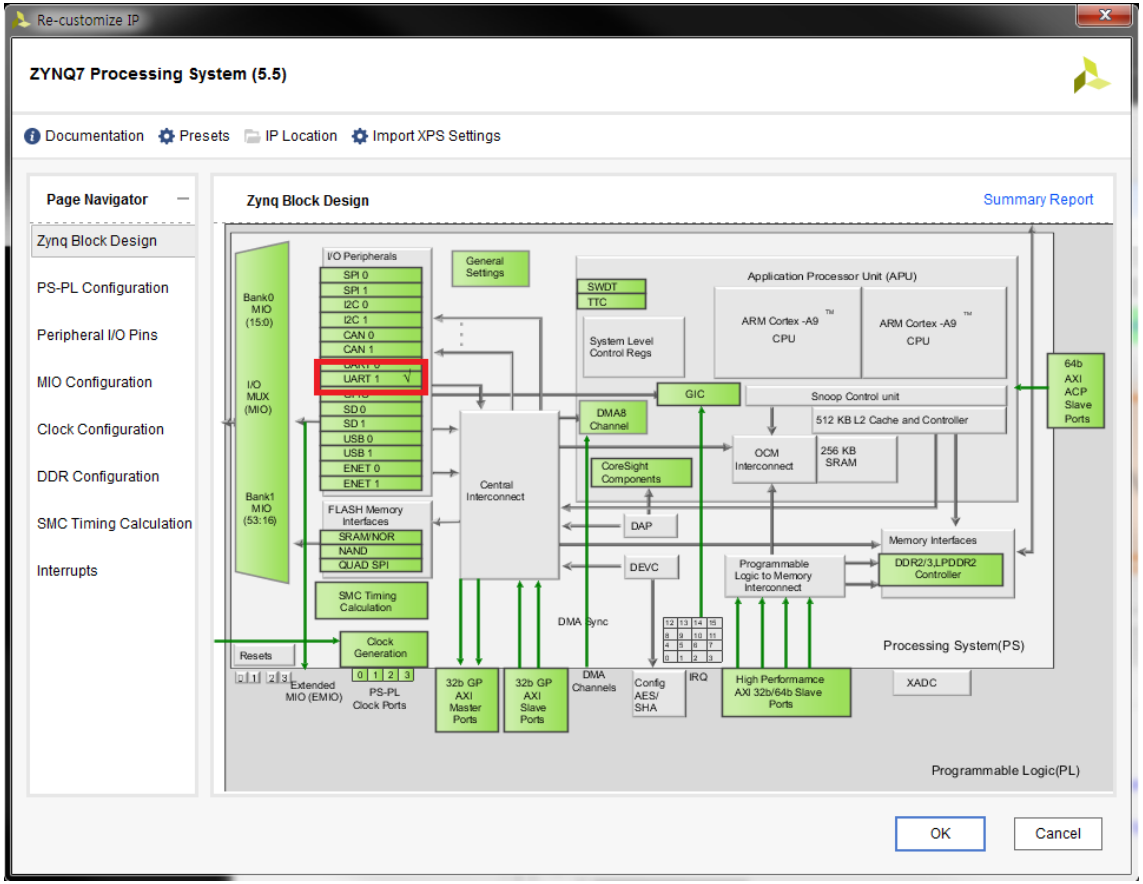
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Interrupts의 Fabric Interrupts를 체크 해제한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Page Navigator에 있는 Zynq Block Design에서 UART 1이 사용으로 체크되어 있는지 확인한 뒤 OK를 클릭해 Processing System의 설정을 끝낸다.



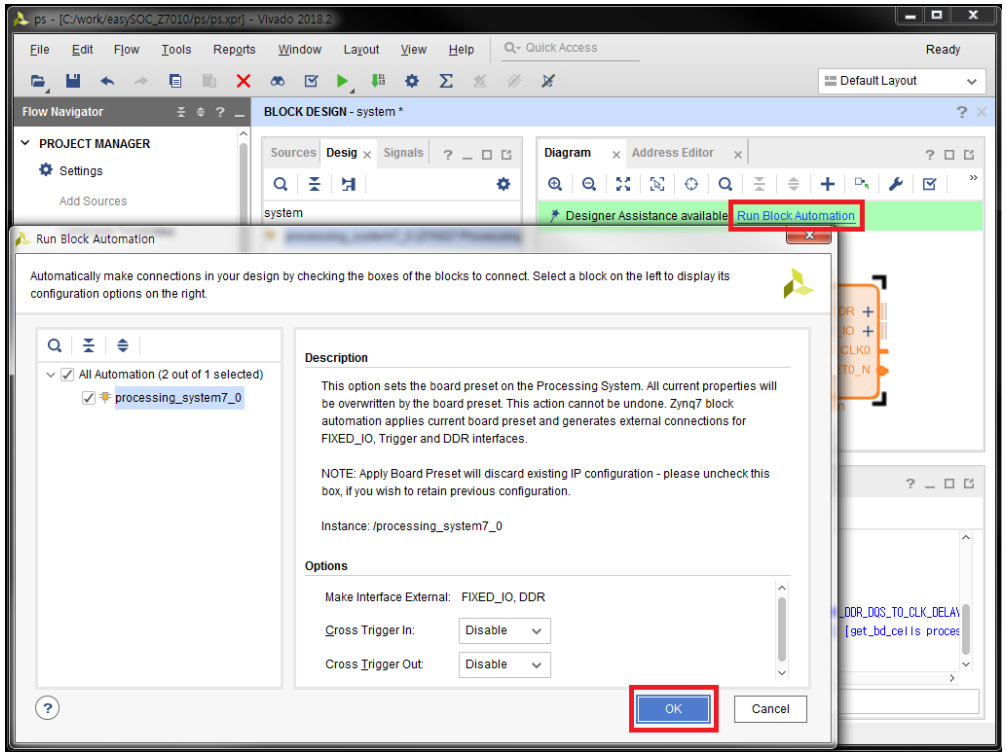
HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Processing System Configuration설정
  - Zynq 7010의 Preset을 적용하지 않을 경우 다음과 같이 설정한다.
    - ✓ PS-PL Configuration – GP Master AXI Interface
      - M AXI GP0 Interface 체크 해제
    - ✓ DDR Controller - DDR Controller Configuration
      - Memory Type : DDR 3(Low Voltage)
      - Memory Part MT41J256M16 RE-125
    - ✓ MIO Configuration – I/O Peripherals
      - UART 1 (MIO 8, 9) 사용

Processing System을 포함한 프로젝트 생성 및 동작시키기

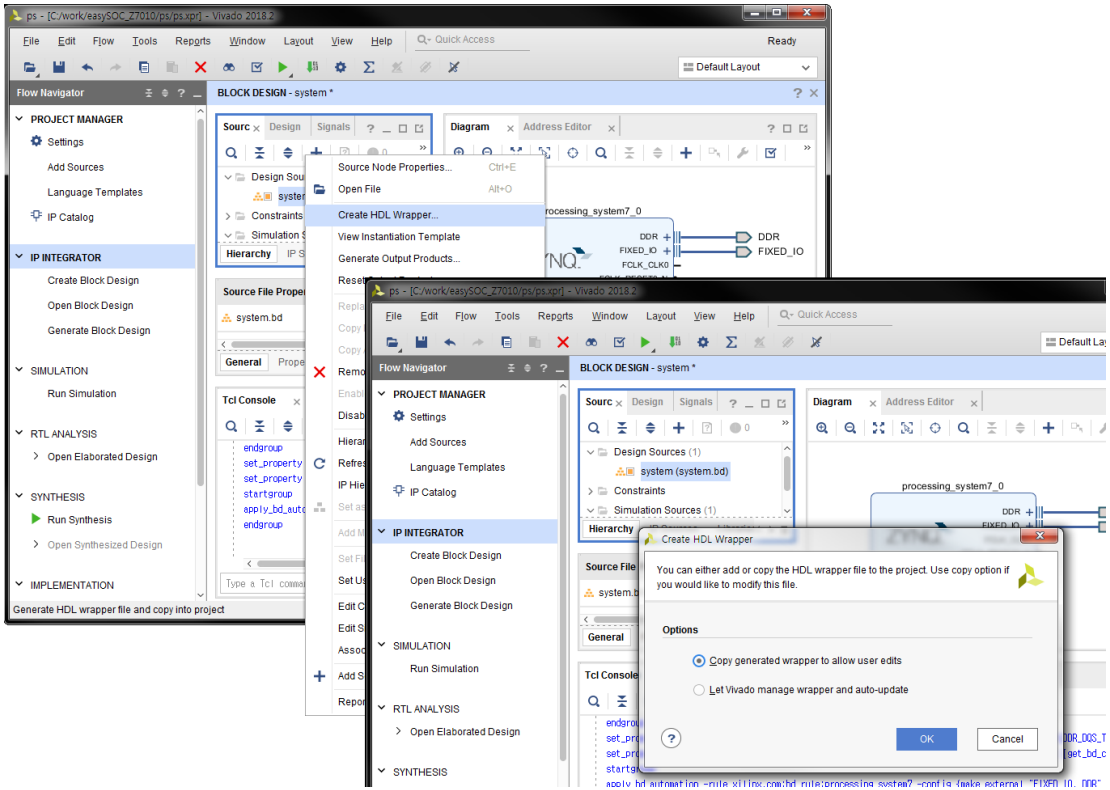
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ System block의 입출력 포트를 자동으로 생성하기 위해 Diagram 서브 창에서 Run Block Automation을 클릭한다.
    - ✓ /processing_system7_0 메뉴를 클릭 후 Cross Trigger in/Out이 모두 Disable 되어 있는지 확인한 후 OK 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

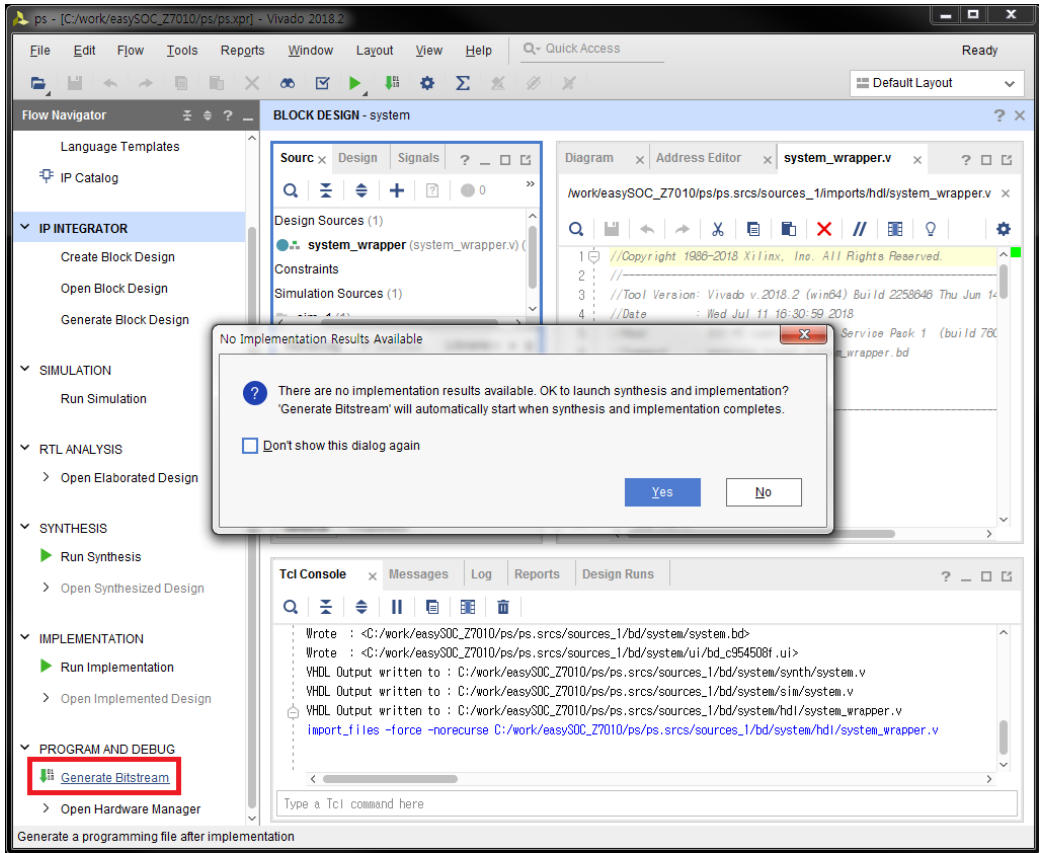
- Processing System을 이용한 실습
  - Processing System 추가
    - ✓ Sources 탭의 system 블록디자인을 오른쪽 클릭하여 Create HDL Wrapper를 선택하며, System 블록의 Wrapper을 생성한다. 프로젝트의 Target Language에 따라 Verilog 또는 VHDL로 생성된다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

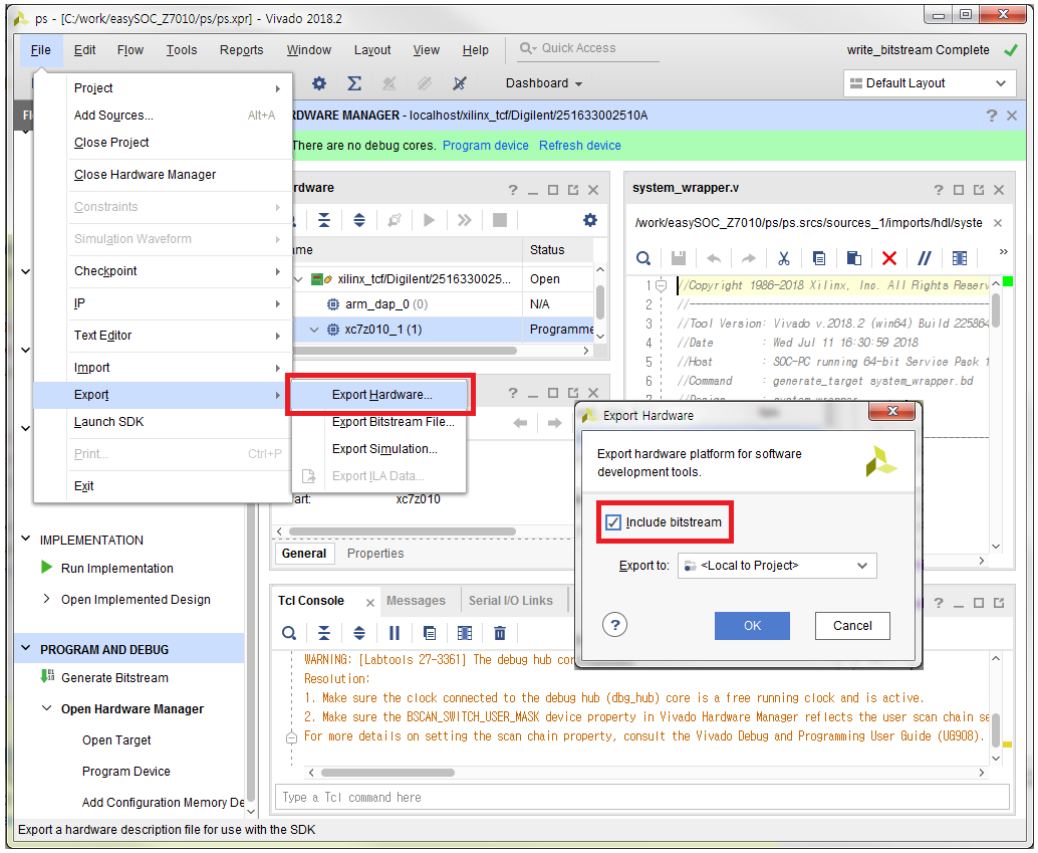
- Processing System을 이용한 실습
  - Synthesis/Implementation/Program
    - ✓ 하드웨어 구성이 끝났으므로 Flow Navigator의 Program and Debug → Generate Bitstream을 클릭해 Zynq 칩에 다운로드 할 bitstream 파일을 생성한다. 이 과정에서 Synthesis, Implementation 또한 모두 실행된다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

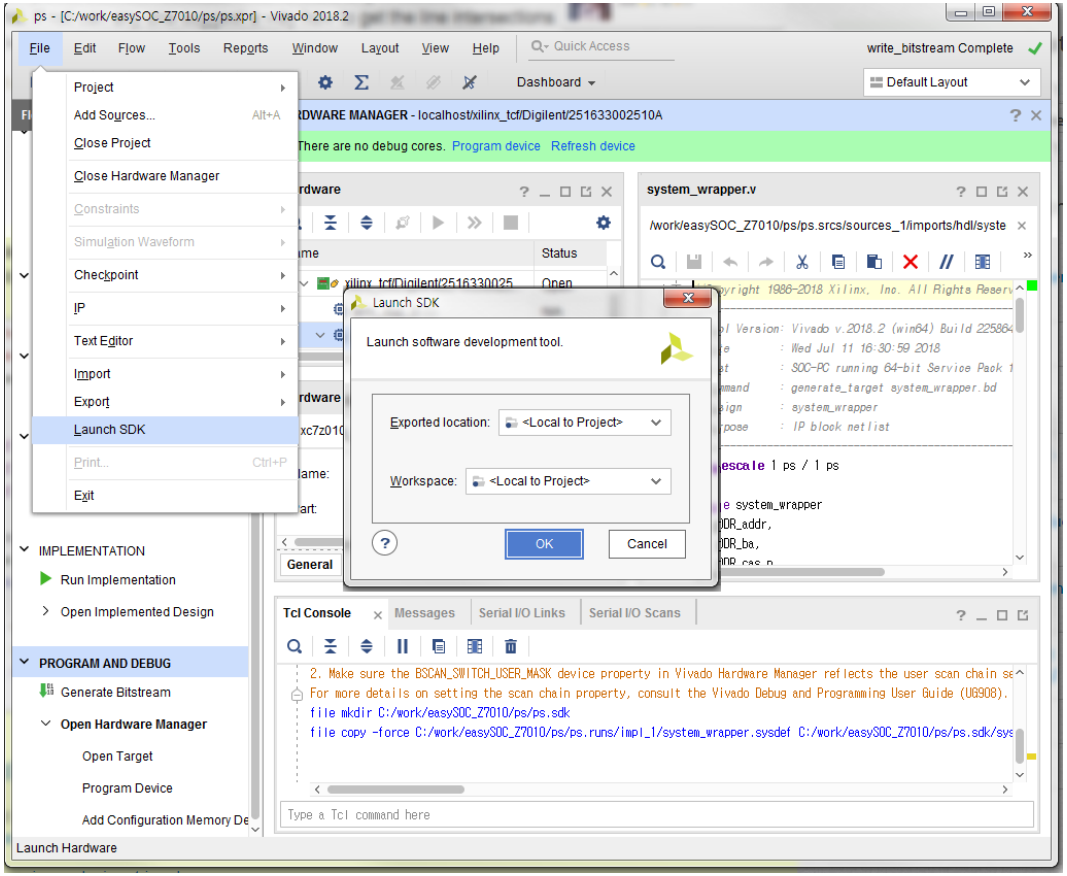
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ Vivado 창에서 File → Export → Export Hardware를 선택한다.
    - ✓ Include bitstream 옵션을 선택 한 후 경로를 지정하여 Export한다.



HUINS

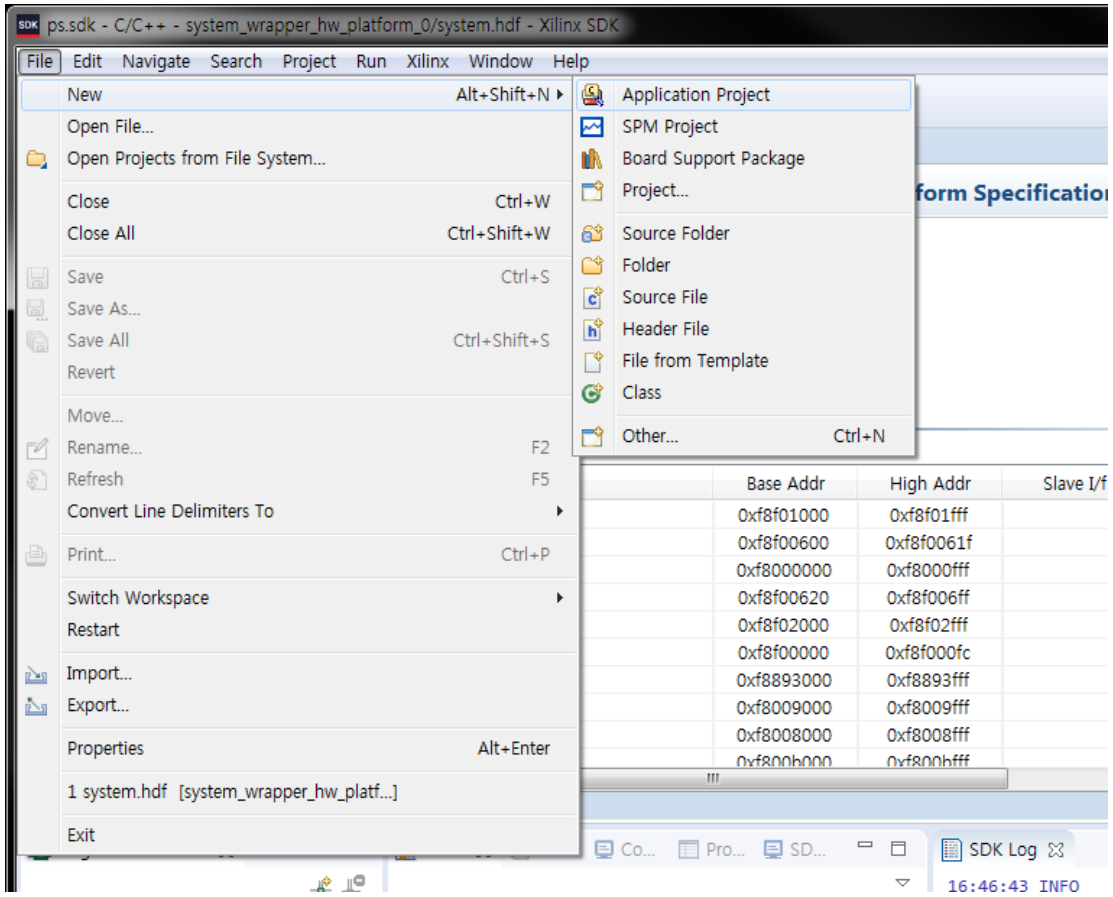
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ Vivado 창에서 File => Launch SDK 메뉴를 선택한다.
    - ✓ H/W Export 경로와 생성할 SDK 프로젝트 경로를 지정하여 실행한다.



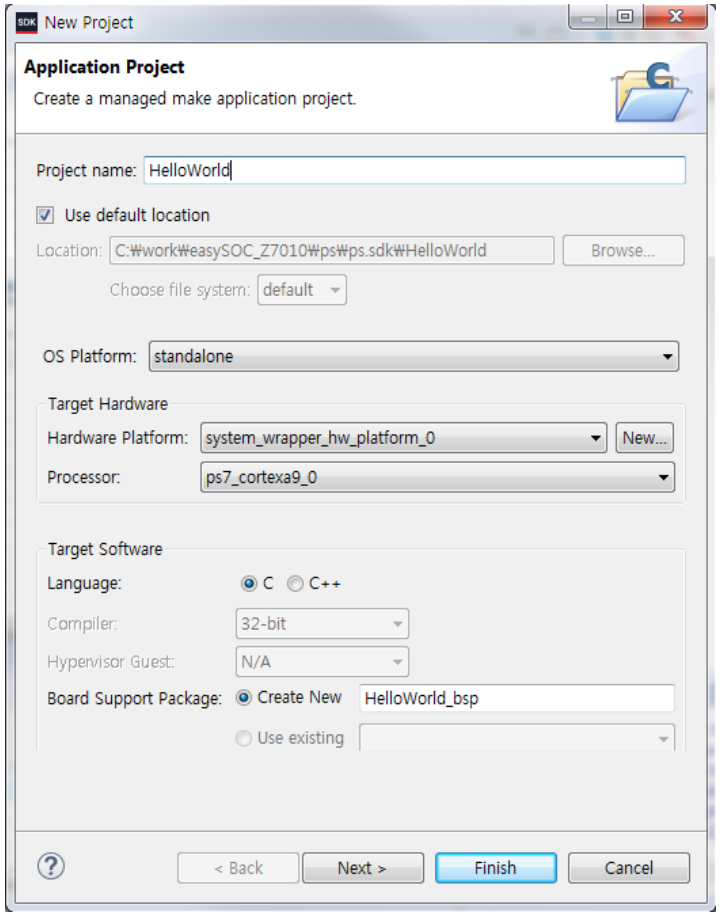
Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ 새로운 F/W 코드 생성을 위해 File → New → Application Project로 진입한다.



Processing System을 포함한 프로젝트 생성 및 동작시키기

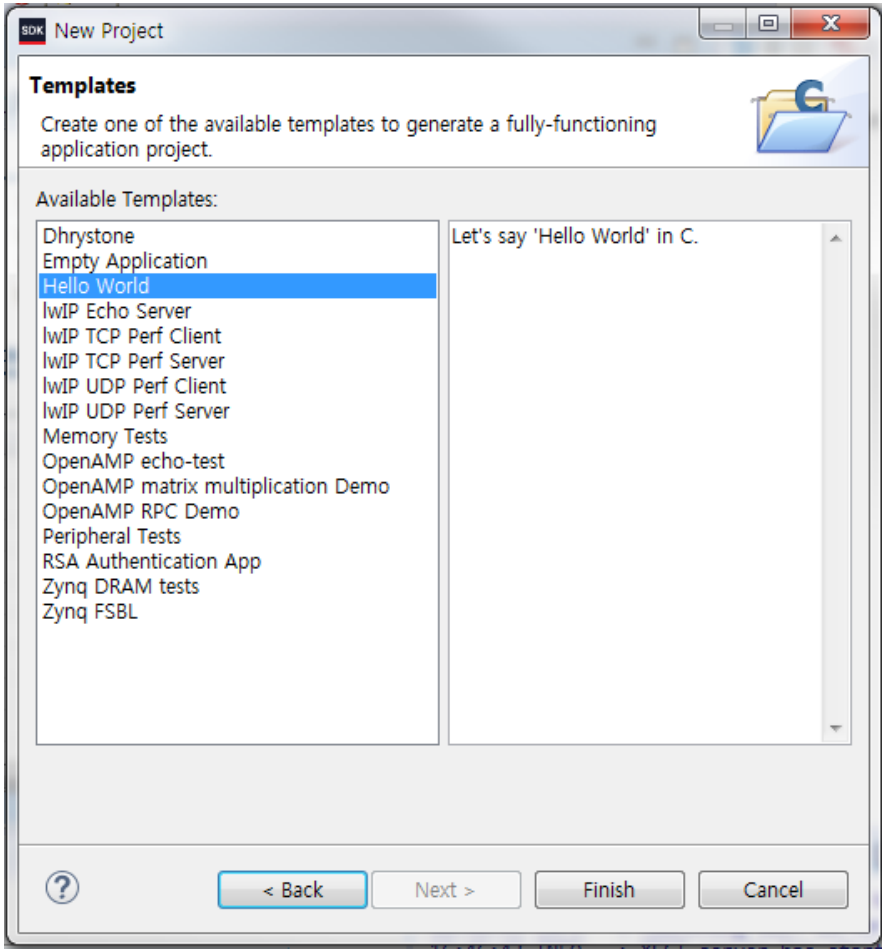
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ New Project 대화상자에서 프로젝트 이름을 설정한 뒤 Next 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ Available Template에 있는 Hello World 항목을 선택한 뒤 Finish 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

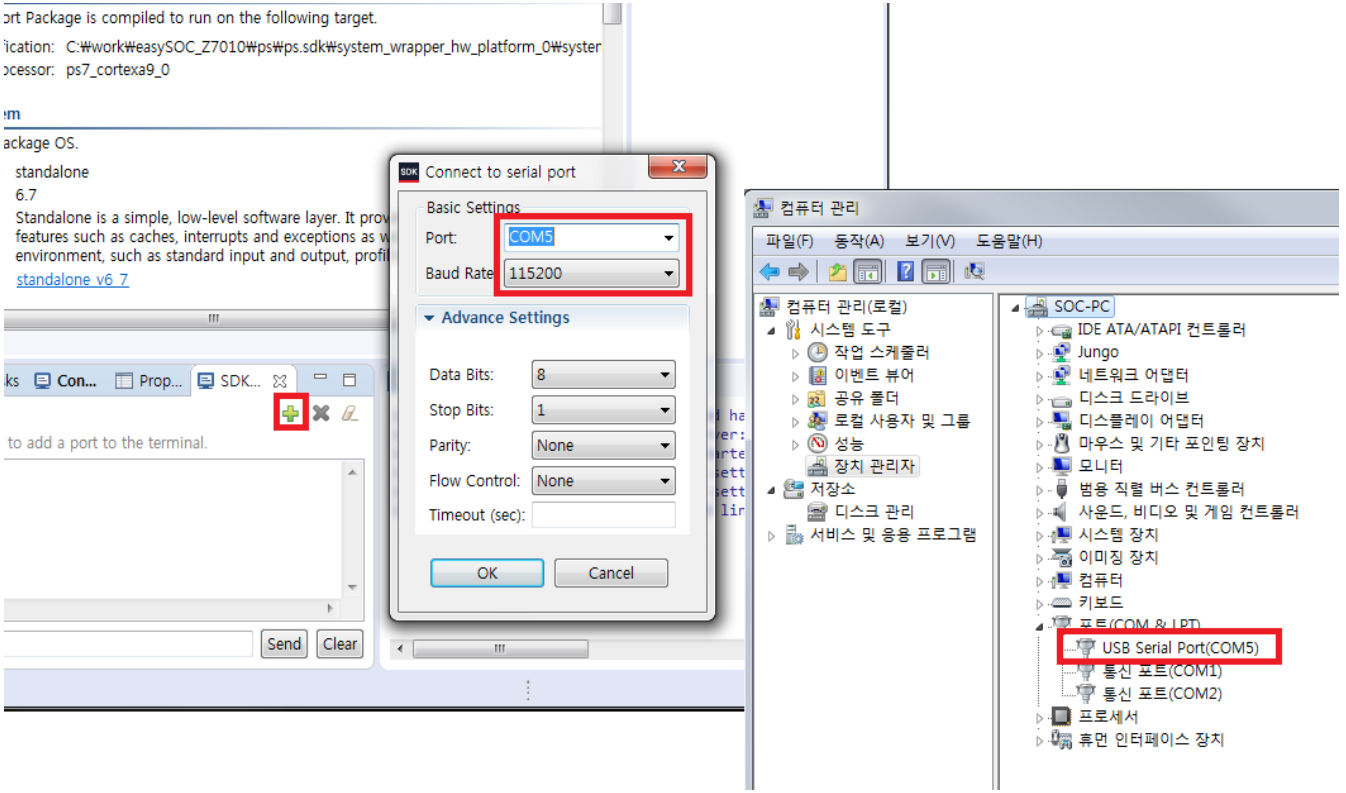
- Processing System을 이용한 실습
  - helloworld.c
    - ✓ UART Serial로 Hello World를 출력하는 애플리케이션이다.
    - ✓ Processing System를 설정할 때 UART1 MIO 8,9 핀을 사용하며, Baud Rate가 115200으로 설정되어 있어 설정에 맞춰 출력하게 된다.

```c
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"


int main()
{
    init_platform();

    print("Hello World\n\r");

    cleanup_platform();
    return 0;
}
```

HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

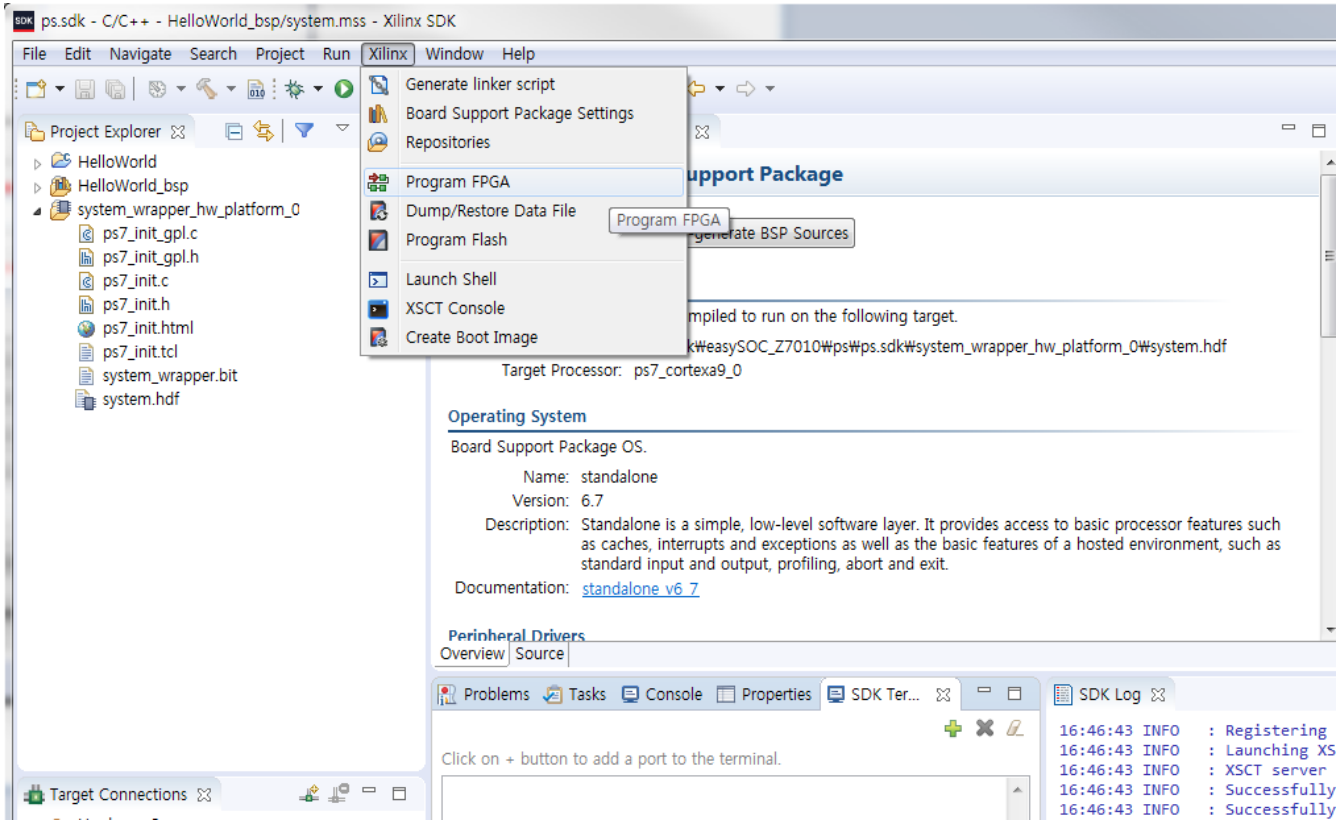
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ SDK 창 하단에 SDK Terminal을 선택하여 서브 창을 연 후, 연결 버튼을 눌러 "Terminal Setting" 대화 상자를 띄운다.
    - ✓ 장치관리자에서 확인한 USB Serial Port에 맞게 포트를 설정해 연결한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

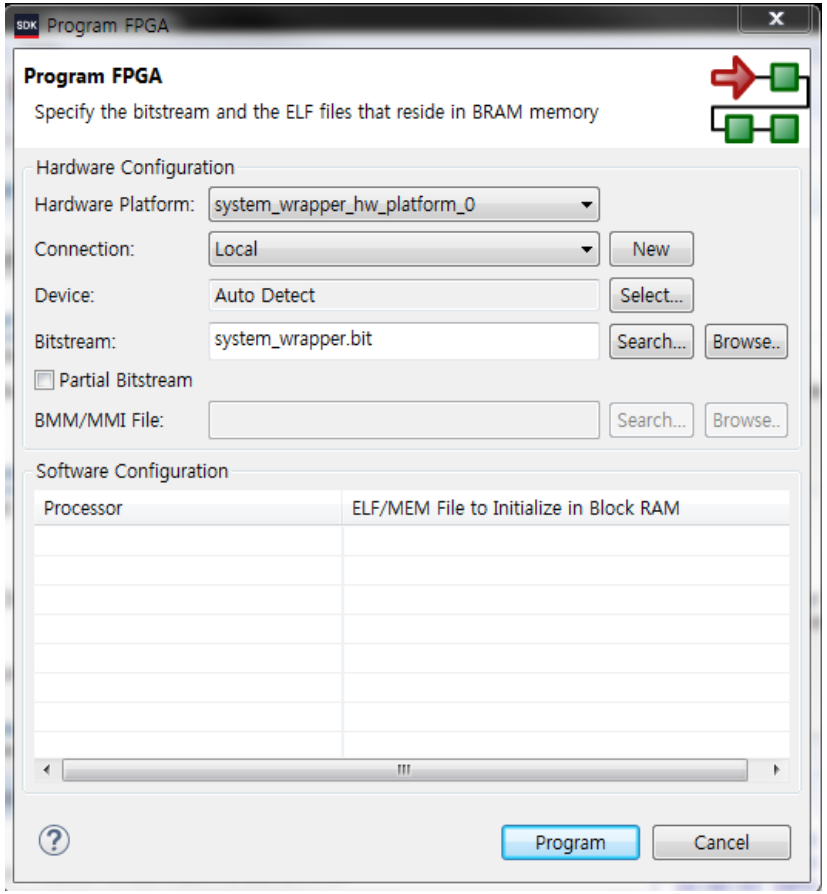
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ SDK 창에서 Xilinx → Program FPGA를 선택한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

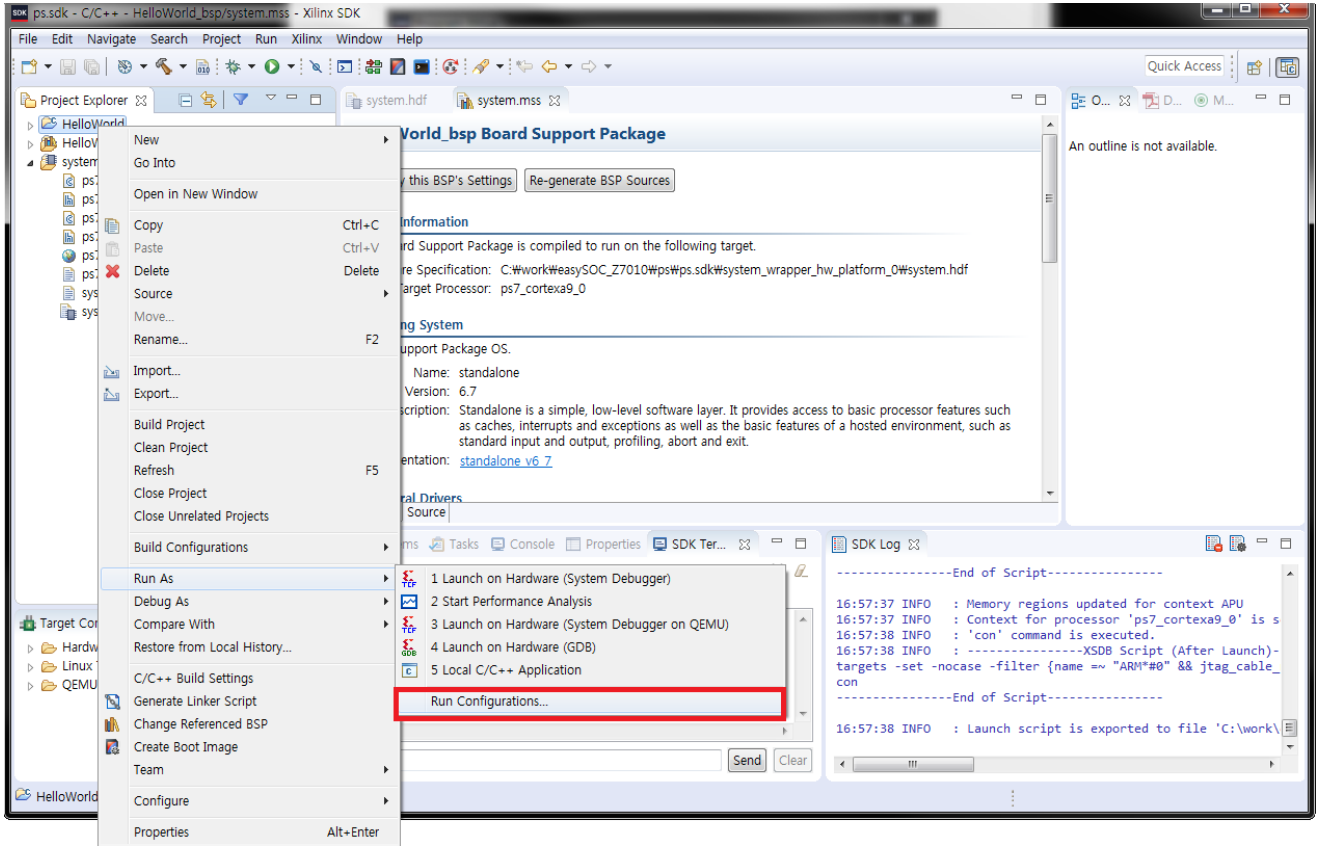
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ Program 버튼을 클릭하여 Zynq 칩에 bitstream 파일을 다운로드 한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

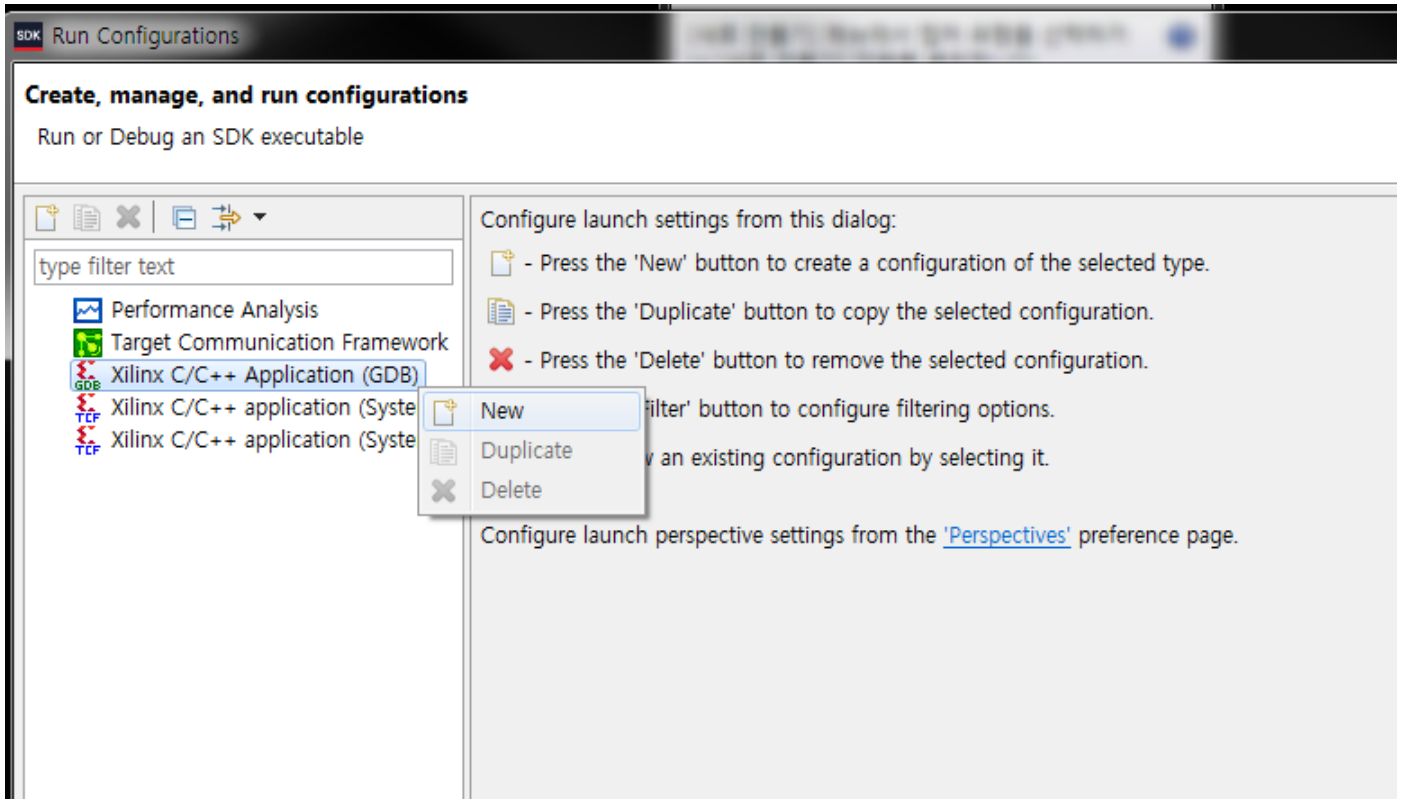
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ 앞서 생성한 HelloWorld 프로젝트를 오른쪽 클릭 한 후 Run As → Run Configurations를 선택한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

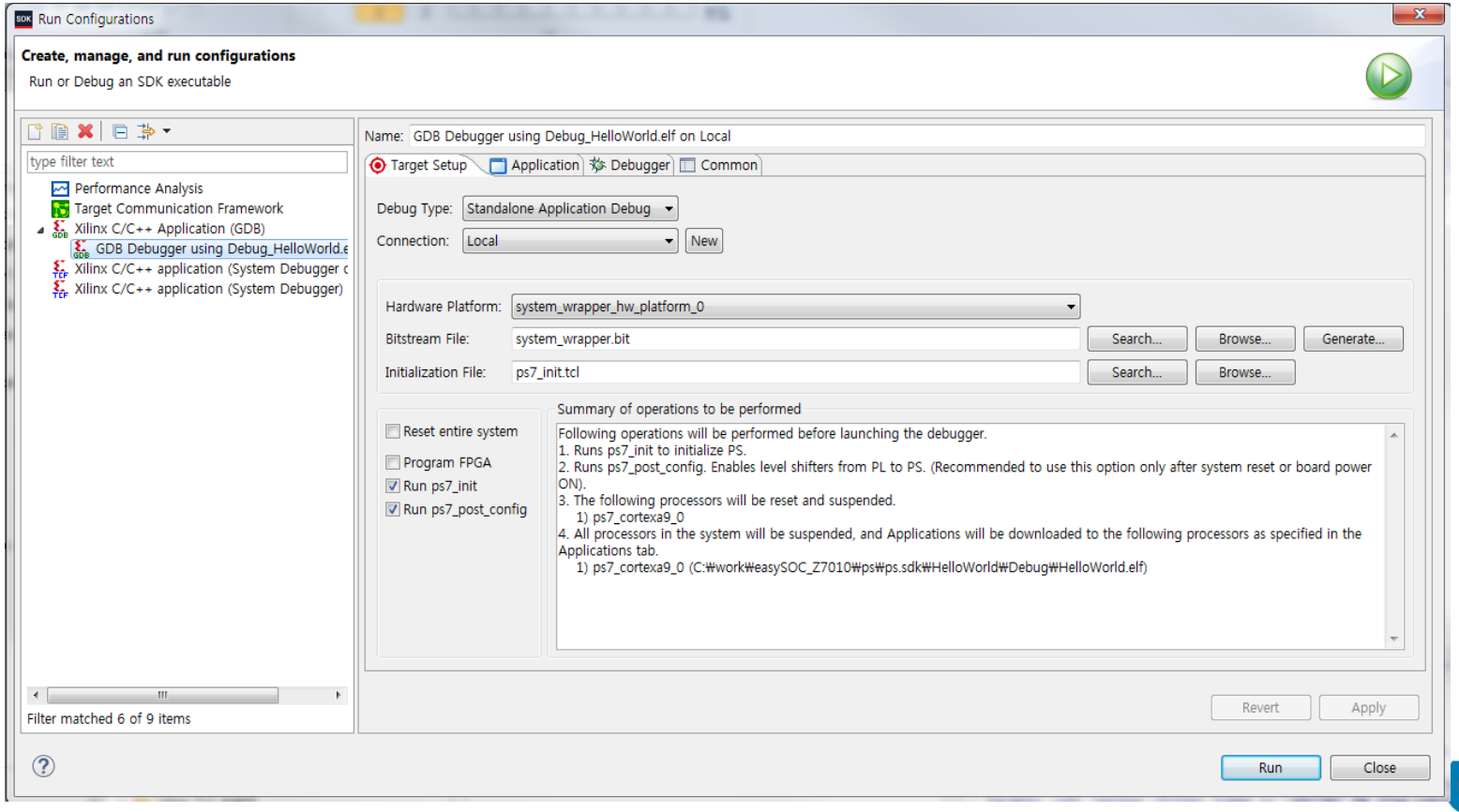
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ Run Configurations 대화상자에서 Xilinx C/C++ application(GDB)를 우 클릭 한 후 New를 선택한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

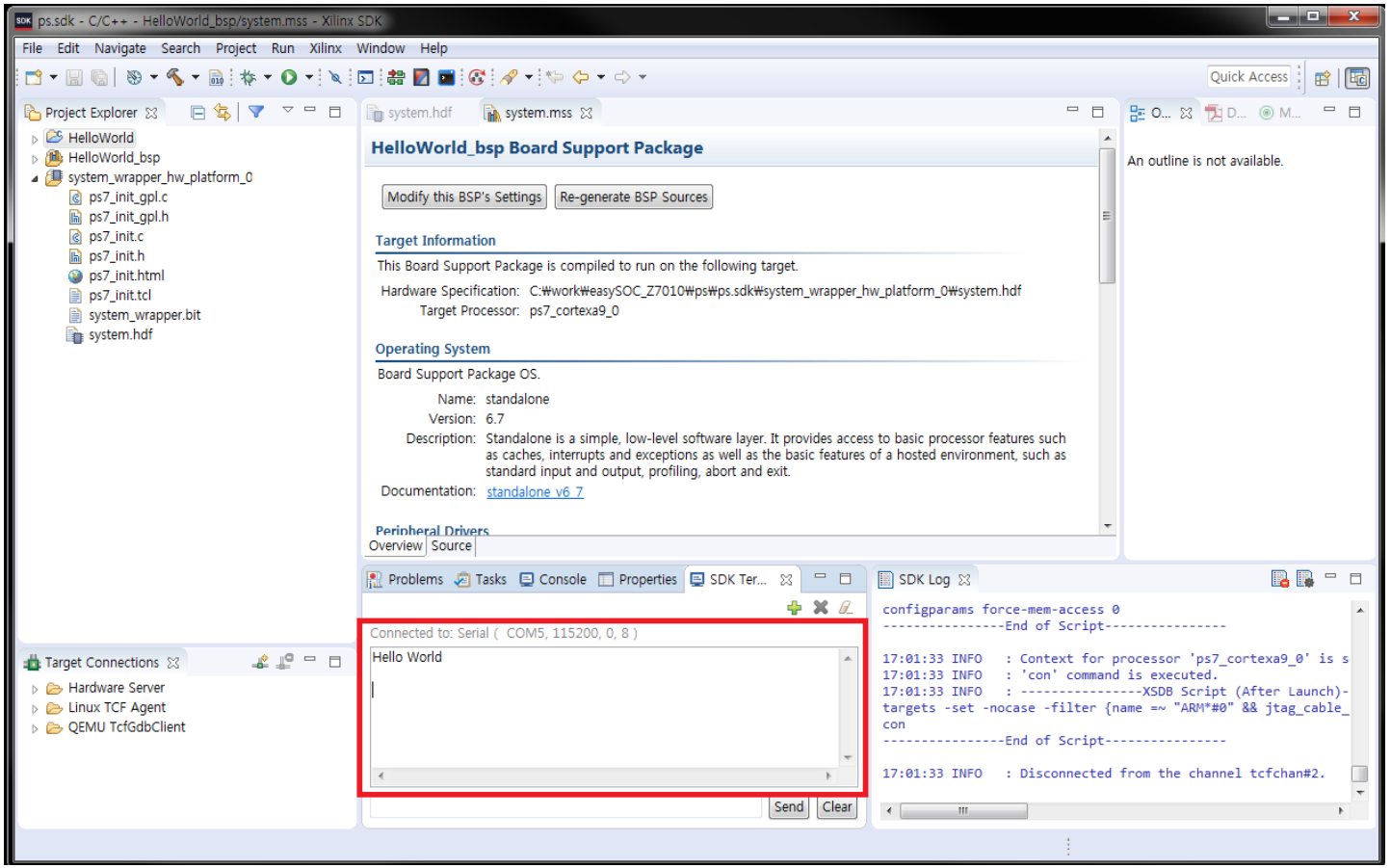
- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ HelloWorld Debug 항목이 선택되어 있는 상태에서 Run 버튼을 클릭한다.



HUINS

Processing System을 포함한 프로젝트 생성 및 동작시키기

- Processing System을 이용한 실습
  - Software 개발 및 실행
    - ✓ 하단의 SDK Terminal tab에서 Hello World 출력을 확인한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

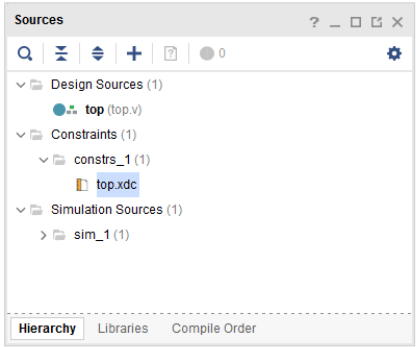
- Programmable Logic 소개
  - 사용자가 제작한 RTL 설계와 Xilinx사가 제공하는 Soft IP의 경우 Programmable Logic으로 다운로드 되어 동작한다.

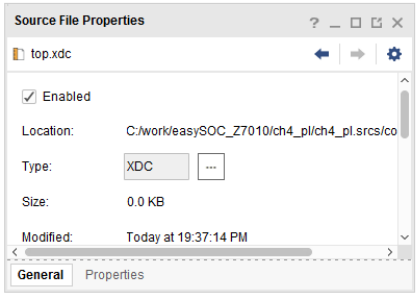
		Low-End Portfolio			Mid-Range Devices			
	Device Name	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™ with CoreSight™						
	Processor Extensions	NEON™ & Single / Double Precision Floating Point for each processor						
	Maximum Frequency	866MHz			Up to 1GHz[1]			
	L1 Cache	32KB Instruction, 32KB Data per processor						
	L2 Cache	512KB						
	On-Chip Memory	256KB						
	External Memory Support[2]	DDR3, DDR3L, DDR2, LPDDR2						
	External Static Memory Support[2]	2x Quad-SPI, NAND, NOR						
	DMA Channels	8 (4 dedicated to Programmable Logic)						
	Peripherals	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO						
	Peripherals w/ built-in DMA[2]	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO						
	Security[3]	RSA Authentication of First Stage Boot Loader, AES and SHA 256b Decryption and Authentication for Secure Boot						
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)	2x AXI 32b Master, 2x AXI 32b Slave 4x AXI 64b/32b Memory AXI 64b ACP 16 Interrupts						
Programmable Logic	7 Series Programmable Logic Equivalent	Artix®-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Kintex®-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA
	Logic Cells (Approximate ASIC Gates[4])	28K (~430K)	74K (~1.1M)	85K (~1.3M)	125K (~1.9M)	275K (~4.1M)	350K (~5.2M)	444K (~6.6M)
	Look-Up Tables (LUTs)	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Total Block RAM (# 36Kb Blocks)	2.1Mb (60)	3.3Mb (95)	4.9Mb (140)	9.3Mb (265)	17.6Mb (500)	19.1Mb (545)	26.5Mb (755)
	Programmable DSP Slices (18x25 MACCs)	80	160	220	400	900	900	2,020
	Peak DSP Performance (Symmetric FIR)	100 GMACs	200 GMACS	276 GMACs	593 GMACs	1,334 GMACs	1,334 GMACs	2,622 GMACs
	PCI Express® (Root Complex or Endpoint)	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC[2]	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs						
	Security[3]	AES and SHA 256b Decryption and Authentication for Secure Programmable Logic Configuration						

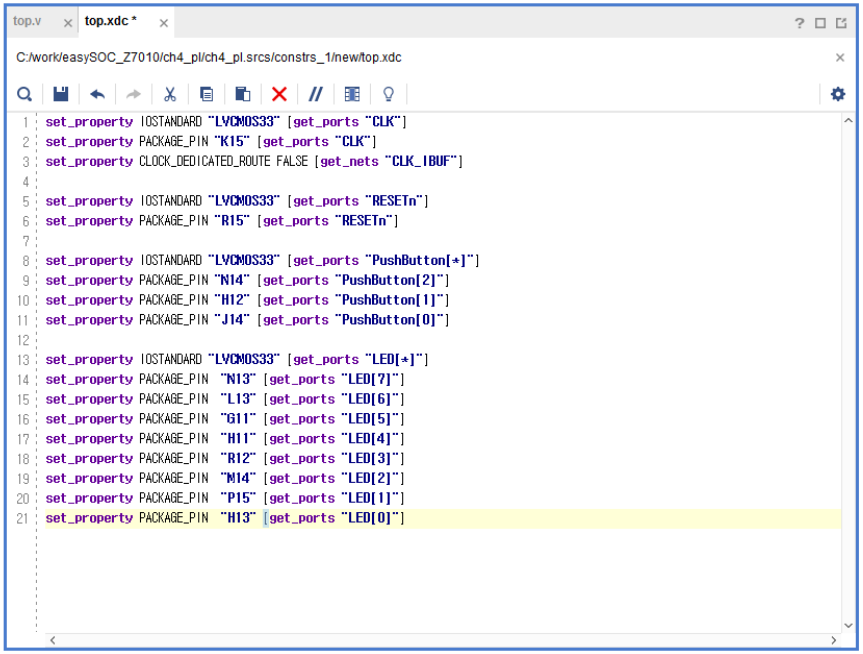
HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Constraint information
  - Vivado tool에서 PL 영역을 사용하기 위해서는 constraint 파일이 필요하며 확장자는 xdc이다.
  - xdc 파일에 timing, physical, configuration에 대한 constrain를 지정할 수 있다.
  - 본 교재에서는 xdc의 physical constraints만 사용한다.
    - ✓ Line 1,5,8,13 : Top RTL 코드의 포트와 연결된 pin type 지정.
    - ✓ 그 외 Line : Top RTL 코드의 포트와 연결된 pin 번호 지정.



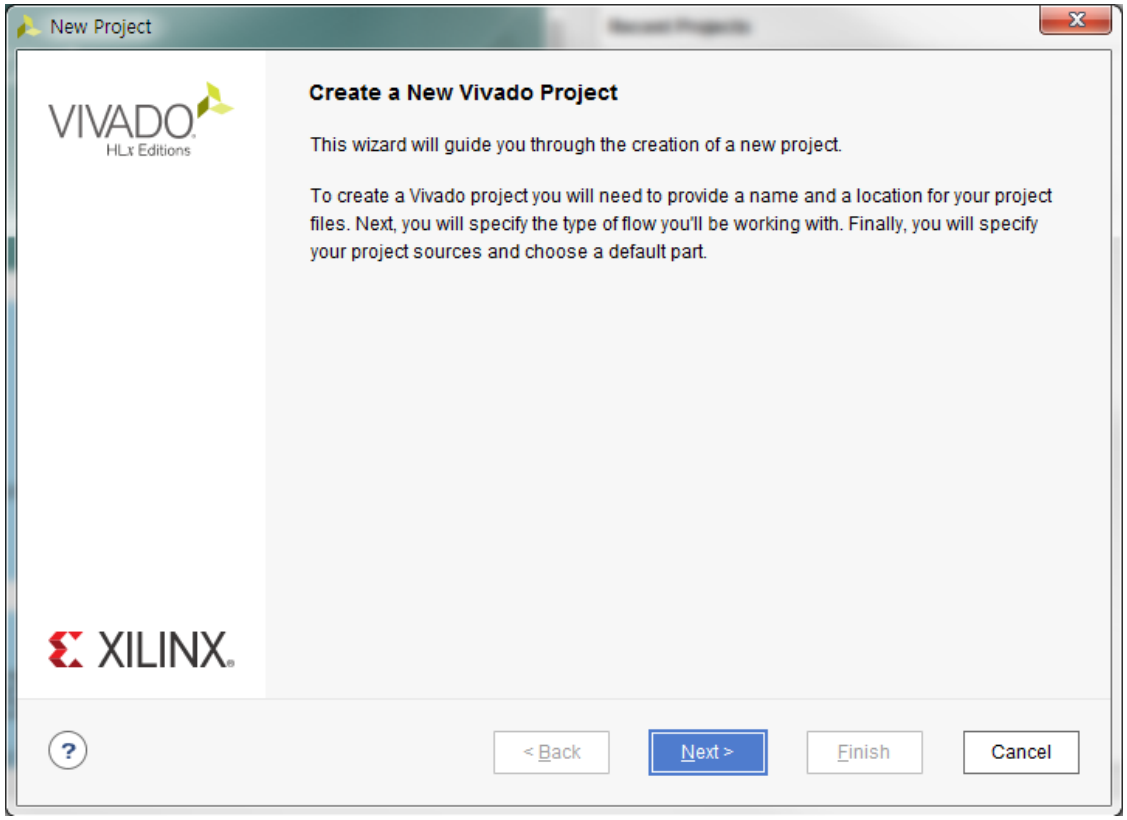




HUINS

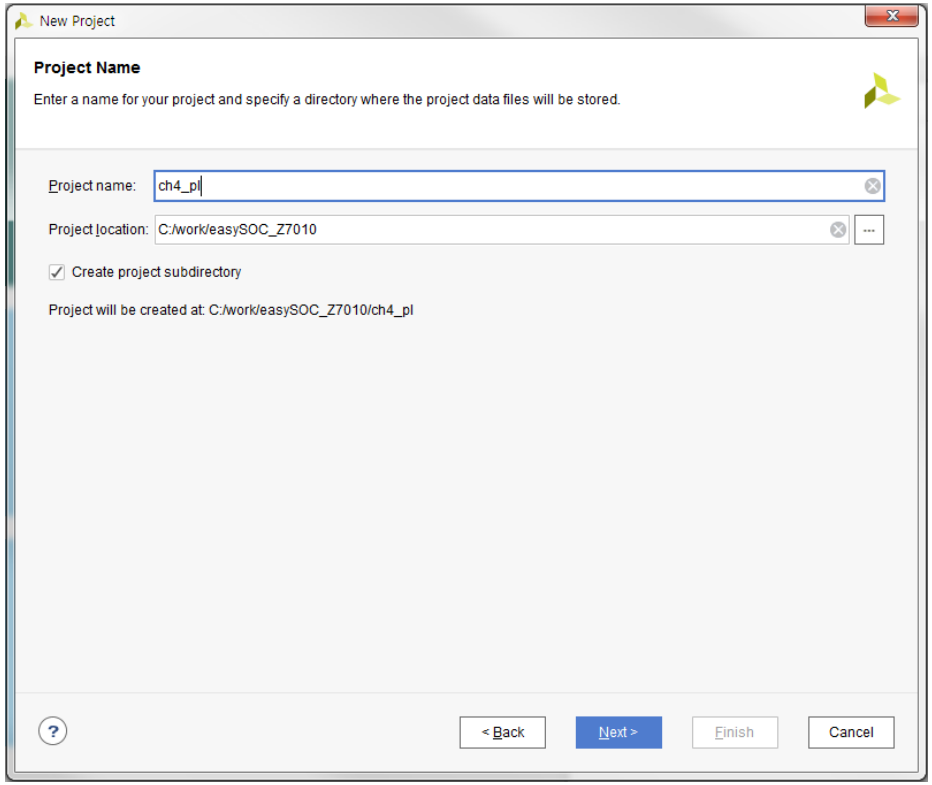
Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ Vivado 실행 후 "Create New Project" 클릭해 새 프로젝트를 생성한다.



Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

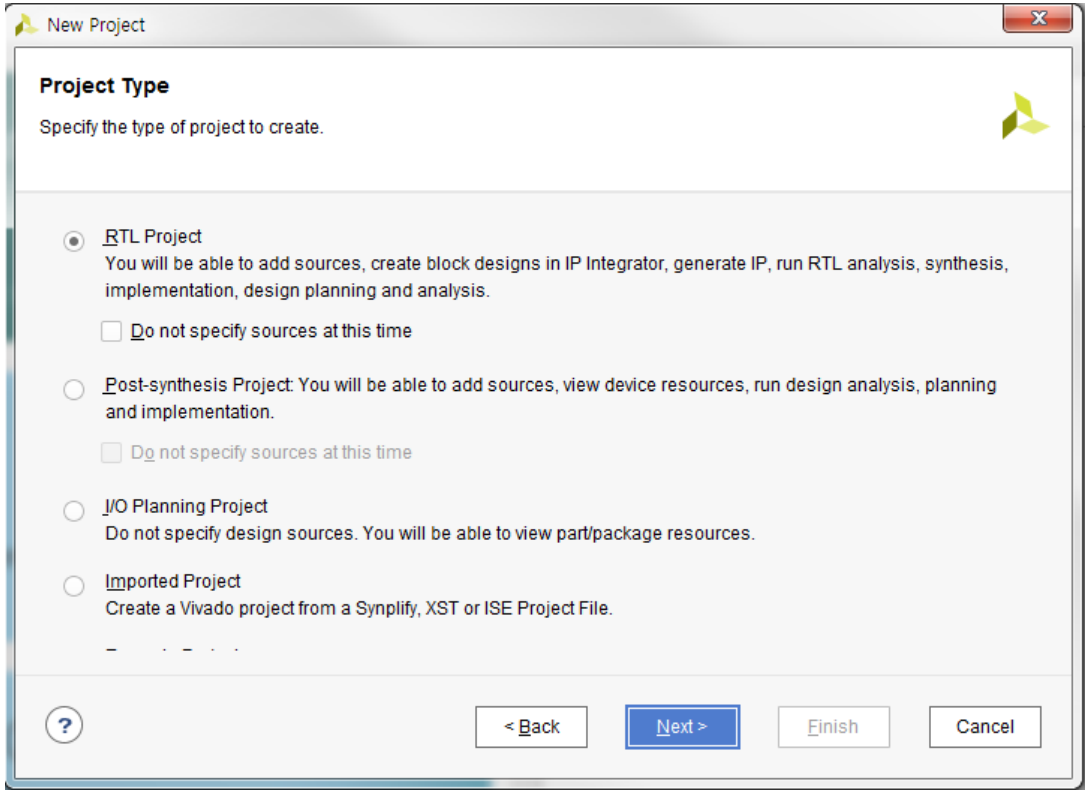
- Programmable Logic 을 이용한 실습
  - Project 생성
    - ✓ Project의 경로와 이름을 설정한다.
    - ✓ "Create project subdirectory"를 선택 할 경우 선택한 경로 아래에 프로젝트 명으로 새 디렉터리가 생성된다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

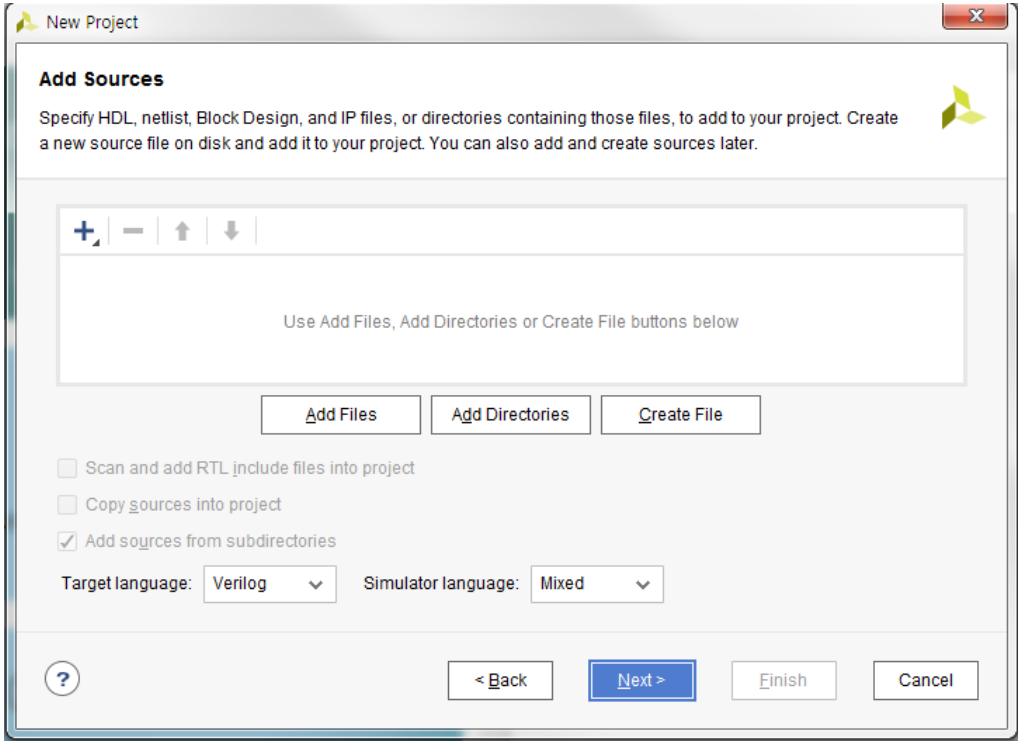
- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ Project Type을 RTL Project로 선택하고 Next 버튼을 클릭한다.
    - ✓ RTL Project에서 HDL Code를 생성하거나 수정한 후 synthesis와 implement를 할 수 있다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

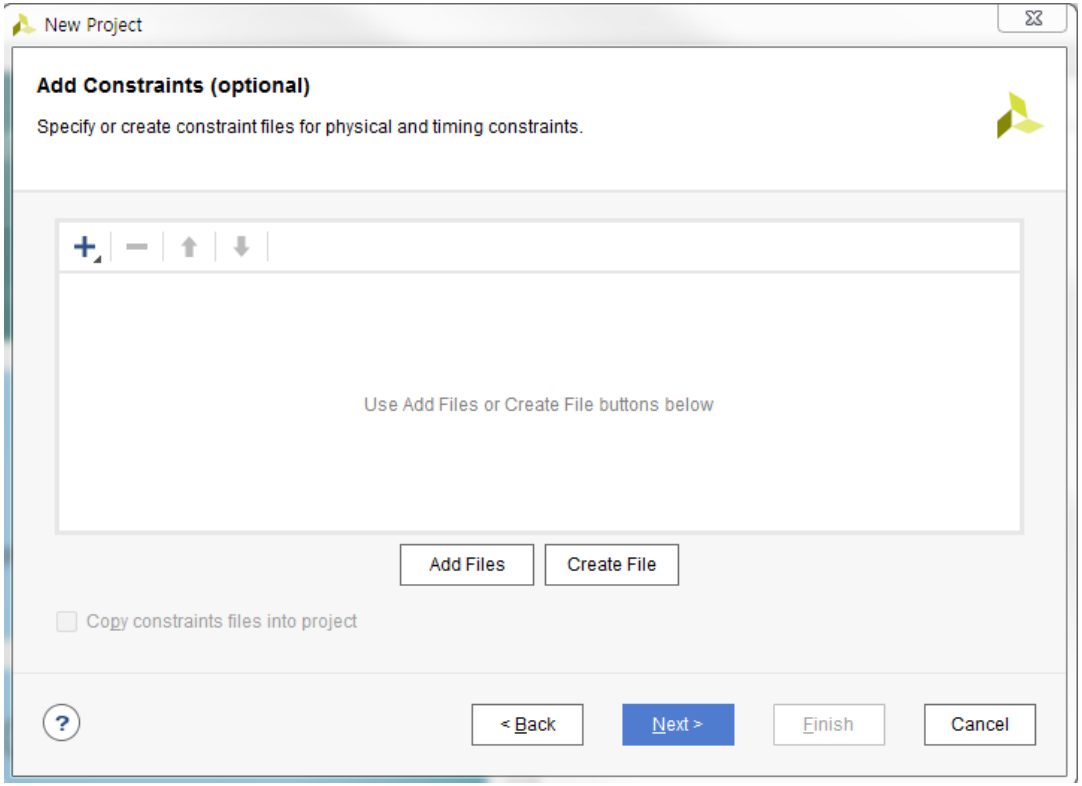
- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ HDL code를 생성하거나 추가하는 단계이며, Target Language 항목을 Verilog로 하고 Next 버튼을 클릭한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

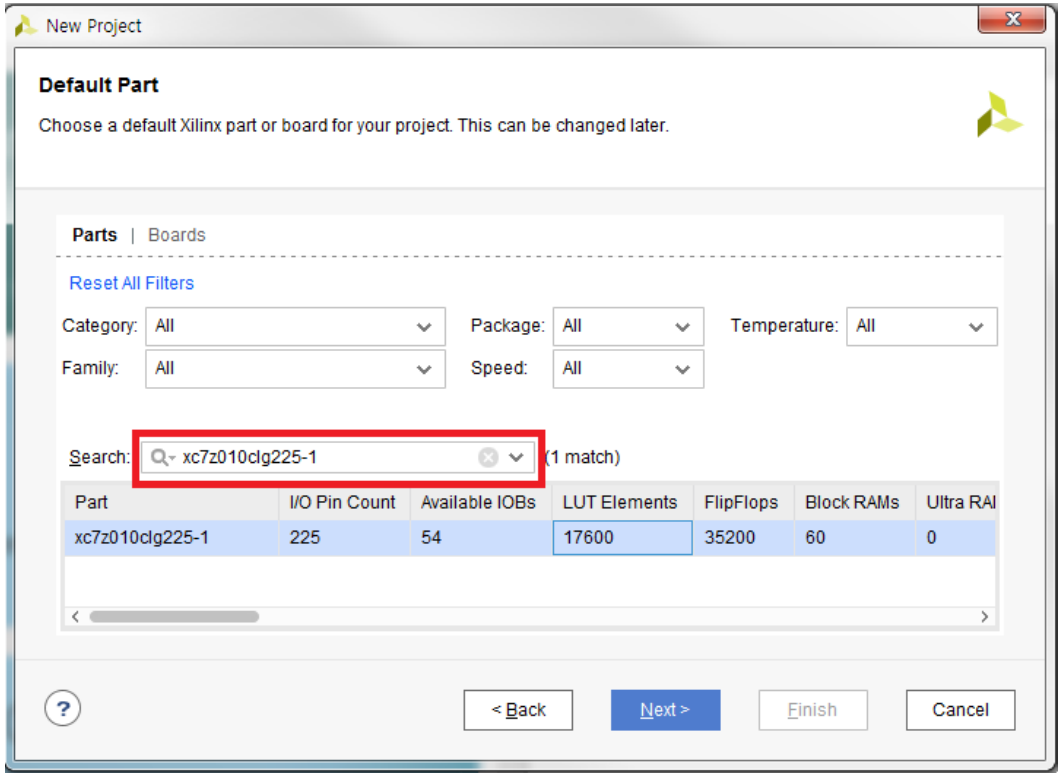
- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ Constraint 파일(.xdc)을 추가하는 단계이며 Next 버튼을 클릭해 진행한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

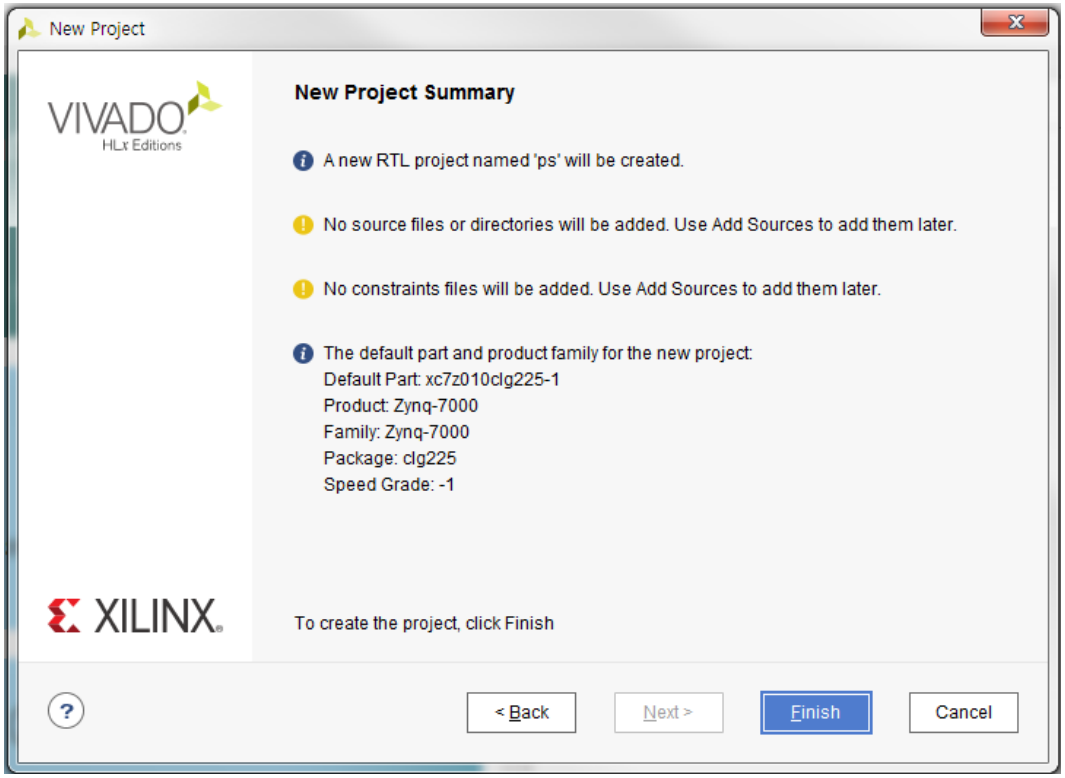
- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ Xilinx FPGA 또는 Zynq를 선택하는 단계. Search 입력란에 xc7z010clg225-1를 입력한 후 동일한 파트를 선택한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

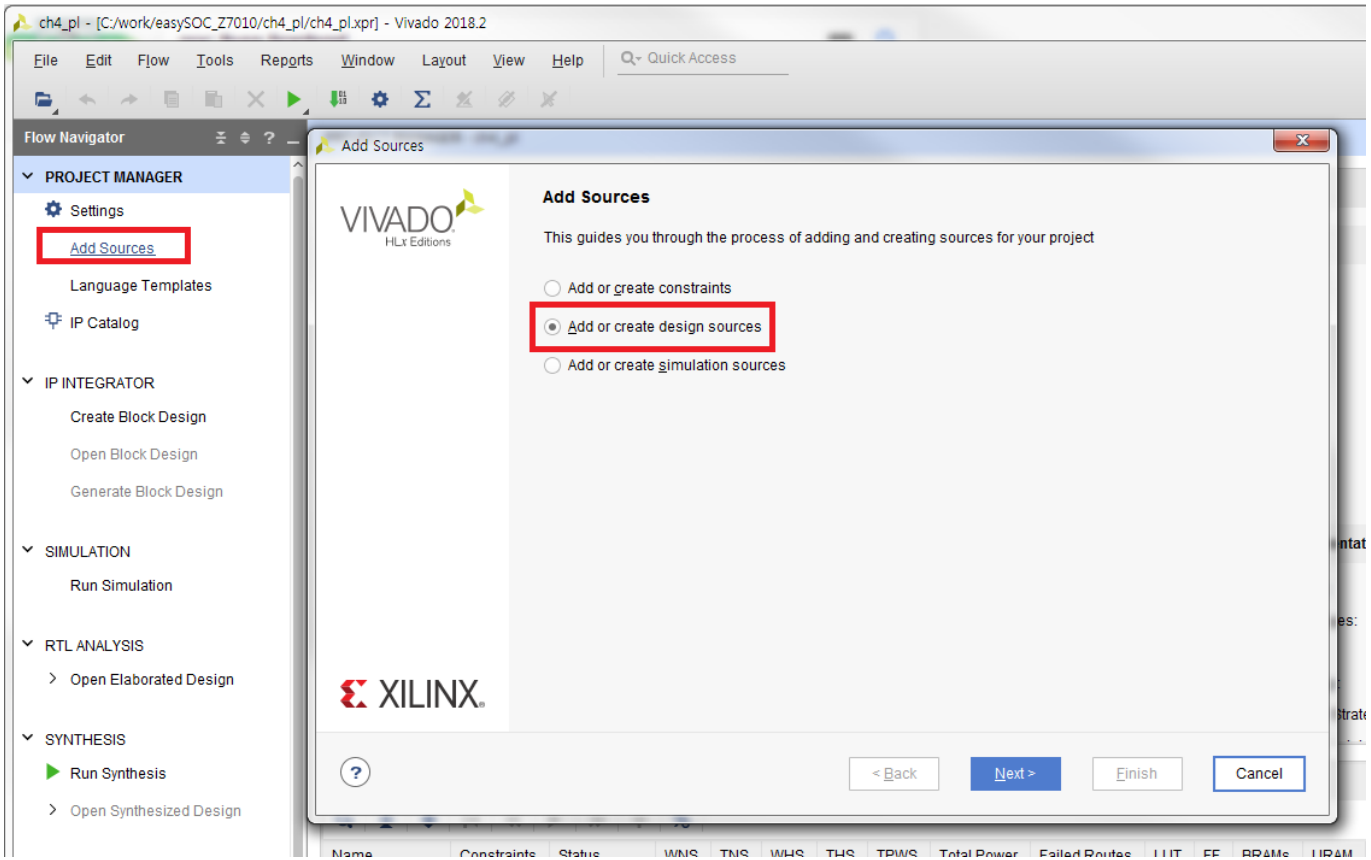
- Programmable Logic을 이용한 실습
  - Project 생성
    - ✓ 설정 한 부분들에 대해서 최종적으로 확인 후 Finish 버튼을 클릭해 프로젝트를 생성한다.



HUINS

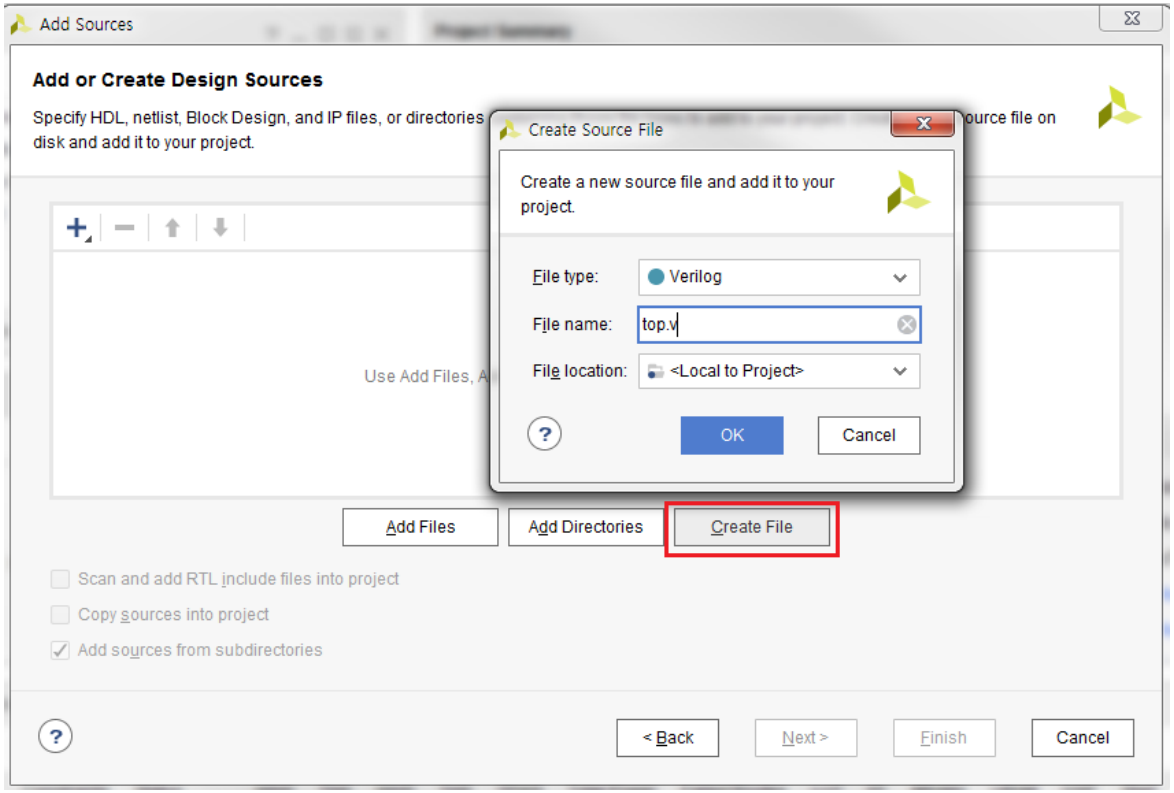
Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - RTL 코드 파일을 새로 생성하기 위해 PROJECT MANAGER에서 차례대로 Add Sources, Add or create design sources 항목을 선택한다.



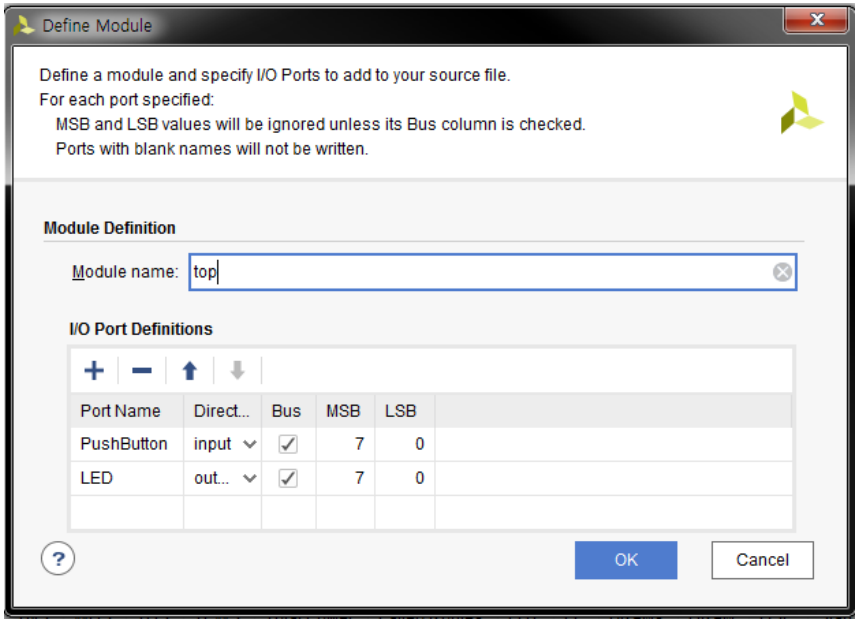
Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - Create File을 클릭해 새로운 HDL 파일을 생성한다.



Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - Define Module에서 입력과 출력 포트를 설정한다.
  - 입력인 PushButton의 Direction을 input으로, 출력인 LED는 output으로 설정한다.
  - PushButton과 LED는 여러 비트이므로 Bus에 항목에 체크하며 각각 8개를 사용하므로 [7:0]으로 선언한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - 생성된 top.v에 다음과 같이 내용을 입력한다.
  - PushButton의 입력이 LED로 출력된다.

Project Summary | top.v

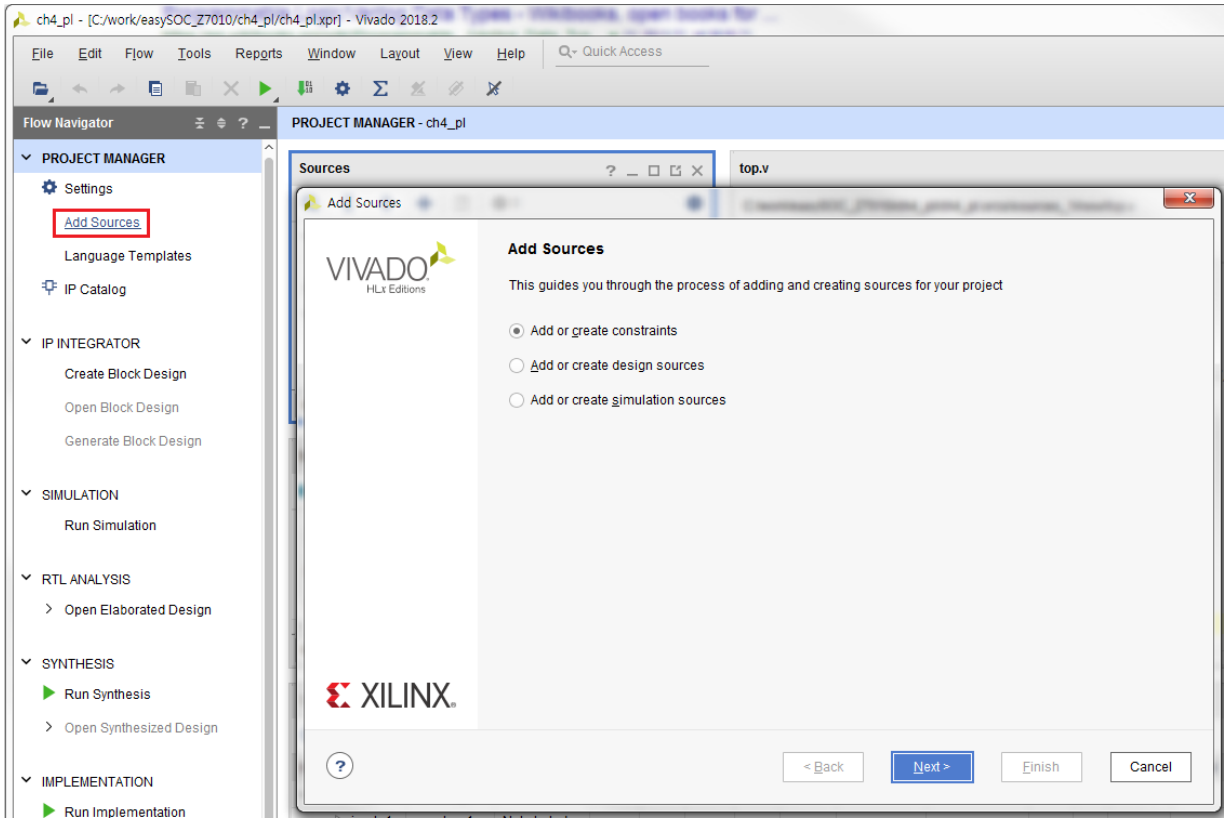
C:/work/2_EasySOC_Z7010/ch4_pl/ch4_pl.srcs/sources_1/new/top.v

```
1   `timescale 1ns / 1ps
2
3   module top(
4       input [7:0] PushButton,
5       output [7:0] LED
6   );
7
8   assign LED = PushButton;
9
10  endmodule
11
```

HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

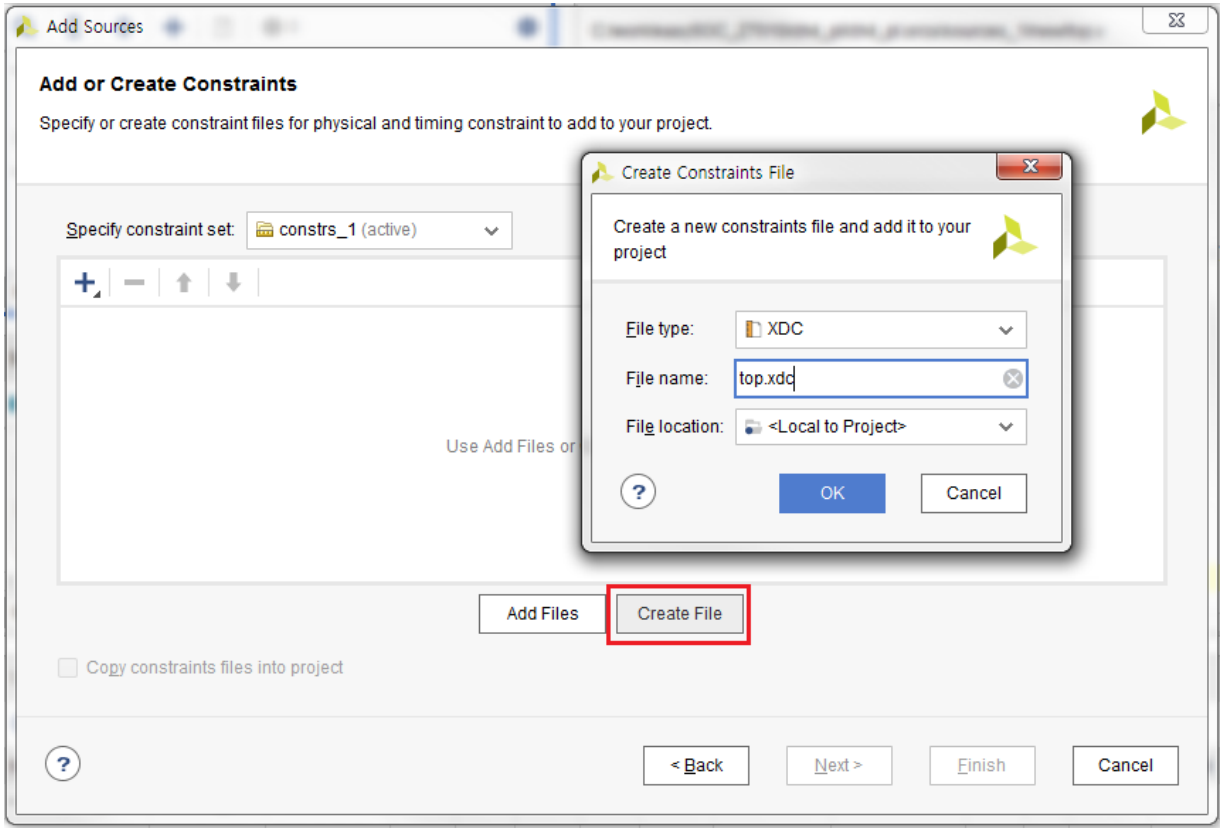
- Programmable Logic(PL)을 이용한 실습
  - constraints 파일을 추가하기 위해, PROJECT MANAGER 에서 Add Sources, Add or create constraints를 차례대로 선택한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - Create File을 클릭한 뒤 XDC파일의 이름을 설정한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - 생성된 top.xdc에 다음과 같이 내용을 입력한다.

Project Summary | top.v | top.xdc

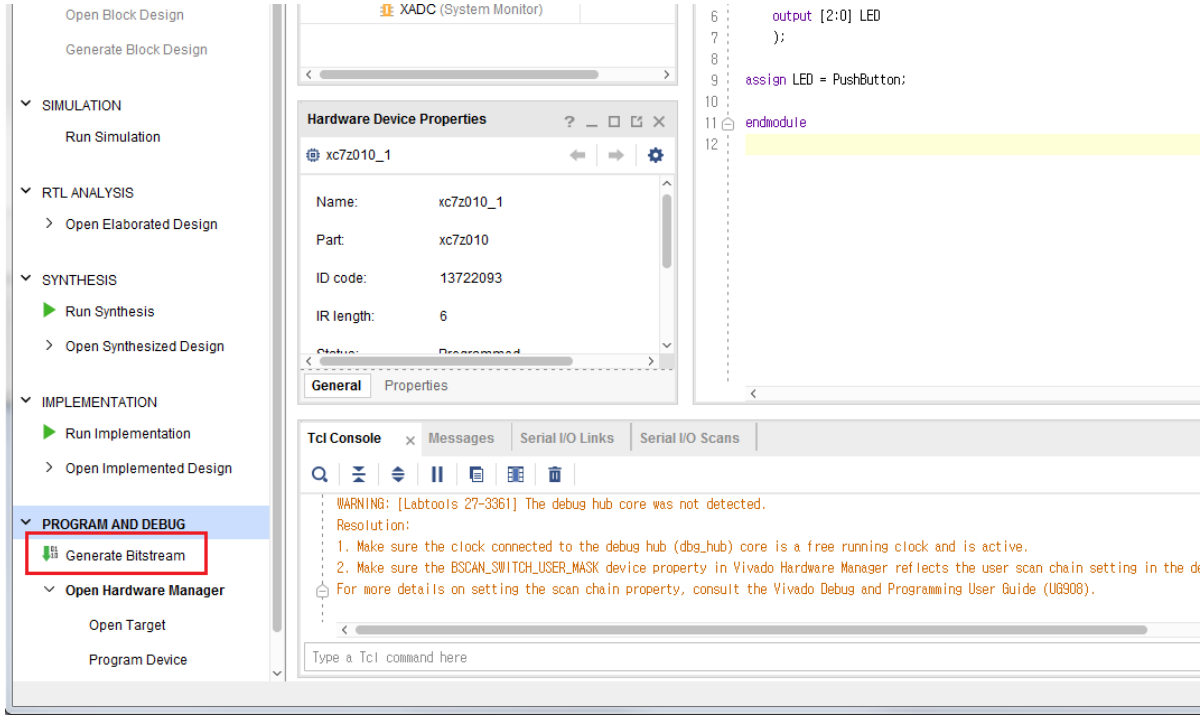
C:/work/2_EasySOC_Z7010/ch4_pl/ch4_pl.srcs/constrs_1/new/top.xdc

```
1  set_property IOSTANDARD "LVCMOS33" [get_ports "PushButton[*]"]
2  set_property PACKAGE_PIN "H14" [get_ports "PushButton[7]"]
3  set_property PACKAGE_PIN "J13" [get_ports "PushButton[6]"]
4  set_property PACKAGE_PIN "L15" [get_ports "PushButton[5]"]
5  set_property PACKAGE_PIN "L14" [get_ports "PushButton[4]"]
6  set_property PACKAGE_PIN "R15" [get_ports "PushButton[3]"]
7  set_property PACKAGE_PIN "N14" [get_ports "PushButton[2]"]
8  set_property PACKAGE_PIN "H12" [get_ports "PushButton[1]"]
9  set_property PACKAGE_PIN "J14" [get_ports "PushButton[0]"]
10
11 set_property IOSTANDARD "LVCMOS33" [get_ports "LED[*]"]
12 set_property PACKAGE_PIN  "N13" [get_ports "LED[7]"]
13 set_property PACKAGE_PIN  "L13" [get_ports "LED[6]"]
14 set_property PACKAGE_PIN  "G11" [get_ports "LED[5]"]
15 set_property PACKAGE_PIN  "H11" [get_ports "LED[4]"]
16 set_property PACKAGE_PIN  "R12" [get_ports "LED[3]"]
17 set_property PACKAGE_PIN  "M14" [get_ports "LED[2]"]
18 set_property PACKAGE_PIN  "P15" [get_ports "LED[1]"]
19 set_property PACKAGE_PIN  "H13" [get_ports "LED[0]"]
```

HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

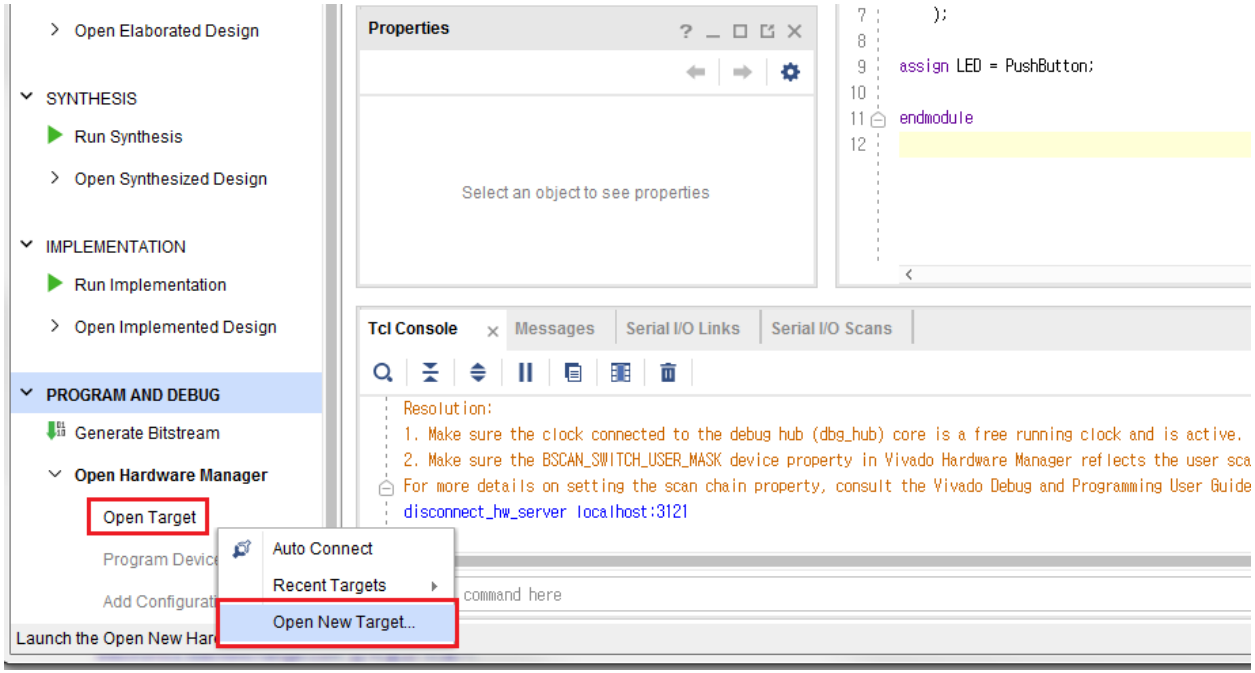
- Programmable Logic(PL)을 이용한 실습
  - Bitstream 파일을 생성하기 위해 Generate Bitstream 항목을 클릭함.
  - Generate Bitstream 항목을 클릭하면 자동으로 synthesis, implementation, generate bitstream 작업이 순차적으로 진행된다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

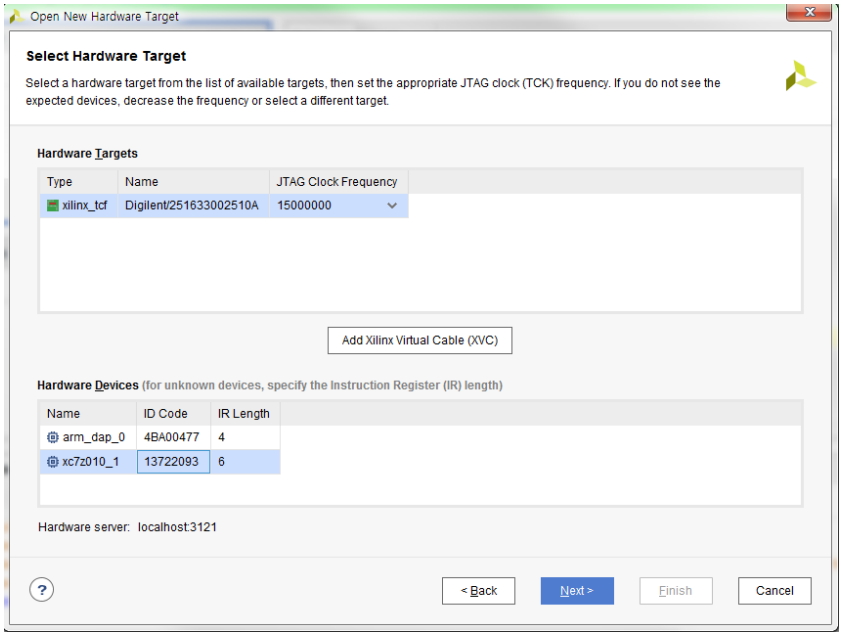
- Programmable Logic(PL)을 이용한 실습
  - 보드와 host 컴퓨터를 USB cable로 연결한다.
  - Vivado 툴에서 Open Hardware Manager => Open Target 항목을 클릭한 후 Open New Target을 선택한다.

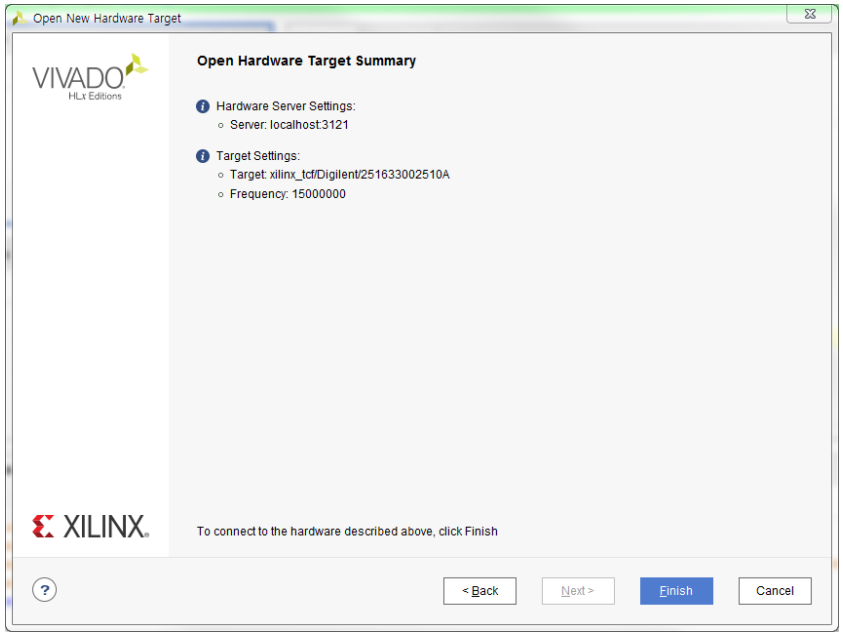


HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - Hardware Devices에 XC7Z010_1을 선택하고 next 버튼을 클릭한다.
  - ZYNQ-7000 PL에 Bitstream 파일을 다운로드 하기 위한 설정 내용을 알려주는 대화상자에서 Finish버튼을 클릭해 연결을 완료한다.

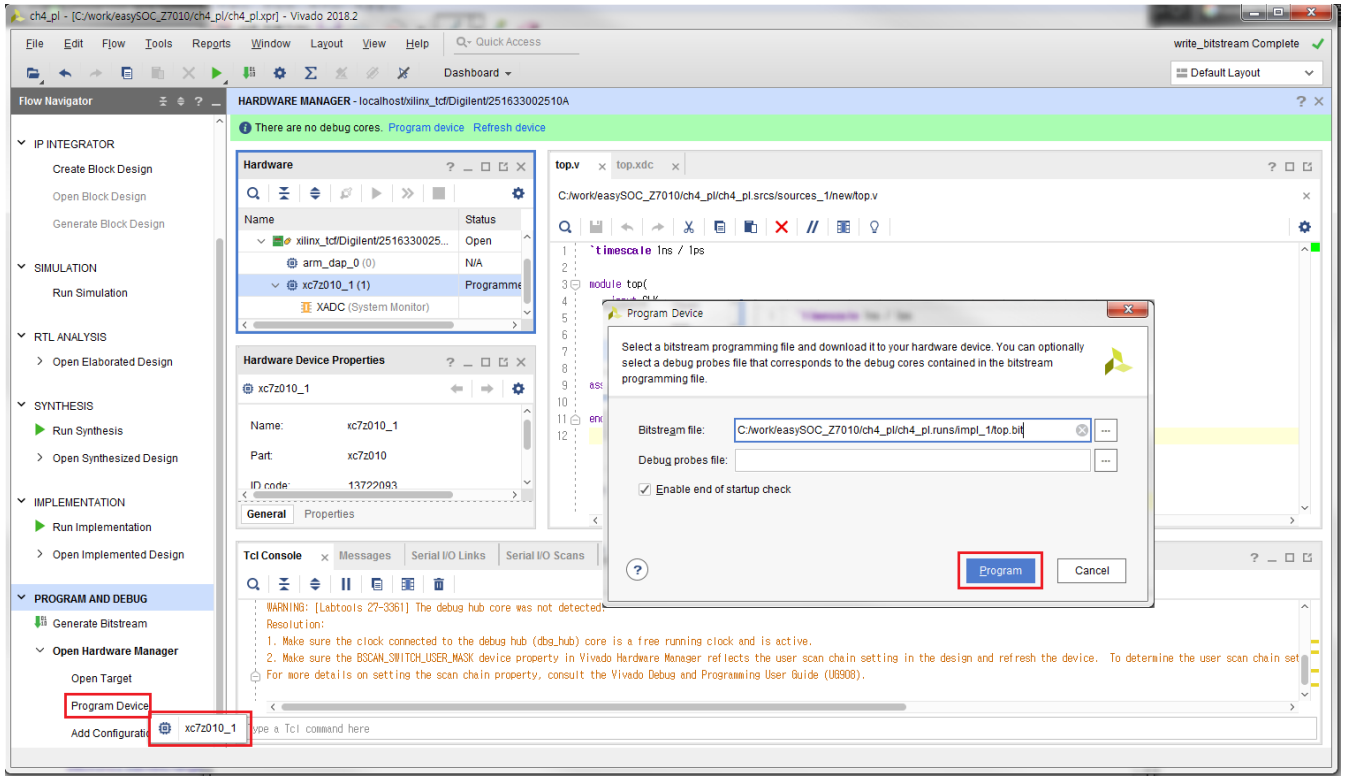




HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

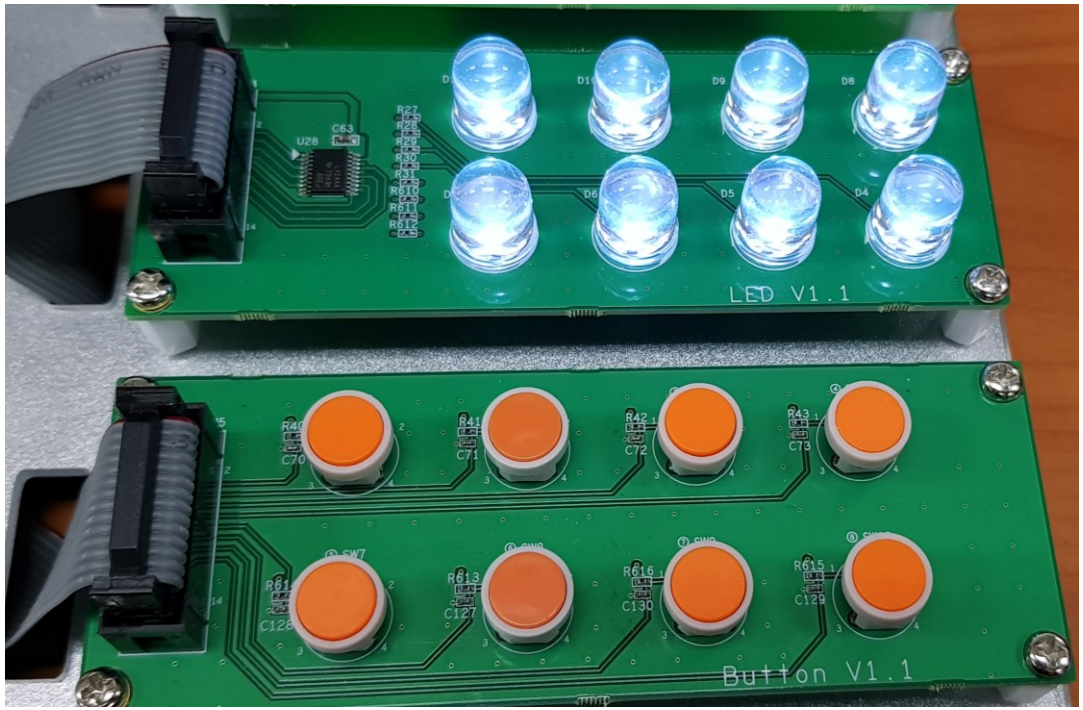
- Programmable Logic(PL)을 이용한 실습
  - Hardware Manager => Program Device 선택 후 xc7z10_1 항목을 선택한다.
  - Program 버튼을 클릭하여 Bitstream 파일을 선택하여 다운로드 한다.



HUINS

Programmable Logic을 포함한 프로젝트 생성 및 동작시키기

- Programmable Logic(PL)을 이용한 실습
  - Bitstream 파일 다운로드 완료 후 LED가 점등된다.
  - PushButton은 High로 풀업 되어 있으며 눌렀을 때 '0'이 된다.
  - LED는 PushButton에 연결되어 버튼을 누를 때 꺼짐을 확인할 수 있다.



HUINS

감사합니다.

HUINS