

AMD SoC(Zynq, Zynq UltraScale+, Versal 등) 개발 보드에 대한 정식 보드 파일(**board file**)이 있다면, **Vivado**에서 **PS(Processing System)**의 핵심 설정은 추가 정보 없이 자동으로 구성할 수 있으며, 이를 통해 XSA 파일 생성이 가능합니다.

요약

보드 파일은 특정 개발 보드의 하드웨어 정보를 담고 있는 설명서와 같습니다. 여기에는 PS와 연결된 DDR 메모리, 클럭, 주요 페리페럴(Peripheral) 등의 설정 정보가 모두 포함되어 있습니다. 따라서 Vivado에서 프로젝트를 생성할 때 이 보드 파일을 지정하면, '**Block Automation**' 기능을 통해 클릭 한 번으로 복잡한 PS 설정을 자동으로 완료할 수 있습니다.

상세 설명

1. 보드 파일(Board File)의 역할 💡

보드 파일은 단순한 파일이 아니라, 특정 보드에 대한 다음과 같은 중요한 하드웨어 설정 정보를 담고 있는 파일들의 집합입니다.

- **SoC** 파트 정보: 보드에 장착된 정확한 Zynq 또는 Versal 칩 모델명
- **DDR** 메모리 설정: DDR 메모리의 종류(DDR3, DDR4, LPDDR4 등), 용량, 타이밍, 핀 연결 등 매우 복잡하고 중요한 정보
- 클럭(**Clock**) 설정: PS에 입력되는 메인 클럭의 주파수 및 소스 정보
- 주요 페리페럴(**Peripheral**) 설정: UART, I2C, SPI, Ethernet, USB 등 보드에 물리적으로 연결된 주변장치들이 SoC의 어떤 MIO 핀에 연결되어 있는지에 대한 정보
- 리셋 및 전압 정보 등

이 정보들이 미리 정의되어 있기 때문에 사용자가 데이터시트를 일일이 찾아보며 수동으로 설정할 필요가 없습니다.

2. 보드 파일을 이용한 XSA 생성 과정

Vivado에서 보드 파일을 사용하여 XSA를 생성하는 일반적인 과정은 다음과 같습니다.

1. **Vivado** 프로젝트 생성:

- 새 프로젝트를 만들 때 "RTL Project"를 선택합니다.
- "Default Part" 단계에서 "Parts" 대신 **"Boards"** 탭을 선택하고, 가지고 있는 개발 보드(예: ZCU102, Kria KV260 등)를 검색하여 선택합니다. (만약 보드 파일이 목록에 없다면, 제조사 홈페이지에서 다운로드하여 Vivado에 설치해야 합니다.)

2. **Block Design** 생성:

- "IP Integrator" 메뉴에서 "Create Block Design"을 선택하여 새로운 블록 다이어그램을 엽니다.

3. **SoC IP** 추가 및 자동 설정:

- 다이어그램에서 '+' 버튼을 눌러 "Zynq UltraScale+ MPSoC" 또는 사용하는 칩에 맞는 IP를 추가합니다.
- IP가 추가되면 다이어그램 상단에 녹색 배너로 *****Run Block Automation*****이 나타납니다.
- 이 버튼을 클릭하면 Vivado가 보드 파일에 정의된 정보를 읽어와 **DDR** 컨트롤러, 클럭, 페리페럴 등 모든 **PS** 관련 설정을 자동으로 구성해 줍니다.

4. **XSA** 파일 생성:

- PS 설정이 완료된 블록 디자인의 유효성을 검사(Validate Design)합니다.
- "Sources" 탭에서 블록 디자인 파일을 우클릭하여 "Create HDL Wrapper"를 실행합니다.
- (PL 로직이 있다면) "Generate Bitstream"을 실행합니다.
- 메인 메뉴에서 *****File -> "Export" -> "Export Hardware"*****를 선택하여 XSA 파일을 생성합니다.

이 과정에서 사용자가 PS의 내부 레지스터나 핀 설정을 직접 건드릴 필요가 거의 없습니다.

추가 정보가 필요한 예외 경우

물론, 다음과 같은 경우에는 보드 파일만으로는 부족하며 사용자의 추가 설정이 필요합니다.

- **커스텀 보드(Custom Board):** 직접 설계한 보드처럼 공식 보드 파일이 없는 경우에는 모든 PS 설정을 데이터시트를 참고하여 수동으로 직접 입력해야 합니다.
- **보드 개조:** 기존 개발 보드의 클럭 소스를 바꾸거나, 특정 페리페럴의 연결을 수정하는 등 하드웨어를 변경한 경우, 보드 파일의 기본 설정값을 수동으로 수정해야 합니다.
- **특수 목적 설정:** 기본 보드 설정과 다른 특별한 DDR 동작 속도나 클럭 주파수가 필요한 경우, Block Automation 실행 후에 PS 설정 창을 열어 원하는 값으로 직접 변경해야 합니다.

결론

표준 개발 보드를 그대로 사용한다면, 제조사가 제공하는 보드 파일만으로도 **PS**의 기본 설정을 모두 완료하고 **XSA** 파일을 생성하는 것이 충분히 가능합니다. 이는 **AMD SoC** 기반 개발의 복잡성을 크게 줄여주는 매우 편리한 기능입니다.

PL(Programmable Logic)에 로직을 추가할 경우, 보드 파일 외에 반드시 사용자가 직접 제공해야 하는 핵심적인 정보들이 있습니다.

이전 답변에서 보드 파일은 **PS(Processing System)**를 위한 자동 설정 가이드라고 설명했습니다. 이제 PL을 추가한다는 것은 그 위에 사용자만의 맞춤형 회로를 구축하는 것과 같으며, 이 회로에 대한 모든 정보는 사용자가 직접 만들어야 합니다.

PL 로직 추가 시 보드 파일 외에 필요한 핵심 요소는 다음과 같습니다.

1. RTL 소스 파일 (HDL Code: Verilog/VHDL)

- 이것이 무엇인가요?
PL에 구현하고 싶은 디지털 회로의 동작과 구조를 설명하는 코드입니다. AND 게이트, 카운터, 상태 머신(State Machine)부터 복잡한 신호 처리 알고리즘까지 모든 로직이 여기에 해당합니다.
- 왜 필요한가요?
Vivado는 이 코드를 읽어서 실제 회로(Netlist)로 변환(합성, Synthesis)합니다. 즉, PL 설계의 '설계도' 원본입니다.
- 예시: led_blinker.v (Verilog), my_filter.vhd (VHDL)

2. 제약 파일 (Constraints File - XDC)

- 이것이 무엇인가요?
RTL 코드라는 논리적 설계를 실제 칩의 물리적 현실에 매핑하는 규칙 파일입니다.
- 왜 필요한가요?
XDC 파일 없이는 Vivado가 설계한 회로를 어떻게 칩 외부의 핀과 연결하고, 얼마나 빠른 속도로 동작시켜야 할지 알 수 없습니다.
- 주요 내용:
 - 핀 제약 (**Pin Constraints**): 설계의 입출력 포트(예: led_output)를 FPGA 칩의 실제 핀(예: AF12)에 할당합니다. 보드 파일이 보드에 있는 기본 LED, 스위치 등의 핀 정보를 제공해주기도 하지만, 사용자가 추가한 PL 로직의 입출력은 직접 지정해야 합니다.
 - 타이밍 제약 (**Timing Constraints**): 설계가 동작해야 하는 클럭 속도(예: "100MHz 클럭 생성")를 정의하고, 입출력 신호의 타이밍 요구사항을 지정합니다. 이는 설계가 원하는 속도로 안정적으로 동작하기 위해 필수적입니다.
- 예시: constraints.xdc

3. PL-PS 인터페이스 설정 (in Block Design)

- 이것이 무엇인가요?
사용자가 만든 PL 로직이 PS(프로세서)와 데이터를 주고받거나 제어를 받아야 할 때, 둘 사이의 연결을 정의하는 작업입니다.
- 왜 필요한가요?
PL이 독립적으로만 동작하는 것이 아니라, 프로세서의 제어를 받거나(예: 레지스터 설정), PL이 처리한 데이터를 프로세서가 가져가야 하는(예: AXI 인터페이스를 통해 DDR 메모리에 접근) 경우가 많습니다. 이 '소통 경로'를 만들어야 합니다.
- 주요 작업:
 - Vivado Block Design에서 Zynq/Versal IP를 열고, **AXI (Advanced eXtensible Interface)** 포트와 같은 PS-PL 인터페이스를 활성화합니다.
 - 활성화된 AXI 인터페이스를 사용자가 만든 PL 로직(보통 IP로 패키징)에 연결합니다.
 - 메모리 맵 주소(Address Map)를 할당하여 프로세서가 PL의 특정 레지스터에 접근할 수 있도록 설정합니다.

요약 비교

구분	보드 파일 (Board File)의 역할	사용자가 추가해야 하는 것 (PL 설계)
목적	하드웨어 플랫폼(PS) 자동 구성	사용자 정의 기능(PL) 구현
주요 내용	DDR, 클럭, MIO 등 PS 관련 설정 정보	1. RTL 소스 파일 (.v, .vhd): 회로의 동작 정의
		2. 제약 파일 (.xdc): 핀 매핑 및 타이밍 정의
		3. PL-PS 인터페이스: 프로세서와의 통신 경로 설정
비유	잘 지어진 아파트 건물	그 안에 들어가는 가구와 인테리어

결론

보드 파일은 **PS**라는 튼튼한 기반을 자동으로 마련해주는 역할**까지입니다. 그 위에 **PL**이라는 공간에 어떤 기능의 회로를, 어떤 핀에 연결하여, 얼마나 빠른 속도로 동작시킬 것인지는 전적으로 개발자의 몫이며, 이를 위해 **RTL 코드**, **XDC 제약 파일**, 그리고 **PL-PS** 인터페이스 설정이 반드시 필요합니다.