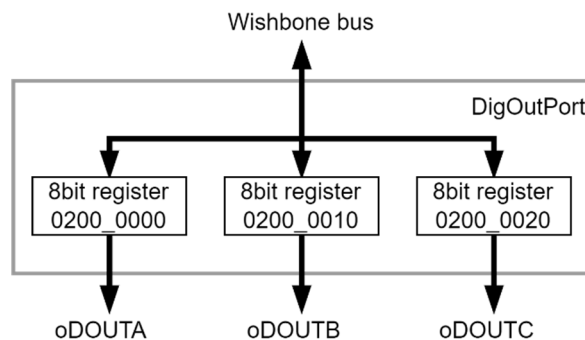


Lab 1. Digital Port 및 레지스터 인터페이스

제공된 Vivado Project를 활용하여 FPGA 내부에 Digital Port 및 레지스터를 생성하고 인터페이스한다.

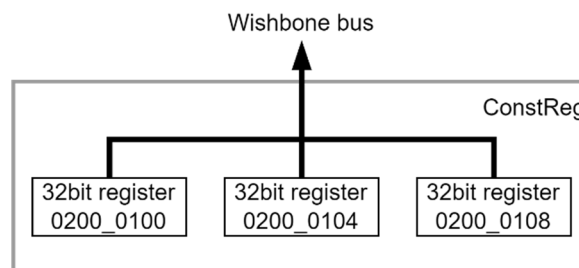
실습 1. [따라하기]FPGA 내부에 Digital Output Port 생성 및 버스 인터페이스 (WR/RD transaction)

1. 디지털 출력 단자를 생성하고, 이를 Wishbone bus에 인터페이스하여 버스 마스터에서 이 출력 단자 각각의 상태를 ON/OFF로 제어하려고 함.
2. 디지털 출력 포트의 주소는 32'h0200_0000으로 할당하고, 32bit register 중 하위 8개 bit를 디지털 출력 단자(oDOUTA[7:0])로 출력.
3. 문제를 단순화하기 위하여 asynchronous slave 방식 적용. (Bus transaction에 1 cycle이 소요)
4. reg 타입의 변수를 8 bit로 정의하여 8 bit register를 하나 만들고, 그 register 값을 8개 디지털 출력 단자에 연결. 이제 이 register 값을 변경하면 출력 단자의 상태가 변경됨.
5. STB와 ADR이 맞으면 digital output port 모듈이 선택된 것이며, WE가 1이면 clock의 posedge에 register 값을 DAT 버스 값으로 변경. WE가 0이면 DAT 출력 버스에 register 값을 연결.
6. 단, 초기 상태를 우리가 원하는 값으로 설정하기 위하여, clock의 posedge에서 RST인 경우에 우선 순위를 높여서 초기값을 설정하도록 always 구문을 보완.
7. [연습과제] oDOUTB[7:0]과 oDOUTC[7:0]도 각각 32'h0200_0010과 32'h0200_0020의 주소로 접근하여 단자의 상태를 변경할 수 있도록 수정하고, testbench 코드를 수정하여 동작 확인. 하나의 IP 블록에서 oDOUTA, oDOUTB, oDOUTC가 출력되도록 코딩.

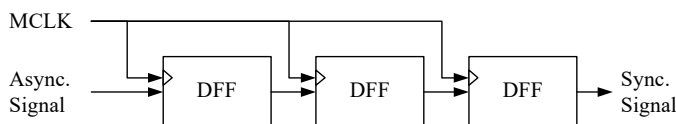


실습 2. [연습과제]FPGA 내부에 저장된 상수 값을 read하는 기능 구현 (RD transaction)

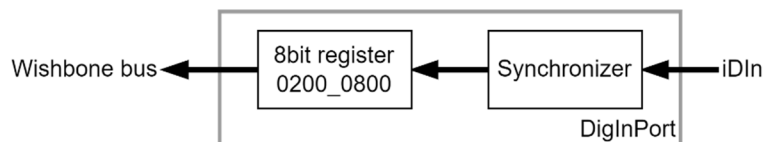
1. 많은 디지털 프로세서의 경우 제공되는 레지스터 중 한 개를 magic number 또는 ID, Version 등을 기억하는 상수 레지스터로 제공. 이러한 기능을 제공할 목적으로 ConstReg IP를 생성.
2. 상수 값을 읽기 위한 주소로 32'h0200_0100을 할당하고 ID_VER 상수 용도로 할당. 상수 값을 32'h0123_4567의 32 bit로 설정.
3. 앞의 실습 1에서 출력 디지털 단자는 삭제하고, 상수값을 읽는 기능만 남기도록 수정하면 ConstReg IP를 쉽게 구성할 수 있음. Testbench 코드를 수정하여 동작 확인. Testbench 코드에는 실습1의 IP는 삭제하고 ConstReg IP만을 테스트.
4. 32'h0200_0104와 32'h0200_0108에는 임의의 다른 magic number를 저장하고, RD transaction으로 읽어갈 수 있도록 IP를 수정. Test bench 코드를 수정하여 동작 확인. 하나의 IP 블록에서 3개의 주소에 3개의 magic number를 저장하도록 수정.

**실습 3. [연습과제]FPGA 외부 신호 입력을 위한 Digital Input Port 생성 및 버스 인터페이스 (RD transaction)**

1. FPGA 외부에서 입력되는 디지털 신호를 입력 받아 그 상태를 마스터에 전달하기 위한 digital input port를 생성하고 버스에 인터페이스 함. 입력 신호는 iDIn[7:0]의 8bit 사용하며, 주소는 32'h0200_0800으로 설정. IP 이름은 DigInPort 로 설정.
2. 외부의 디지털 신호는 입력 초단에 무조건 동기화기(synchronizer)를 적용하여 metastability의 위험성을 제거하여야 함. 2~3개 정도의 DFF에 순차적으로 적용하여 2~3개 클럭 사이클 지연된 신호를 사용.



3. 동기화기 출력 신호 상태를 위에 할당한 주소의 RD transaction에 마스터에 전달.
4. DigInPort IP를 구현하고, testbench를 수정하여 동작을 확인. 여기서도 DigInPort IP만을 testbench에 남겨 테스트.

**실습 4. [연습과제]위 3개 IP 모듈을 testbench에 모두 생성하여 동작 확인**

1. 지금까지는 DigOutPort, ConstReg, 그리고 DigInPort를 각각 testbench에 객체 생성하여 동작을 확인 함. 여기서는 3개의 IP 모듈을 testbench에 모두 객체를 생성하여 동작을 확인.
2. 마스터에 ACK 신호를 전달하는 방법을 생각해보아야 함.