

Ultra96 Training Kit User Guide

Hardware : Ultra96 Training Kit

Software : Xilinx SDSoC 2018.3

UG Version : 2019



목차

Chapter 1 소개 및 툴 설치	4
1-1 Zynq UltraScale+ MPSoC 소개	5
1-2 Xilinx 및 SDSoC 툴 소개	7
1-3 이니프로 및 Ultra96 Training Kit 소개	9
1-4 SDSoC 툴 설치	11
Chapter 2 Vivado 를 사용한 FPGA 설계	20
2-1 Vivado Design Flow	21
2-2 LED 카운터 구현	36
2-3 7-Segment 디지털시계 구현	37
2-4 VGA 출력 구현	38
Chapter 3 SDK 를 사용한 Zynq 설계	39
3-1 SDK Development Flow	40
3-2 IIC Interface 구현	41
3-3 SPI Interface 구현	42
3-4 UART Interface 구현	43

3-5 Custom IP 생성	44
3-6 UART 로 Custom IP 컨트롤 구현	45
3-7 SD Card Booting	46

Chapter 1 소개 및 툴 설치

1-1 Zynq UltraScale+ MPSoC 소개

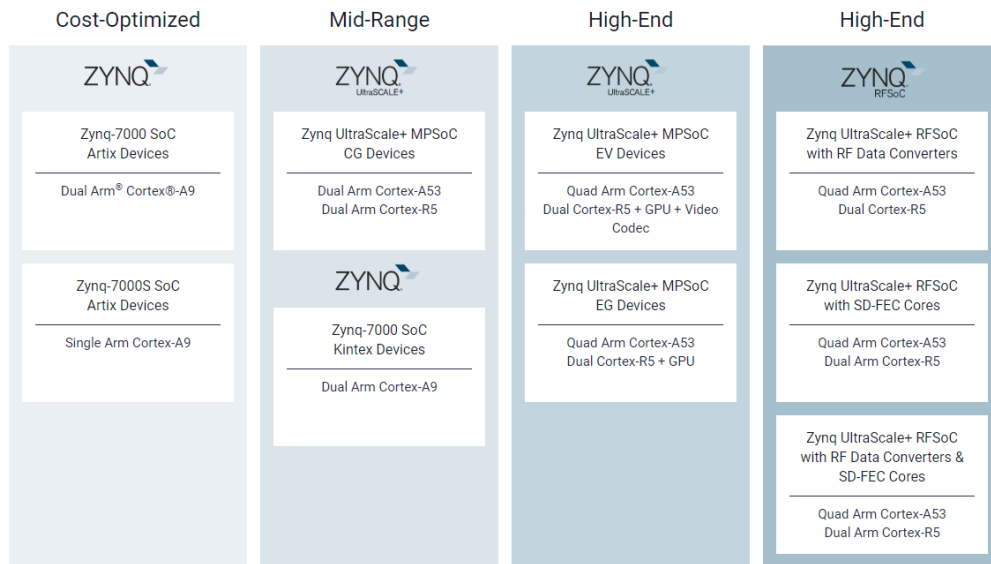
1-2 Xilinx 및 SDSoC 툴 소개

1-3 이니프로 및 Ultra96 Training Kit 소개

1-4 SDSoC 툴 설치

1-1 Zynq UltraScale+ MPSoC 소개

Zynq 는 FPGA 와 Processing System 을 하나의 IC 로 만든 자일링스의 SoC 타입의 디바이스이다. Zynq 디바이스는 현재 Zynq-7000 SoC, Zynq UltraScale+ MPSoC 및 Zynq UltraScale+ RFSoc 와 같은 3 종류의 디바이스로 판매되고 있다. 기본적인 구조는 FPGA 와 Processing System 을 하나의 IC 에 결합한 형태이지만 각각의 디바이스는 그 목적에 맞는 구성을 가지고 있다.



[출처] 자일링스 홈페이지(<http://www.xilinx.com>)

그림 1-1 Xilinx Zynq Devices

Zynq-7000 SoC 는 저가형 디바이스 단가가 민감한 임베디드 시스템에 적절한 디바이스이고 Zynq UltraScale+ MPSoC 는 고해상도의 영상 또는 고속의 신호를 처리하기에 적합한 디바이스이고 Zynq UltraScale+ RFSoc 는 수기가 샘플링이 가능한 DAC 및 ADC 가 있어서 5G 와 같은 고속 통신 및 고속 DAQ 장비에 사용되는 디바이스이다.

본 교재에서는 3 종류의 Zynq 디바이스 중 Zynq UltraScale+ MPSoC 가 탑재된 Ultra96 보드를 사용하여 실습을 통해 Zynq UltraScale+ MPSoC 을 어떻게 디자인하는지에 대한 실질적인 내용을 다룬다.

Zynq UltraScale+ MPSoC 는 CG, EG, EV 디바이스의 3 종류의 디바이스가 있다. 3 종류의 디바이스는 그림 1-1 과 같이 각각의 특징을 가지고 있다. CG 디바이스는 Dual Arm Cortex-A53 과 Dual Arm Cortex-R5 를 가지고 있고 EG 디바이스는 Quad Arm Cortex-A53 과 Dual Arm Cortex-R5 를 가지고 있고 EV 디바이스는 EG 디바이스의 Quad Arm Cortex-A53, Dual Arm Cortex-R5 와 함께 GPU 및 Video Codec 이 포함되어 있다.

Ultra96 보드에 탑재된 디바이스는 3 종류의 디바이스 중 EG 디바이스로 Quad Arm Cortex-A53 과 Dual Arm Cortex-R5 를 가지고 있는 디바이스이다. 그리고 Zynq UltraScale+ MPSoC EG Devices 는 Quad Arm Cortex-A53 과 Dual Arm Cortex-R5 가 포함된 Processing System 과 FPGA 형태인 Programmable Logic 이 결합된 형태의 디바이스이다. 여기서 Processing System 은 줄여서 PS 라 부르고 FPGA 부분은 PL(Programmable Logic)이라고 부른다.

PS 와 PL 이 결합된 형태인 Zynq 디바이스가 출시되어 지금과 같이 큰 인기를 끌게 된 이유는 일반적인 SoC 와 같은 형태인 PS 에서 실행되는 소프트웨어를 좀 더 빨리 처리할 수 있는 하드웨어 가속기를 PL 에 구현하여 전체 시스템의 성능을 높일 수 있기 때문이다. 그리고 시스템에 필요한 다양한 하드웨어를 사용자가 원하는 대로 PL 에 구현이 가능하다는 이점 때문이다.

PL 에 구현된 하드웨어 가속기 또는 특정 목적의 하드웨어를 SoC 설계 단계에서 고려하여 SoC 내에 포함한 SoC 를 만들면 좀 더 좋은 성능을 낼 수 있지만 고정되어 있는 하드웨어 기능을 가지고 있는 SoC 디바이스는 특정 목적으로만 사용 가능한 디바이스가 된다. 특정 목적으로 한정되면 수요가 많이 줄어들게 되어 SoC 개발 및 생산에 대한 수지타산이 맞지 않는 경우가 많이 발생하여 이런 SoC 는 많지 않다. 그래서 이런 분야에서는 자일링스의 SoC 디바이스가 많이 쓰이고 있다. 하지만 최근 수요가 급격하게 많아지고 있는 인공지능 분야에서는 딥러닝 알고리즘을 가속할 수 있는 특정 목적의 디바이스들이 나오고 있다. 그래서 이 분야에서는 자일링스 SoC 및 FPGA 를 포함한 여러 가지 디바이스들의 경쟁이 격화되고 있는 상태이다.

1-2 Xilinx 및 SDSoC 툴 소개

자일링스는 FPGA 업계에서 점유율 1 위로 FPGA 전체 시장의 50%정도를 점유하고 있다. 기존에는 FPGA 로 대표되는 Programmable 디바이스를 설계하여 판매하는 Fabless 였다면 지금은 FPGA 뿐만 아니라 FPGA 에 ARM 프로세서를 포함한 여러 가지 하드 매크로 IP 코어가 결합된 형태의 SoC 를 설계하여 판매하는 회사로 확장되고 있다.

최근 인공지능을 위한 딥러닝의 등장으로 많은 연산량을 빠르게 처리할 수 있는 가속기의 수요가 많이 늘어났고 이를 충족하는 NVIDIA 의 GPGPU 를 많이 사용하게 되었다. 그런데 FPGA 를 사용하면 GPGPU 보다 성능대비 전력소비가 적어서 최근 FPGA 를 사용한 가속기에 대한 수요가 늘어나고 있는 추세이다. 이런 시장 상황에 맞추어 자일링스는 Alveo 라는 그래픽 카드와 비슷한 형태의 PCI Express 카드를 출시하였다. Alveo 는 자일링스의 최신 FPGA 가 탑재되며 SD Accel 이라는 툴을 사용하여 가속기를 구현하여 사용할 수 있다.

Alveo 는 서버 또는 데스크탑과 함께 많은 연산량을 빠르게 처리할 수 있는 가속기로 사용된다면 임베디드 시스템에서는 Zynq 디바이스의 PL 을 가속기로 사용하여 구현하는 솔루션을 가지고 있다.

Zynq 디바이스를 디자인하기 위해서는 여러 가지 툴이 필요하다. Zynq 의 PL 에 하드웨어를 구현하기 위해 Vivado 또는 Vivado HLS 를 사용할 수 있고 PS 에서 Running 되는 소프트웨어를 구현하기 위해 Xilinx SDK 툴을 사용할 수 있으며 PS 에 리눅스를 포팅하기 위해 Petalinux 툴을 사용할 수 있다. 그리고 PL 에 하드웨어와 PS 에 Running 할 소프트웨어를 모두 구현하기 위해 SDSoC 툴을 사용할 수 있다.

Vivado 는 기본적으로 Verilog, VHDL, Schematic 을 사용하여 하드웨어를 설계하는 툴이다. 여기에 자일링스는 IPI(IP Integrator)라는 블록 디자인 형태의 툴을 추가하여 하드웨어를 좀 더 쉽게 설계할 수 있도록 하였다. 하드웨어를 설계하는 또 하나의 툴은 Vivado HLS 이다. Vivado HLS 는 C/C++/OpenCL 로 하드웨어를 설계할 수 있는 툴이다. HDL 보다 추상화레벨을 한단계 높은 형태의 설계 방법이라고 할 수 있다. 여기에 더해 SDSoC 는 C/C++/OpenCL 을 사용하여 Zynq 의 PL(하드웨어)과 PS(소프트웨어)를 동시에 설계할 수 있도록 만든 툴이다. C/C++/OpenCL 을 사용하여 동작에 대한 기능 검증을 한 후 프로파일 기능을 통해 각각의 함수들의 동작 시간들을 파악한다. 이런 함수들 중 동작시간이 오래 걸리는 함수를 설정을 통해 Vivado HLS 툴로 하드웨어로 합성할 수

있다. 그리고 소프트웨어에서 처리한 데이터를 하드웨어에 넘겨주고 하드웨어에서 처리한 데이터를 다시 소프트웨어에서 받아서 처리할 수 있도록 PL 과 PS 사이의 인터페이스를 자동 생성해 준다. 이렇게 SDSoC 는 소프트웨어로 처리하던 부분을 하드웨어를 사용하여 좀 더 빠르게 처리하도록 하드웨어 가속기를 만들어주고 관련 인터페이스를 자동 생성해주는 툴이라고 할 수 있다.

FPGA 를 사용하여 하드웨어를 설계하는 분야는 1990 년대 후반부터 많이 사용하기 시작하였다. 하드웨어를 설계하던 엔지니어는 스케메틱으로 하드웨어를 설계하는 것이 익숙하여 처음에는 스케메틱으로 많이 설계를 하였다. 이 때 설계를 하던 엔지니어들은 1 세대라고 한다면 이후에 2000 년대에 들어서 HDL 사용자가 많아지던 상황이었다. 이 때에도 변화가 있었지만 HDL 을 사용하지 않고 스케메틱을 고집하던 많은 엔지니어 선배들이 있었다. 하지만 결국 HDL 사용이 대세가 되고 스케메틱 사용자는 소수자가 되었다. 이 때 HDL 사용자를 2 세대라고 할 수 있다. 물론 지금도 스케메틱을 고수하는 엔지니어를 종종 볼 수 있다. 그리고 HDL 을 좀 더 추상화 레벨을 올려서 C/C++로 하드웨어를 설계하려는 시도는 2000 년대 초부터 있어왔다. 하지만 이런 시도는 주류가 되지 못하였고 대부분의 엔지니어들은 HDL 을 계속해서 사용하였다. 물론 지금도 HDL 사용자가 대다수이다. 그런데 자일링스는 HLS(High Level Synthesis)라는 상위 레벨의 하드웨어 설계 툴을 내놓았고 엔지니어들 사이에서 호평을 받고 있고 이를 사용하는 엔지니어들이 늘어나고 있는 추세이다. 이런 HLS 를 사용하는 세대를 3 세대라고 한다면 지금은 2 세대에서 3 세대로 넘어가는 변화의 시대라고 할 수 있다. 이제 가까운 미래에는 FPGA 사용자들은 대부분 HDL 이 아닌 HLS 을 사용하는 엔지니어가 많아질 것이다. 왜냐하면 디자인은 점점 더 복잡해지고 시장에서는 더 빠른 개발 속도를 요구하기 때문이다. 더 복잡해지고 더 커진 디자인을 더 빠르게 개발해야 한다면 HDL 로는 감당할 수 없게 될 것이다. 결국 소프트웨어에서 그랬던 것처럼 추상화 레벨을 올리는 수밖에 없다. 하드웨어 분야는 소프트웨어에 비해 변화 속도가 느리다고 할 수 있지만 조금씩 변화하고 있다.

1-3 이니프로 및 Ultra96 Training Kit 소개

이니프로는 FPGA 관련 제품과 교육 및 디자인 서비스를 제공하는 회사이다. 이니프로 홈페이지(www.inipro.net)에 들어가면 이니프로에서 판매하고 있는 다양한 제품들과 제공하고 있는 교육 및 디자인 서비스들을 확인할 수 있다.

본 교재 안에서 사용할 모든 소스코드는 이니프로 GitHub 사이트(<https://github.com/inipro>)의 ultra96 Ug Repository 의 2019 Tag 에 업데이트 되어 있다. 실습을 하려면 소스코드들이 필요하니 위 사이트에 접속하여 PC 에 다운로드 해야 한다.

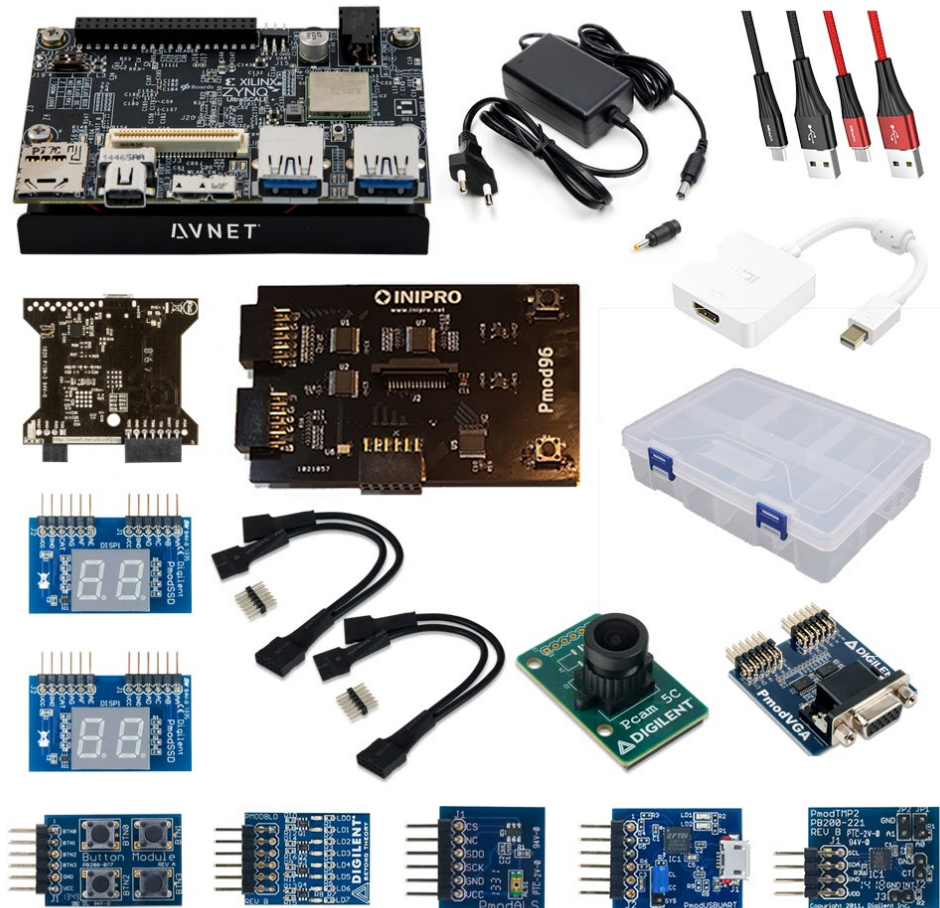


그림 1-2 Ultra96 Training Kit

그림 1-2 와 같이 Ultra96 Training Kit 는 Ultra96 보드에 여러 가지 실습을 하기 위해 필요한 모듈 및 액세서리들을 하나로 묶어서 만든 번들 제품으로 이니프로에서만 판매하는 제품이다. Ultra96 보드는 96Boards 의 한 종류이고 96Boards 는 Linaro 재단에서 ARM 코어가 포함된 여러 가지 SoC 칩들을 하나의 하드웨어 폼 팩터를 가진 보드들로 구성하여 하드웨어들 간의 호환성을 높이고자 만든 것이다. 96Boards 웹사이트 (www.96boards.org)에 들어가면 자세한 내용을 확인할 수 있다. 그래서 96Boards 의 폼 팩터를 가지고 있는 Ultra96 보드는 96Boards 의 폼 팩터를 가지고 있는 여러 가지 Mezzanine 보드들을 연결하여 사용할 수 있다. 마치 아두이노 쉴드와 같이 여러 가지 기능을 하는 보드들을 연결하여 사용할 수 있는 장점이 있다. 그리고 자일링스 MPSoC 를 탑재하고 있는 다른 보드들에 비해 상대적으로 저렴하여 교육용 키트로 적절하다.

본 교재에서는 Ultra96 Training Kit 를 사용한 여러 가지 실습들을 통해 Xilinx Zynq UltraScale+ MPSoC 를 어떻게 설계하는지에 대해서 다룬다.

1-4 SDSoC 톨 설치

SDSoC 를 설치하는 방법에 대해서 알아보자.

Step 1 자일링스 홈페이지에 접속하여 SDSoC 설치 파일을 다운로드 받는다.

1-1 자일링스 홈페이지(www.xilinx.com)에 접속한다.

1-2 SUPPORT 메뉴 아래 Downloads & Licensing 메뉴를 클릭한다.

1-3 Downloads 탭 중 SDSoC Development Environment 탭을 클릭한다.

1-4 좌측의 Version 은 2018.3 을 클릭하여 선택한다.

1-5 Web Installer 를 사용하는 방식과 Single File 을 다운로드하여 설치하는 방식이 있다. Web Install 방식은 작은 사이즈의 Web Installer 파일을 다운로드 받아서 설치할 때 인터넷을 통해 필요한 파일을 다운로드 받아서 설치하는 방식이고 Single File 방식은 설치에 필요한 전체 파일을 다운로드 받아서 설치하는 방식이다. 여기서는 Single File 방식을 사용할 예정이므로 SDx 2018.3 SFD 링크를 클릭한다.

1-4 Xilinx 로그인 화면이 나오면 이미 가입이 되어 있으면 기존 ID 로 로그인하고 가입되어 있지 않으면 계정을 만들어서 로그인해야 한다. 정상적으로 로그인이 되면 다운로드가 시작된다.

Step 2 설치파일을 실행하여 SDSoC 를 설치한다.

2-1 다운로드가 완료되면 다운로드한 파일의 압축을 푼 후 xsetup.exe 파일을 더블 클릭하여 Installer 를 실행한다.

2-2 그림 1-3 와 같은 설치 윈도우가 나오면 Next 버튼을 클릭한다.

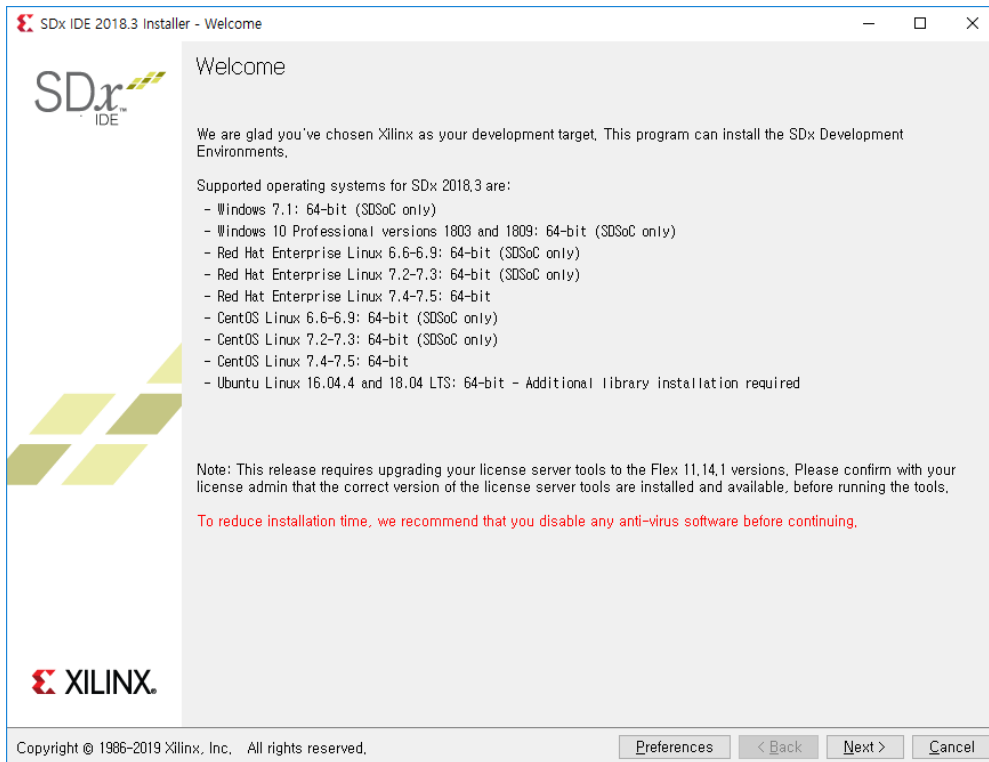


그림 1-3 SDSoC Installer Window 1

2-3 그림 1-4 와 같은 설치 윈도우가 나오면 I Agree 체크박스를 모두 선택 후 Next 버튼을 클릭한다.

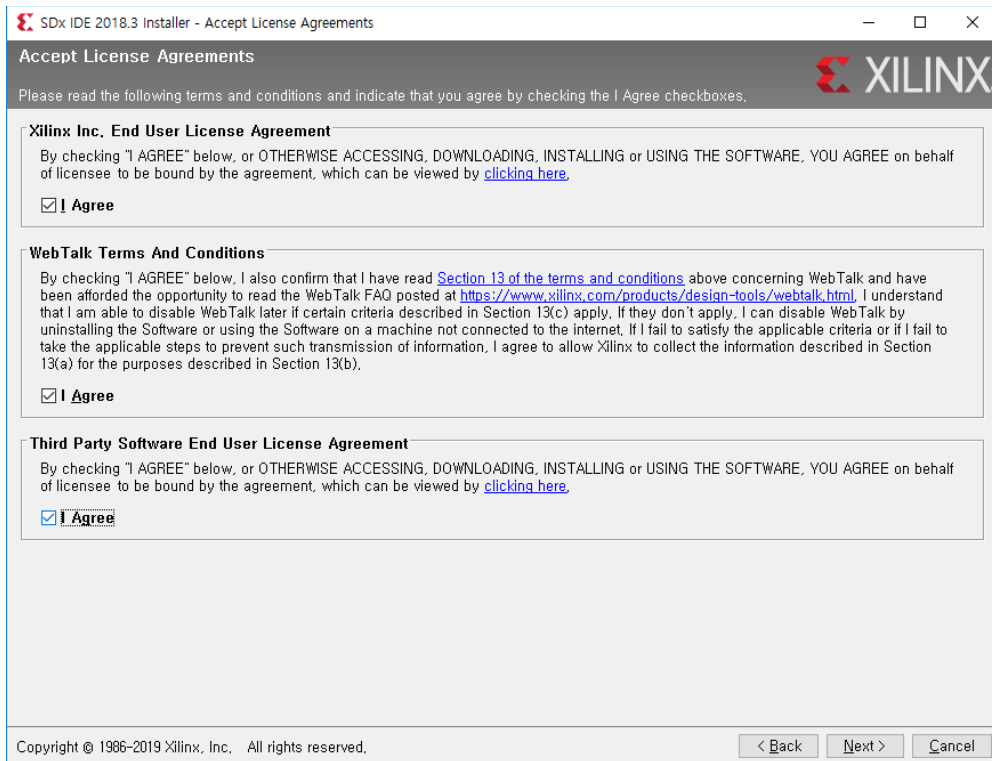


그림 1-4 SDSoC Installer Window 2

2-4 그림 1-5 와 같은 설치 윈도우는 어떤 툴과 디바이스들에 대한 내용들을 설치하지를 선택하는 윈도우이다. 디폴트 상태로 그대로 두고 Next 버튼을 클릭한다.

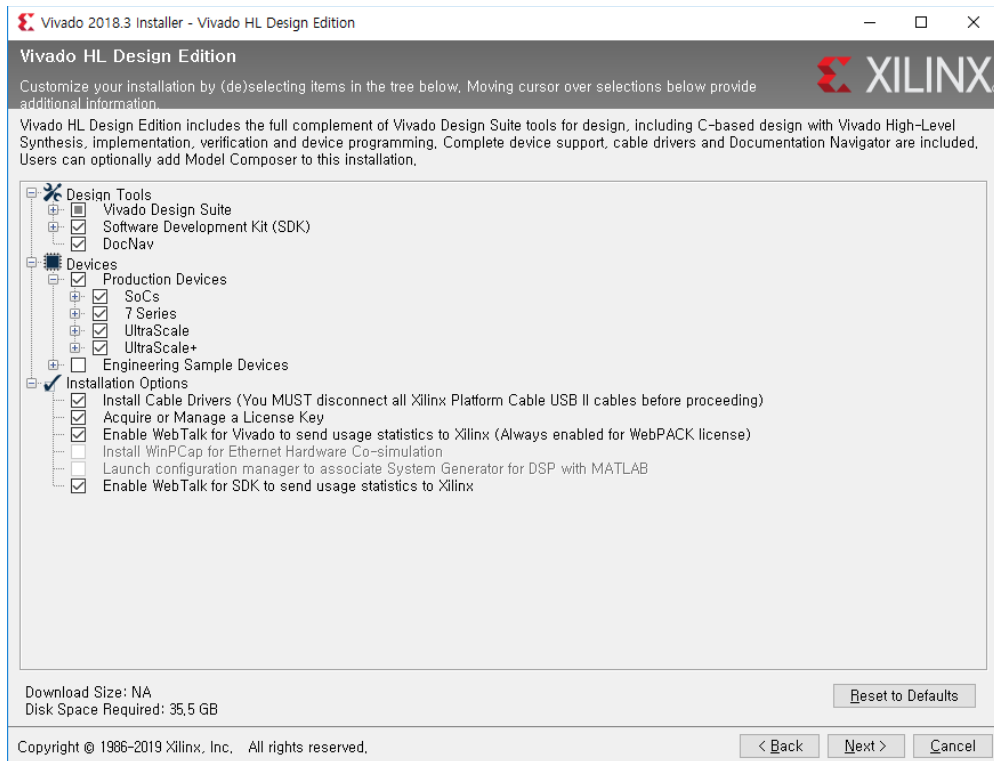


그림 1-5 SDSoC Installer Window 3

2-5 그림 1-6 과 같은 설치 윈도우는 툴을 설치할 경로를 선택하는 윈도우이다. 디폴트 경로에 설치하는 것을 추천하지만 다른 경로를 선택해도 무방하다. 경로 선택 후 Next 버튼을 클릭한다.

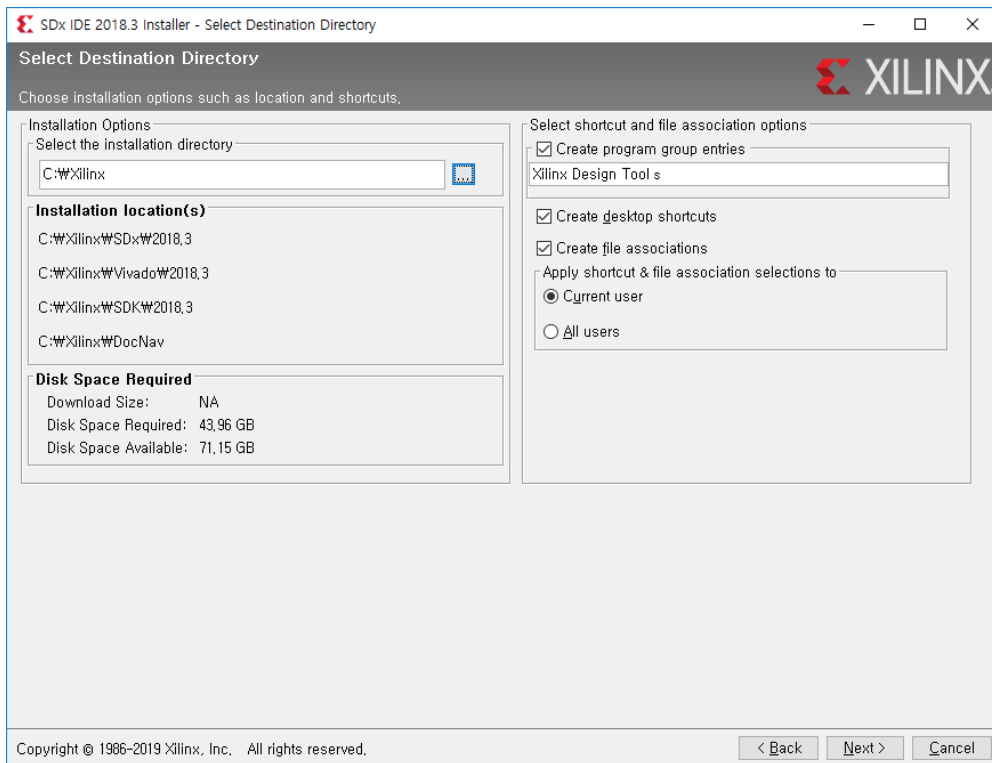


그림 1-6 SDSoC Installer Window 4

2-6 그림 1-7 과 같은 설치 윈도우는 설치할 내용을 종합한 화면이다. 내용을 살펴보고 특이사항이 없으면 Install 버튼을 클릭하여 설치를 시작한다.

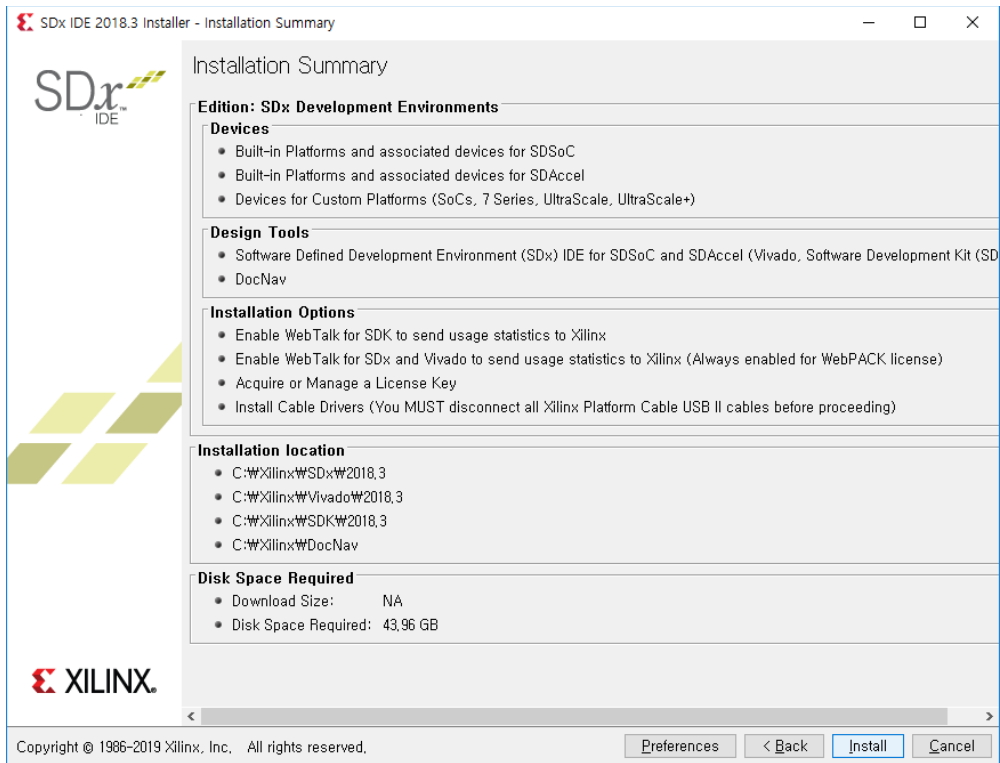


그림 1-7 SDSoC Installer Window 5

2-7 그림 1-8 과 같이 설치가 완료되었다는 윈도우가 나오면 확인 버튼을 클릭한다.

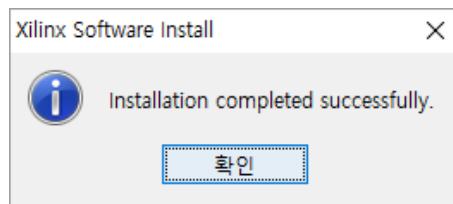


그림 1-8 Xilinx Software Install Window

2-8 설치가 완료되면 그림 1-9 와 같이 License Manager 툴이 활성화된다. 왼쪽 목록에서 Load License 를 선택 후 오른쪽의 Copy License 버튼을 클릭한다. 라이선스 파일 선택하는 화면이 활성화되면 Xilinx Product Licensing Site 에서 발급받은 라이선스 파일을 선택한다. 본 교재에서는 Ultra96 보드를 사용하므로 Ultra96 보드에 포함된 Voucher 를 사용하여 라이선스를 발급받아서 사용한다. Voucher 를 사용한 라이선스 발급 방법에 대해서는 Step 3 를 참조한다.

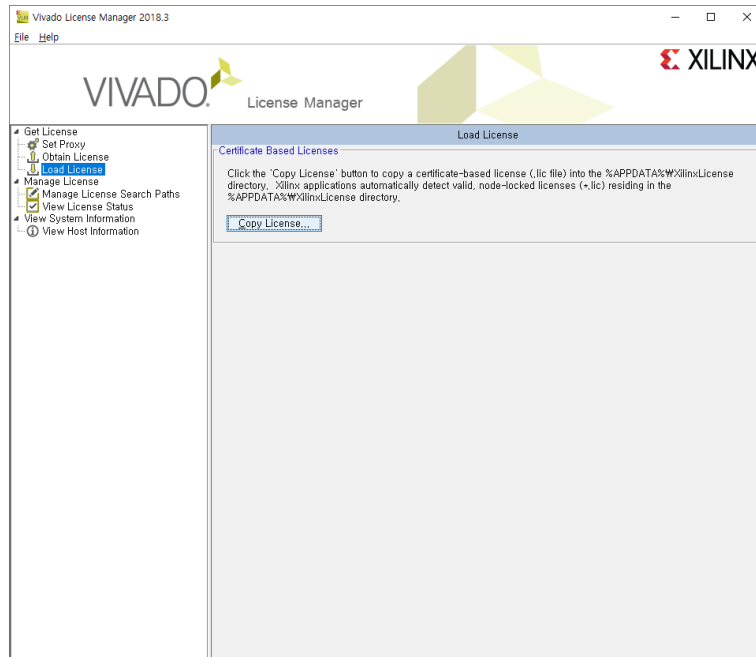


그림 1-9 Vivado License Manager Window

2-9 그림 1-10 과 같이 라이선싱 설치가 완료되었다는 메시지 창이 나오면 확인 버튼 클릭 후 License Manager 를 닫으면 SDSoC 설치가 완료된다.



그림 1-10 Vivado License Manager Window

Step 3 Voucher 를 사용하여 라이선스를 발급 받는다.

3-1 자일링스 홈페이지(www.xilinx.com)에 접속한다.

3-2 SUPPORT 메뉴 아래 Downloads & Licensing 메뉴를 클릭한다.

3-3 Downloads 아래 Licensing Help 메뉴를 클릭한다.

3-4 Xilinx Product Licensing Site 링크를 클릭하고 로그인 화면인 나오면 로그인을 한다.

3-5 그림 1-11 과 같이 Xilinx Product Licensing Site 가 나오면 Voucher Code 를 입력하는 곳에 Ultra96 Board 에 포함된 Voucher Code 를 입력한 후 Redeem Now 버튼을 클릭한다.

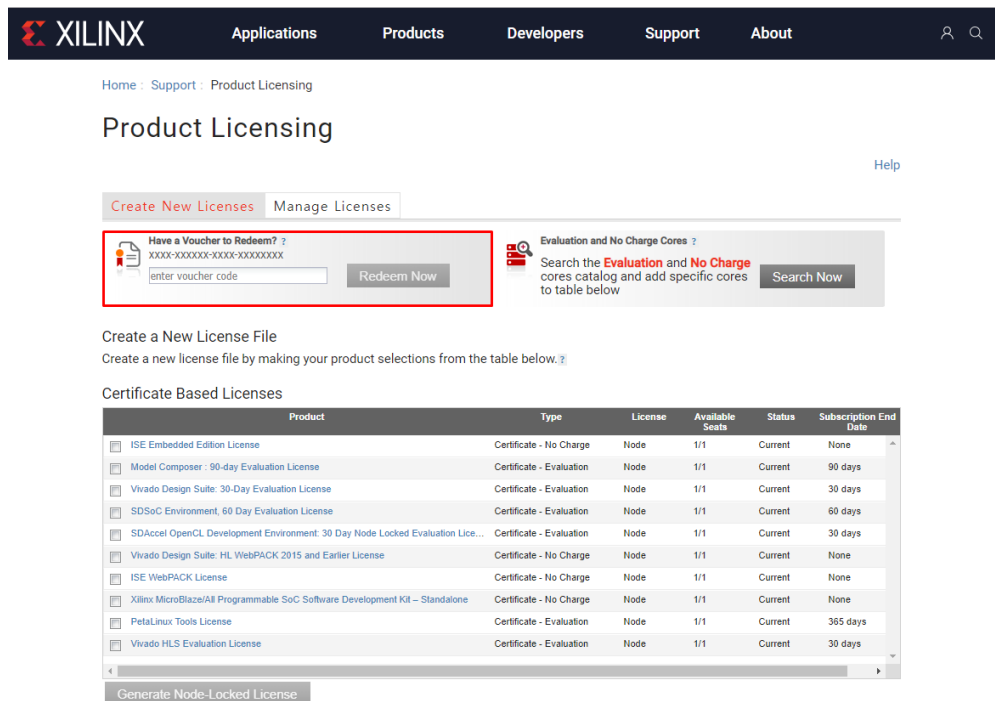


그림 1-11 Xilinx Product Licensing Site

3-6 Ultra96 용 라이선스가 생성되면 해당 라이선스를 선택한 후 Generate Node License 버튼을 클릭한다.

3-7 Genrate Node License 윈도우가 나오면 Host ID 에 PC 의 Host ID 를 추가한 후 Next 버튼을 클릭한다.

3-8 Review 창이 나오면 내용을 확인 후 Next 버튼을 클릭한다.

3-9 자일링스 계정에 등록된 이메일로 라이선스 발급 되었다는 메시지와 함께 라이선스 발급 완료 창이 나오면 창을 닫고 자일링스에서 받은 이메일에 첨부된 라이선스 파일을 디스크에 저장한다.

3-10 2-9 에서 설명한 것과 같이 License Manager 를 사용하여 저장된 라이선스 파일을 로딩하면 된다. (※ License Manager 윈도우는 Vivado 를 실행한 후 Help 메뉴 아래 Manage License 메뉴를 통해 다시 실행할 수 있다.)

Chapter 2 Vivado 를 사용한 FPGA 설계

2-1 Vivado Design Flow

2-2 LED 카운터 구현

2-3 7-Segment 디지털시계 구현

2-4 VGA 출력 구현

2-1 Vivado Design Flow

Vivado 를 사용하여 HDL 코드를 합성하고 Bitstream 파일(FPGA 를 Configuration 할 수 있는 파일)을 생성한 후 Ultra96 보드의 Zynq MPSoC UltraScale+의 PL 에 다운로드하여 원하는 하드웨어를 구현하는 Vivado Design Flow 에 대해서 알아보자.

Step 1 이니프로 GitHub 사이트에서 데이터를 다운받고 BDF 를 설치한다.

1-1 먼저 실습에 필요한 데이터를 받기 위해 이니프로 GitHub 사이트 (<https://github.com/inipro>)에 접속한다.

1-2 ultra96_ug Repository 에서 Tag 를 2019 로 선택한다.

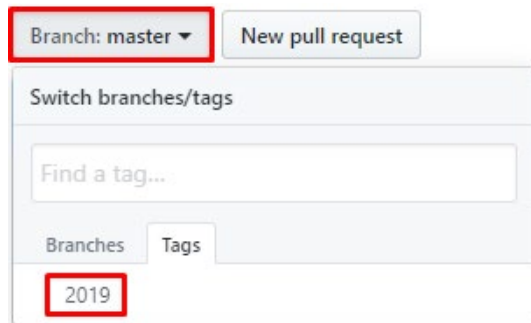


그림 2-1 Tags in GitHub site

1-3 Clone or download 라고 쓰여 있는 녹색 버튼을 클릭한 후 Download ZIP 을 클릭하여 ZIP 파일을 다운로드 받는다.

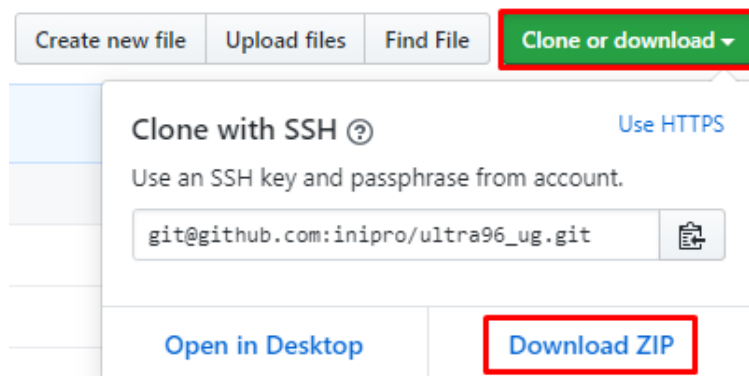


그림 2-2 Download Button

1-3 다운로드 받은 ultra96 Ug-2019.zip 파일을 C:\WExercises 폴더에 복사한 후 압축을 푼다.

1-4 C:\WExercises\ultra96 Ug-2019\data\ultra96v1 폴더를 <SDSoC install location>\Vivado\2018.3\data\boards\board_files 폴더 안에 복사한다.

Step 2 Vivado 를 실행하여 프로젝트를 생성한다.

2-1 바탕화면 또는 시작화면에서 Vivado 를 클릭하여 Vivado 를 실행한다.

2-2 그림 2-3 과 같은 Vivado 실행 초기 윈도우가 나오면 Quick Start 아래 Create Project 를 클릭한다.

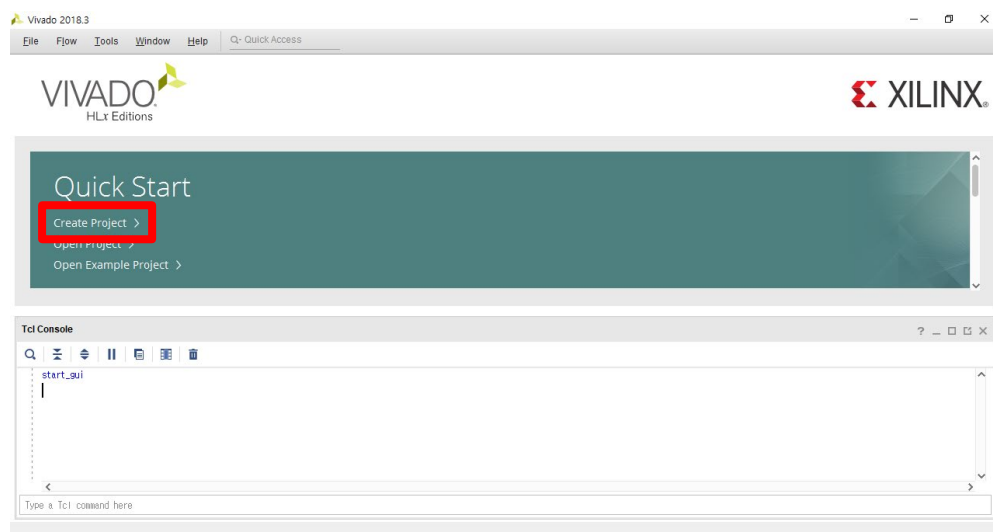


그림 2-3 Vivado 실행 초기 윈도우

2-3 그림 2-4 와 같은 프로젝트 생성을 위한 윈도우가 나오면 Next 버튼을 클릭한다.

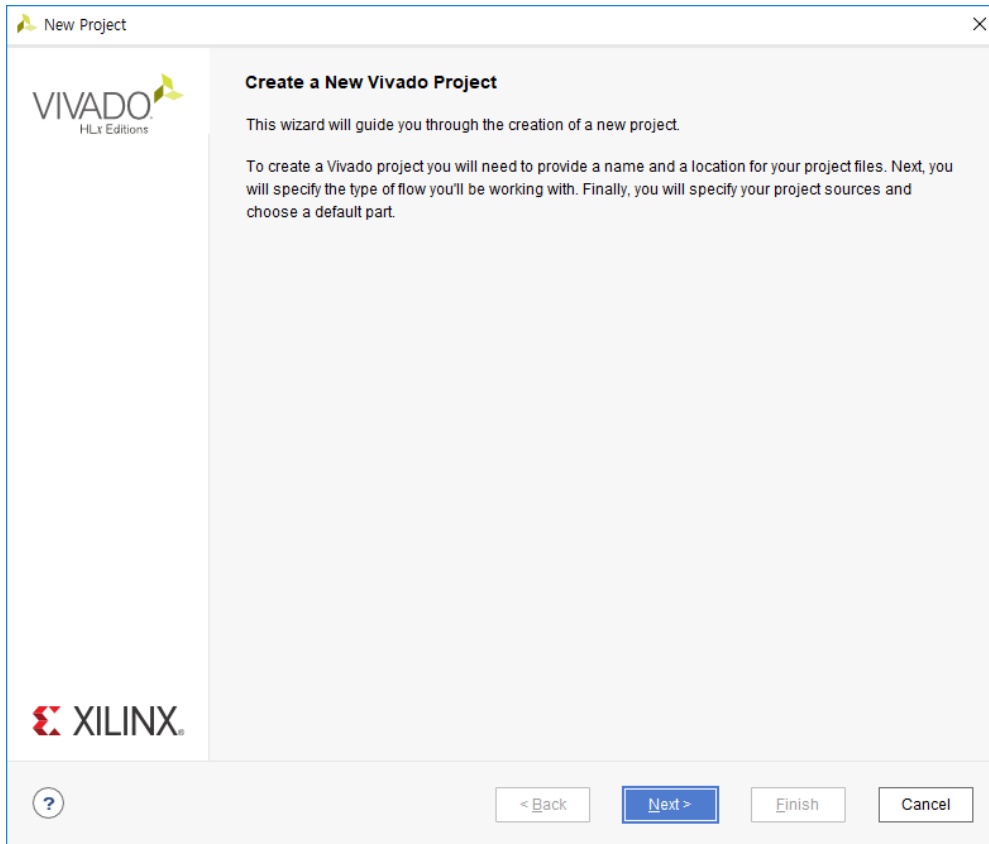


그림 2-4 New Project Window 1

2-4 그림 2-5 와 같이 프로젝트 이름과 경로를 선택하는 윈도우가 나오면 ultra96_ug-2019 경로를 선택하고 프로젝트 이름은 EX2-1 을 입력한다. 폴더가 생성되어 있지 않으면 폴더를 생성해야 하므로 Create project subdirectory 체크박스를 체크하고 Next 버튼을 클릭한다.

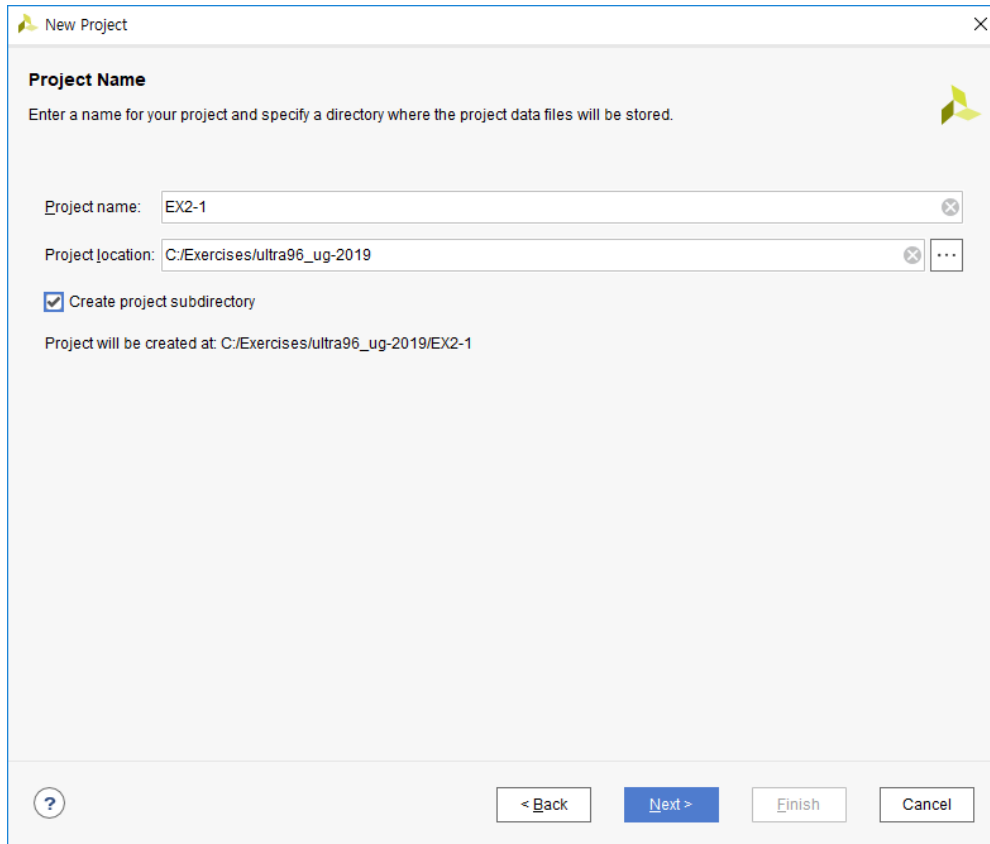


그림 2-5 New Project Window 2

2-5 그림 2-6 과 같이 프로젝트 타입을 선택하는 윈도우가 나오면 RTL Project 를 선택 후 Next 버튼을 클릭한다.

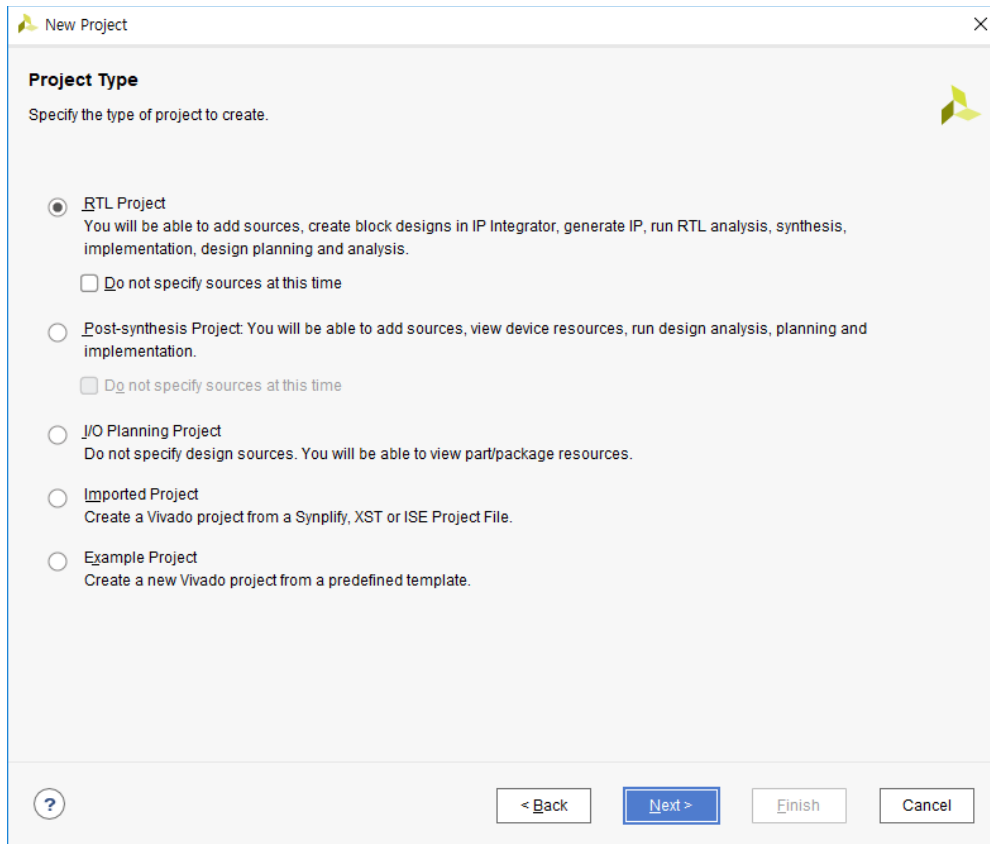


그림 2-6 New Project Window 3

2-6 그림 2-7 과 같이 기존에 작성된 소스파일을 프로젝트에 추가하는 윈도우가 나오면 Add Files 버튼을 클릭한 후 ultra96_ug-2019\sources\EX2-1 폴더 안에 simple_assign.v(Verilog) 또는 simple_assign.vhd(VHDL)을 선택하여 프로젝트에 추가한다.

2-7 하단의 Copy sources into project 체크 박스를 클릭하여 선택한 후 Target language 를 추가한 소스파일 타입으로 수정한 후 Next 버튼을 클릭한다.

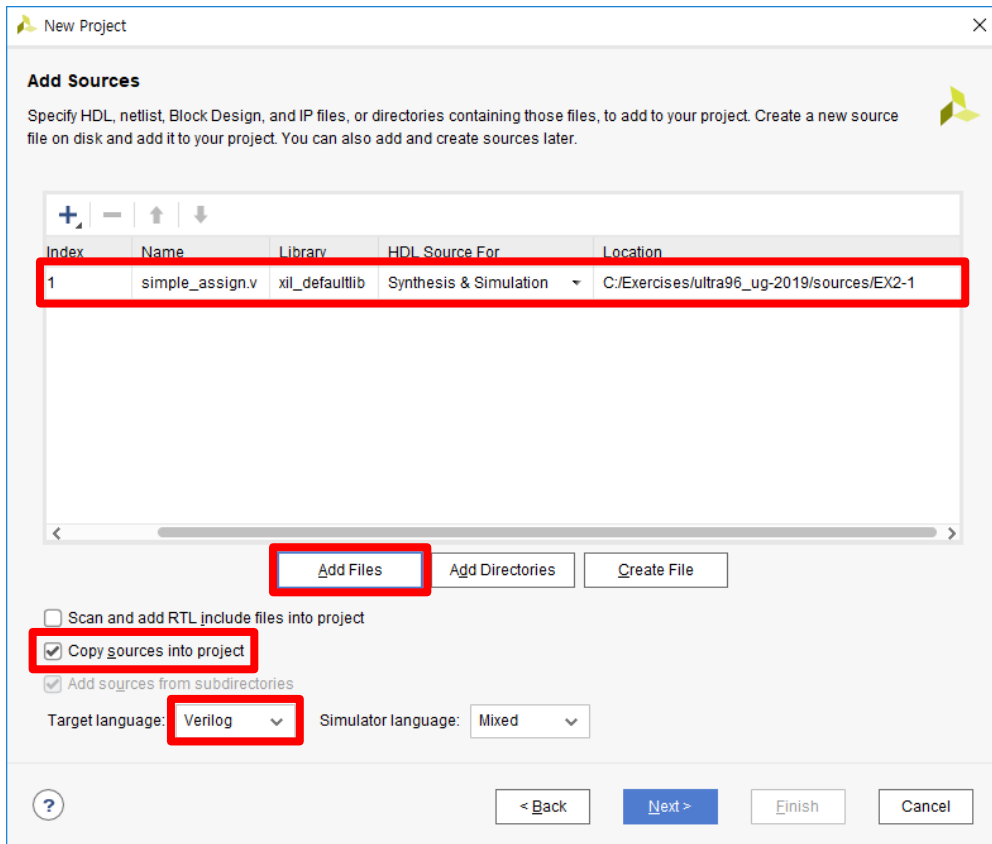


그림 2-7 New Project Window 4

2-8 그림 2-8 과 같이 Constraints 파일을 프로젝트에 추가하는 윈도우가 나오면 Add Files 버튼을 클릭한 후 ultra96_ug-2019\sources 폴더 안에 ultra96_training_kit.xdc (Xilinx Design Constraints)파일을 선택하여 프로젝트에 추가한다.

2-9 하단의 Copy constraints files into project 체크 박스를 클릭하여 선택한 후 Next 버튼을 클릭한다.

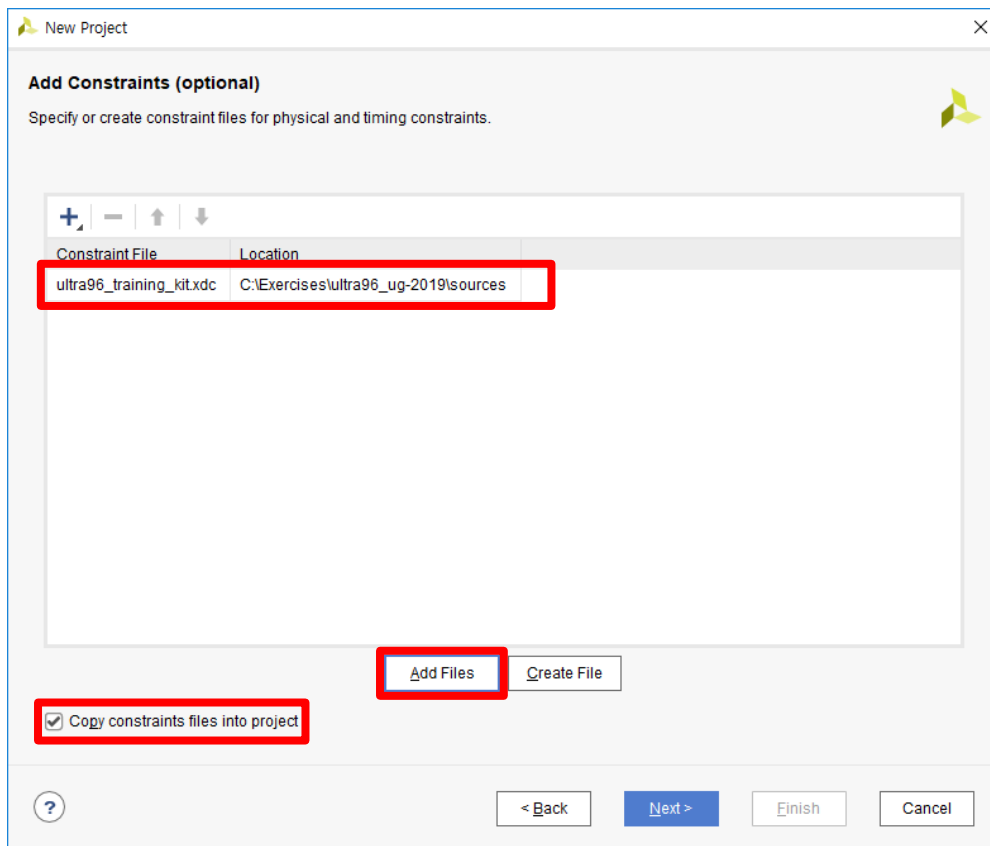


그림 2-8 New Project Window 5

2-10 그림 2-9 와 같이 Xilinx 디바이스 또는 보드를 선택하는 윈도우가 나오면 윈도우 상단의 Parts | Boards 로 되어 있는 부분에서 Boards 를 클릭하여 선택한다.

2-11 Vivado 안에 설치되어 있는 Board File 들이 나오면 Ultra96v1 Evaluation Platform 을 클릭하여 선택한 후 Next 버튼을 클릭한다.

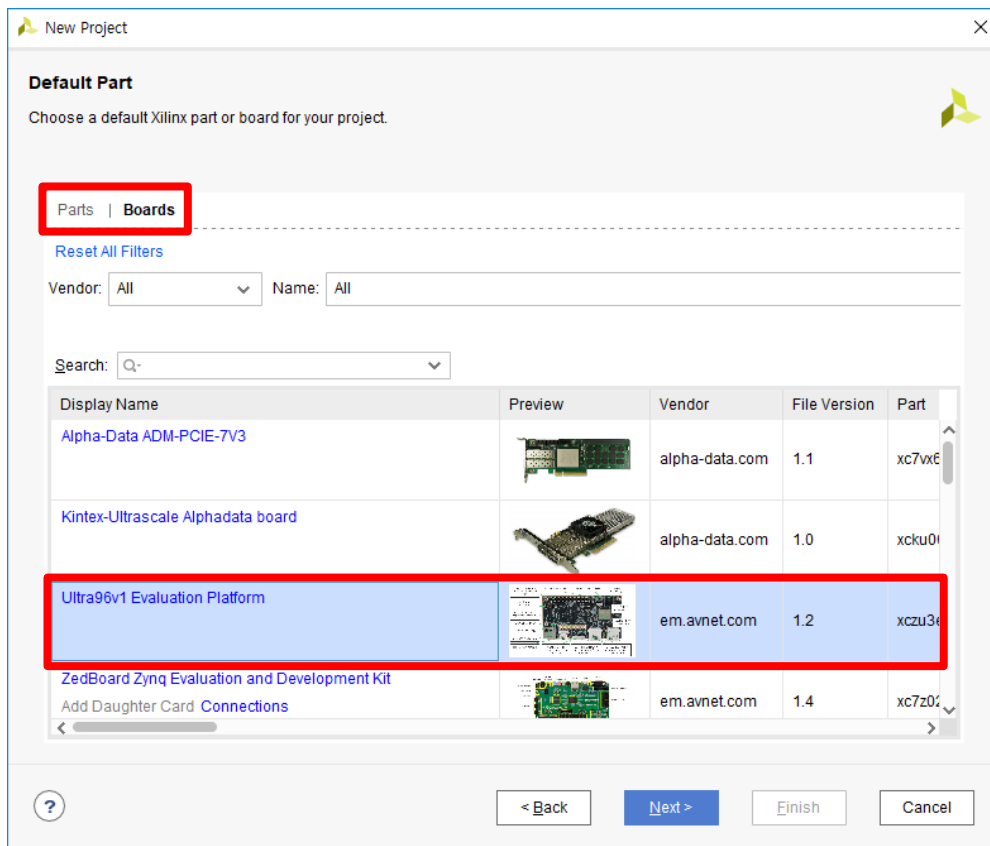


그림 2-9 New Project Window 6

2-12 그림 2-10 과 같이 새로 만든 프로젝트에 대한 내용들을 정리한 윈도우가 나오면 내용을 확인한 후 Finish 버튼을 클릭한다.

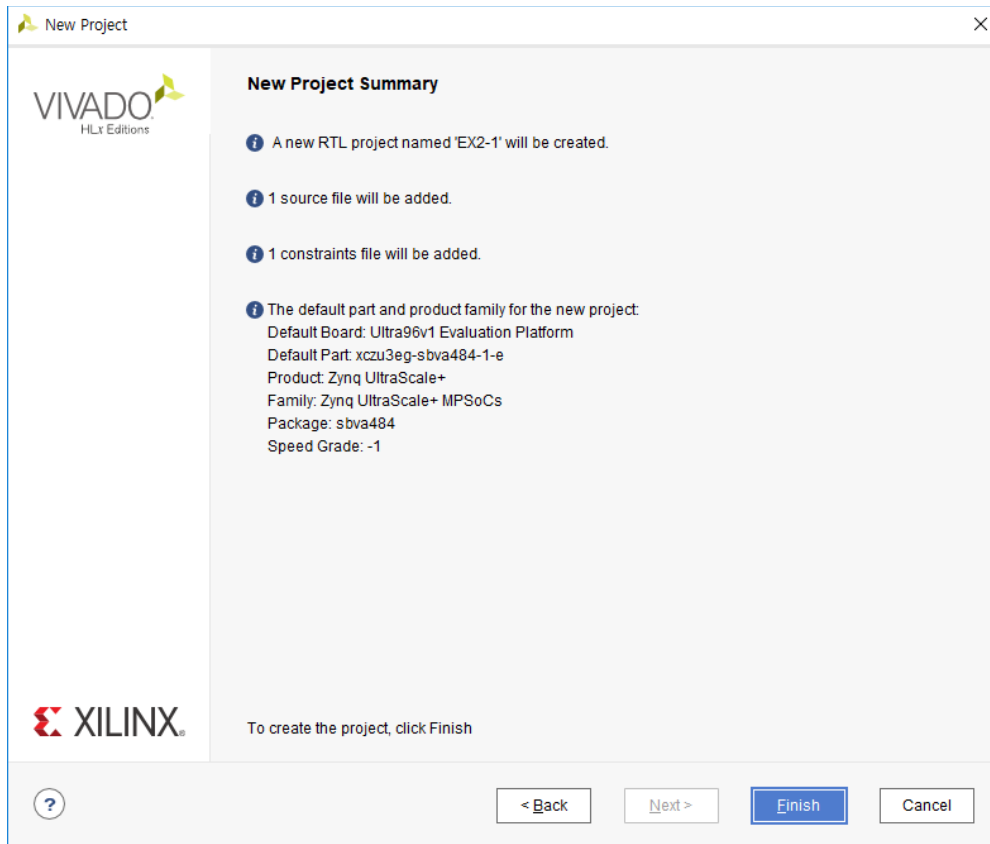


그림 2-10 New Project Window 7

2-13 그림 2-11 과 같이 Vivado 윈도우가 나오면 프로젝트 생성 완료이다.

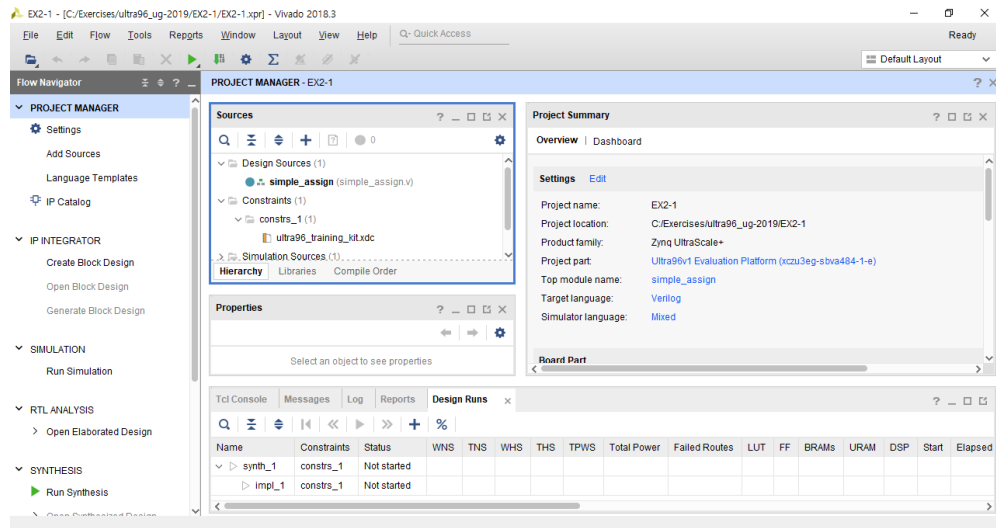


그림 2-11 Vivado Window 1

Step 3 Bitstream 을 생성하여 Ultra96 Board 에 다운로드 한다.

3-1 Project Manager 의 좌측 상단의 Sources 패널을 보면 프로젝트를 생성할 때 추가한 디자인 소스파일과 Constraints 파일이 추가되어 있는 것을 확인할 수 있다.

3-2 프로젝트에 추가된 디자인 소스파일과 Constraints 파일을 합성하여 Ultra96 보드에 다운로드 하려면 Bitstream 을 생성해야 한다. 좌측의 Flow Navigator 의 스크롤을 아래로 내려서 PROGRAM AND DEBUG 아래 Generate Bitstream 을 클릭한다.

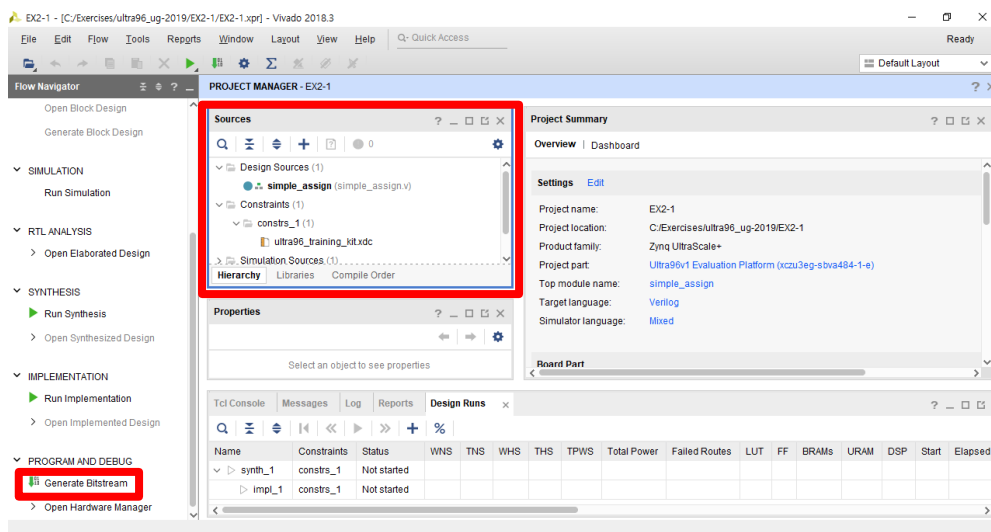


그림 2-12 Vivado Window 2

3-3 그림 2-13 과 같은 Launch Runs 윈도우는 Bitstream 을 만드는데 CPU 리소스를 얼마나 사용할 지 선택하는 윈도우이다. 윈도우 중앙에 Number of jobs 는 CPU 의 쓰레드를 몇 개 사용할 지 선택하는 메뉴이다. 자일링스는 컴퓨터에서 사용 가능한 총 쓰레드 개수의 50%를 사용하는 것을 추천하고 있다. Bitstream 생성을 빨리 하려면 쓰레드를 최대한 많이 사용하는 것이 좋겠지만 100%를 사용하면 시스템에서 처리해야 할 다른 일을 할 수가 없으니 100%를 모두 사용하는 것은 좋지 않다. 그러므로 50%~100% 사이 중에 적당한 개수를 선택한 후 OK 버튼을 클릭한다.

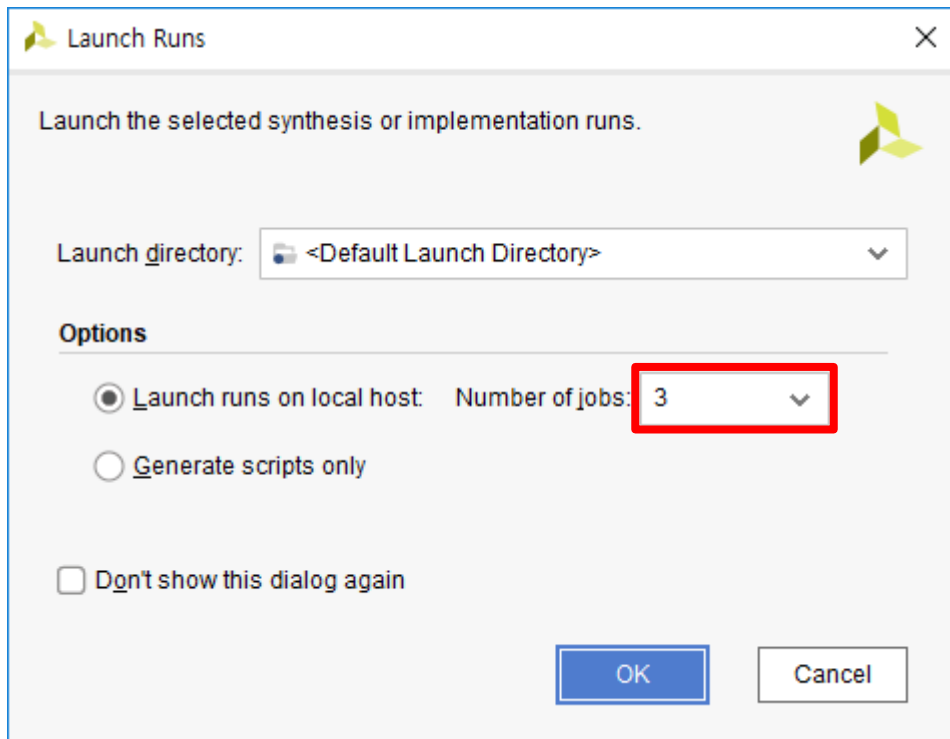


그림 2-13 Launch Runs Window

Vivado 는 Bitstream 을 만들기 위해 Elaboration, Synthesis, Implementation 및 Generate Bitstream 과정을 수행한다. Elaboration 은 RTL 코드를 분석하는 과정이고 Synthesis 는 RTL 코드를 합성하여 산업표준의 하드웨어 표현 방식인 Netlist 를 만드는 과정이고 Implementation 은 Netlist 를 프로젝트를 생성할 때 선택한 자일링스 디바이스에서 구현하기 위해 디바이스의 리소스에 맞게 변환하여 P&R(Place&Route)을 하는 과정이다. 이 결과물을 FPGA 를 Configuration 할 수 있는 .bit(Bitstream)파일을 만드는 과정이 Generate Bistream 과정이다.

3-4 비트스트림 파일 생성이 완료되면 그림 2-14 와 같은 윈도우가 나온다. 생성한 비트스트림 파일을 Ultra96 보드에 다운로드 하기 위해 Open Hardware Manager 를 선택한 후 OK 버튼을 클릭한다. (또는 좌측의 Flow Navigator 에서 Open Hardware Manager 를 클릭한다.)

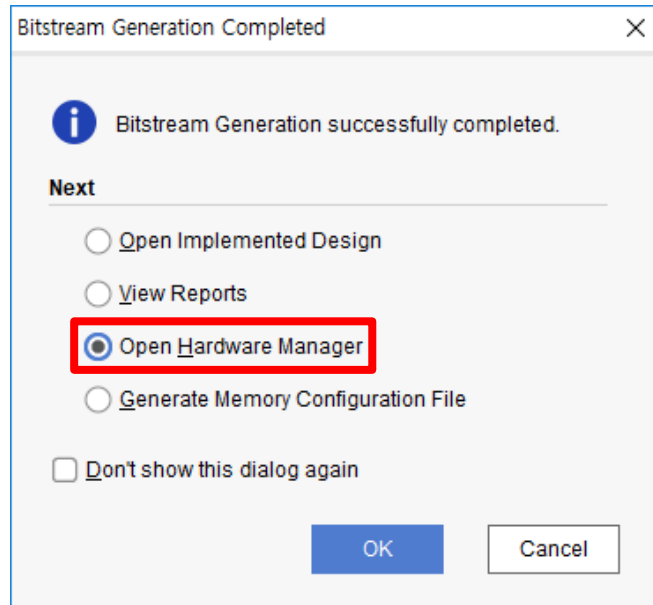


그림 2-14 Bitstream Generation Completed Window

3-5 Hardware Manager 가 열리면 Ultra96 보드에 다운로드 할 수 있도록 보드를 준비한다. 먼저 Ultra96 보드에 Pmod96 보드, Ultra96 USB-to-JTAG/UART Pod 와 전원 아답터를 연결한다.

3-6 Ultra96 USB-to-JTAG/UART Pod 의 USB 포트에 USB 케이블을 연결하고 반대쪽은 PC 와 연결한다. 연결이 되면 USB 전원이 들어와서 Ultra96 USB-to-JTAG/UART Pod 의 LED 가 켜진다.

3-7 디자인 소스 파일에서 구현한 로직이 정상 동작하는지 확인을 위해 Pmod 8LD 를 Pmod96 보드의 PMOD_A 커넥터에 연결하고 Pmod BTN 을 Pmod96 보드의 PMOD_C 커넥터에 연결한다. (※ VCC 와 GND 핀을 주의하여 연결한다.)

3-8 마지막으로 Pmod96 보드의 2 개의 버튼 중 Power 버튼을 눌러서 파워를 켜주면 준비 완료이다.

3-9 그림 2-15 와 같이 Hardware Manager 가 열리면 상단의 녹색 바 안에 Open target 을 클릭한 후 메뉴 중 Auto Connect 를 클릭한다.

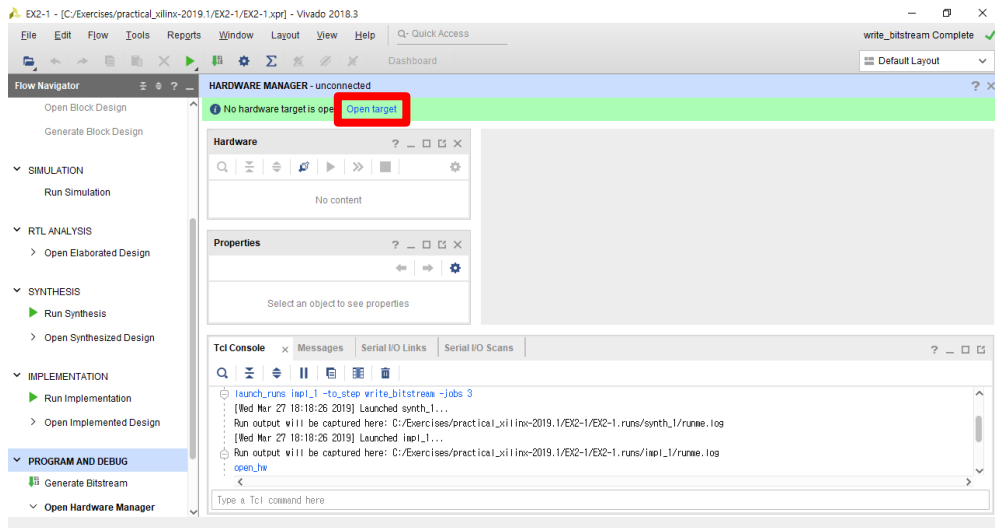


그림 2-15 Vivado Window 3

3-10 PC 와 Ultra96 보드가 정상적으로 연결되어 그림 2-16 과 같이 녹색 바 안에 Program device Refresh device 가 나오면 Program device 를 클릭한다.

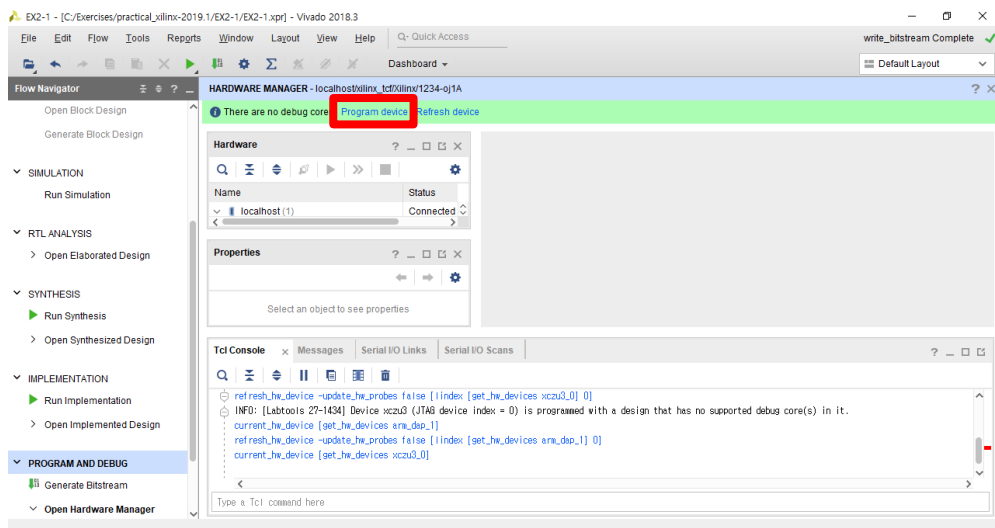


그림 2-16 Vivado Window 4

3-11 그림 2-17 과 같이 Program Device 윈도우가 나오면 다운로드할 Bitstream file 이 디폴트로 선택되어 있으니 그대로 두고 Program 버튼을 클릭한다.

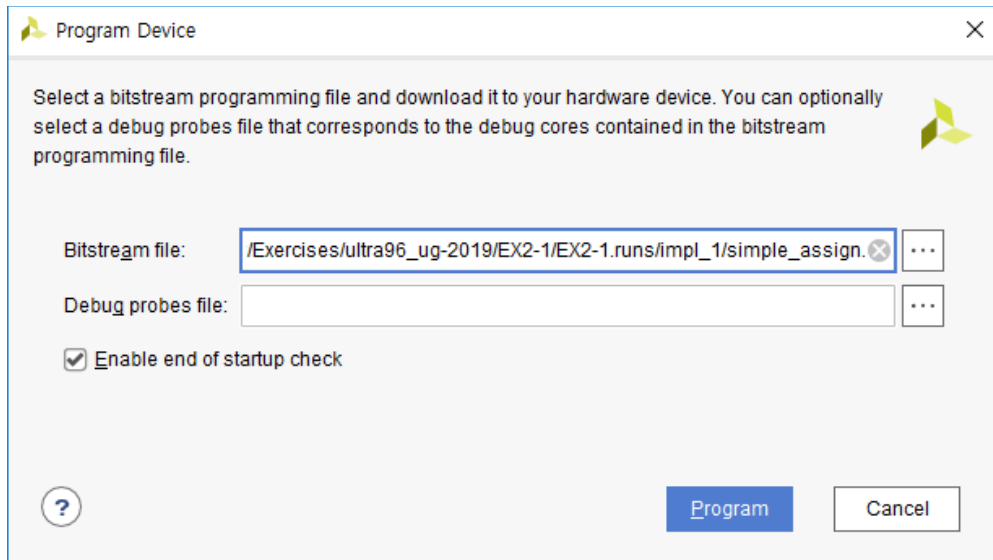


그림 2-17 Program Device Window

3-12 프로그램이 완료되면 Pmod 8LD 의 8 개 LED 중 4 개는 켜져 있고 4 개는 꺼져 있는 것을 확인할 수 있다. Pmod96 의 Pmod I/O 는 모두 Pull-up 저항이 있어서 초기에는 모두 High 상태이다. 그런데 현재 로직으로 PMOD_C 를 PMOD_A 에 연결한 상태이다. 그래서 Pmod BTN 이 모두 눌러지지 않은 상태이니 LOW 값이 PMOD_C 에 assign 되어 4 개의 LED 는 꺼지게 되고 Pmod BTN 이 연결되지 않은 4 개는 켜진 상태가 된다. 여기에서 버튼을 한 개 눌러보면 꺼져 있는 LED 중 하나가 켜지는 것을 확인할 수 있다. Pmod BTN 을 Pmod 커넥터의 위쪽 또는 아래쪽으로 바꾸어서 연결해보면 반대로 켜지는 것을 확인할 수 있다.

3-13 HDL 소스코드를 열어서 구현된 내용을 검토한다.

3-14 Vivado 윈도우를 끄고 Pmod96 의 Power 버튼을 누른 상태로 10 초정도 유지하여 Ultra96 의 전원을 끈다.

2-2 LED 카운터 구현

Step 1 Ultra96 보드의 Zynq MPSoC UltraScale+의 PL 에 LED 카운터를 구현해 보자.

1-1 Chapter 2-1 의 Setp 2 를 참조하여 프로젝트를 생성한다.

1-2 소스를 추가할 때 simple_assign 파일 대신 led_counter.v 또는 led_counter.vhd 를 추가한다.

1-3 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 생성한다.

1-4 PMOD_A 커넥터에 VCC 와 GND 핀을 주의하여 Pmod 8LD 를 연결한다.

1-5 Pmod96 보드의 2 개의 버튼 중 Power 버튼을 눌러서 파워를 켜다.

1-6 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 다운로드한다.

1-7 프로그램이 완료되면 Pmod 8LD 의 8 개 LED 가 카운팅 되어 불이 깜빡이는 것을 확인할 수 있다.

1-8 HDL 소스코드를 열어서 구현된 내용을 검토한다.

1-9 Vivado 윈도우를 끄고 Pmod96 의 Power 버튼을 누른 상태로 10 초정도 유지하여 Ultra96 의 전원을 끈다.

2-3 7-Segment 디지털시계 구현

Step 1 Ultra96 보드의 Zynq MPSoC UltraScale+의 PL 에 7-Segment 디지털시계를 구현해 보자.

1-1 Chapter 2-1 의 Setp 2 를 참조하여 프로젝트를 생성한다.

1-2 소스를 추가할 때 simple_assign 파일 대신 digital_watch.v 또는 digital_watch.vhd 를 추가한다.

1-3 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 생성한다.

1-4 Pmod SSD 2 개를 VCC 와 GND 핀을 주의하여 2x6-pin to Dual 6-pin Pmod Splitter Cable 에 연결한다.

1-5 PMOD_A 와 PMOD_B 커넥터에 VCC 와 GND 핀을 주의하여 2 개의 2x6-pin to Dual 6-pin Pmod Splitter Cable 을 연결한다.

1-6 Pmod96 보드의 2 개의 버튼 중 Power 버튼을 눌러서 파워를 켜다.

1-7 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 다운로드한다.

1-8 프로그램이 완료되면 2 개의 Pmod SSD 에 디지털시계한 구현된 것을 확인할 수 있다.

1-9 HDL 소스코드를 열어서 구현된 내용을 검토한다.

1-10 Vivado 윈도우를 끄고 Pmod96 의 Power 버튼을 누른 상태로 10 초정도 유지하여 Ultra96 의 전원을 끈다.

2-4 VGA 출력 구현

Step 1 Ultra96 보드의 Zynq MPSoC UltraScale+의 PL 에 VGA 출력을 구현해 보자.

- 1-1 Chapter 2-1 의 Setp 2 를 참조하여 프로젝트를 생성한다.
- 1-2 소스를 추가할 때 simple_assign 파일 대신 vga_out.v 또는 vga_out.vhd 를 추가한다.
- 1-3 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 생성한다.
- 1-4 PMOD_A 와 Pmod_B 커넥터에 VCC 와 GND 핀을 주의하여 Pmod VGA 를 연결한다.
- 1-5 VGA 케이블을 한쪽은 Pmod VGA 에 연결하고 한쪽은 모니터에 연결한다.
- 1-6 Pmod96 보드의 2 개의 버튼 중 Power 버튼을 눌러서 파워를 켜다.
- 1-7 Chapter 2-1 의 Setp 3 을 참조하여 .bit(Bitstream)파일을 다운로드한다.
- 1-8 프로그램이 완료되면 모니터 화면에 출력된 영상을 확인할 수 있다.
- 1-9 HDL 소스코드를 열어서 구현된 내용을 검토한다.
- 1-10 Vivado 윈도우를 끄고 Pmod96 의 Power 버튼을 누른 상태로 10 초정도 유지하여 Ultra96 의 전원을 끈다.

Chapter 3 SDK 를 사용한 Zynq 설계

3-1 SDK Development Flow

3-2 IIC Interface

3-3 SPI Interface

3-4 UART Interface

3-5 Custom IP 생성

3-6 UART 로 Custom IP 컨트롤

3-7 SD Card 부팅

3-1 SDK Development Flow

3-2 IIC Interface

3-3 SPI Interface

3-4 UART Interface

3-5 Custom IP 생성

3-6 UART 로 Custom IP 컨트롤

3-7 SD Card 부팅

- 감사합니다. -