

Direct Memory Access using CDMA

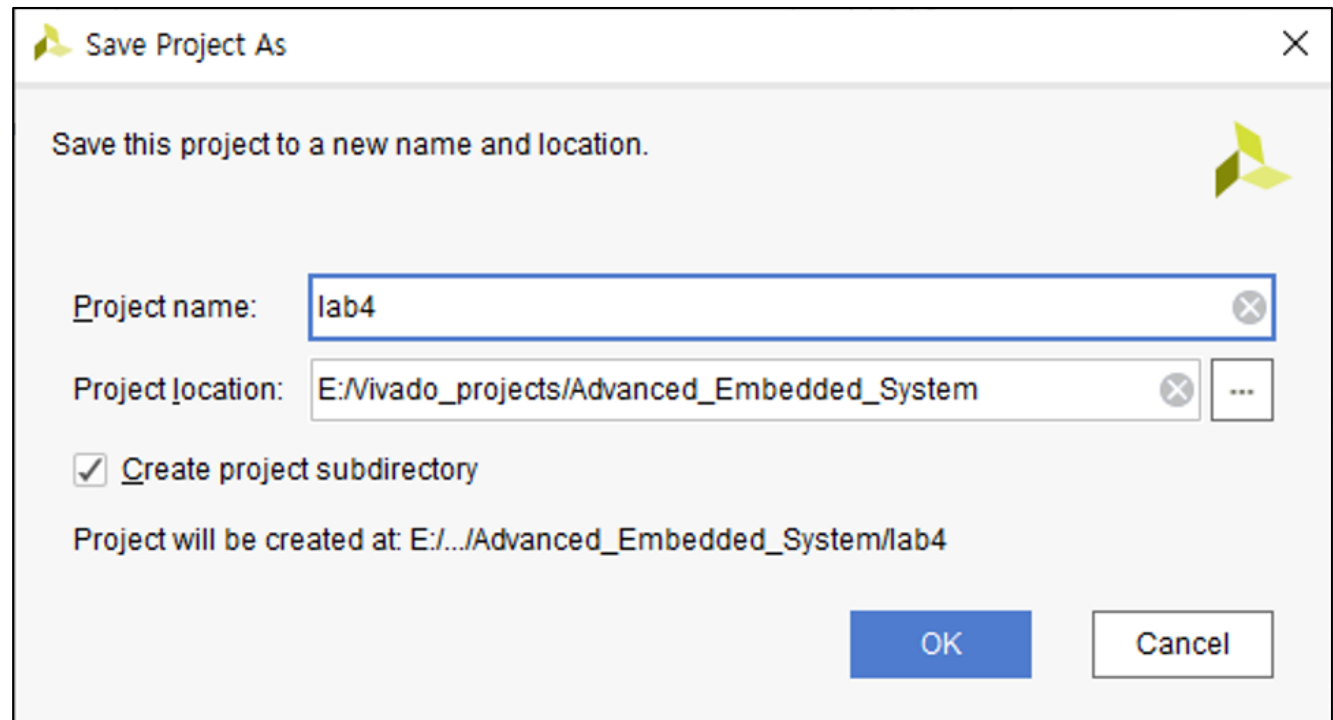
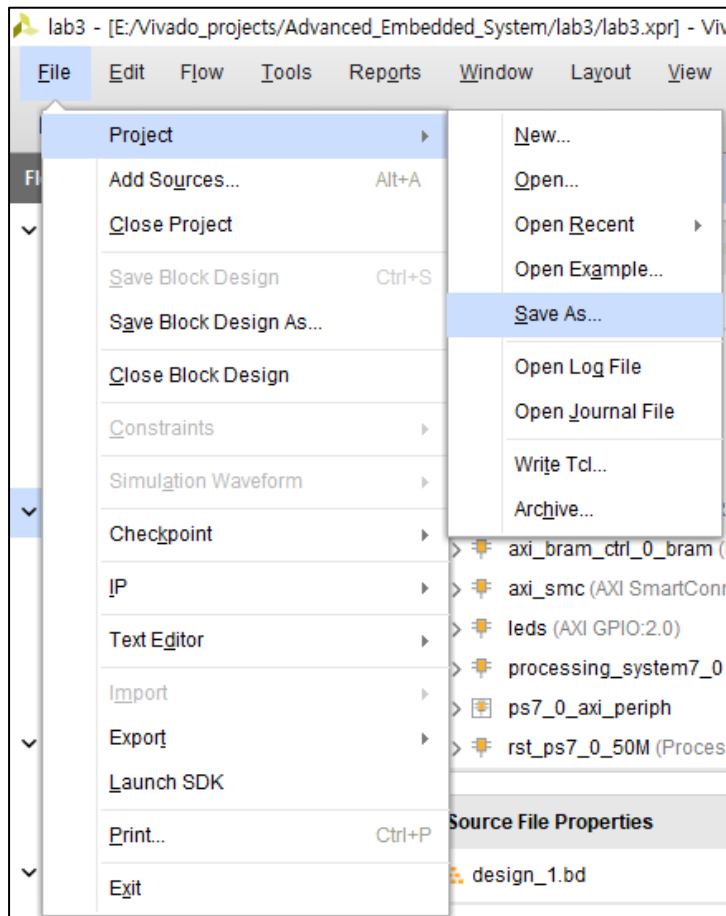
(lab4)

24th February 2023

1. lab4 프로젝트 생성하기

> lab3 프로젝트를 lab4 프로젝트로 복사하기

- 기존 lab3 프로젝트를 열어서 Project > Save As 클릭 -> "lab4"로 이름변경 후 저장



2. S_AXI_HP0를 Enable하기

> ZYNQ7 PS 커스터마이징

- ZYNQ7 Processing System IP 블록을 더블 클릭하여 커스텀 진행
- PS-PL Configuration에서 HP Slave AXI Interface > **S AXI HP0 Interface**를 체크

Page Navigator

Zynq Block Design

PS-PL Configuration

Peripheral I/O Pins

MIO Configuration

Clock Configuration

DDR Configuration

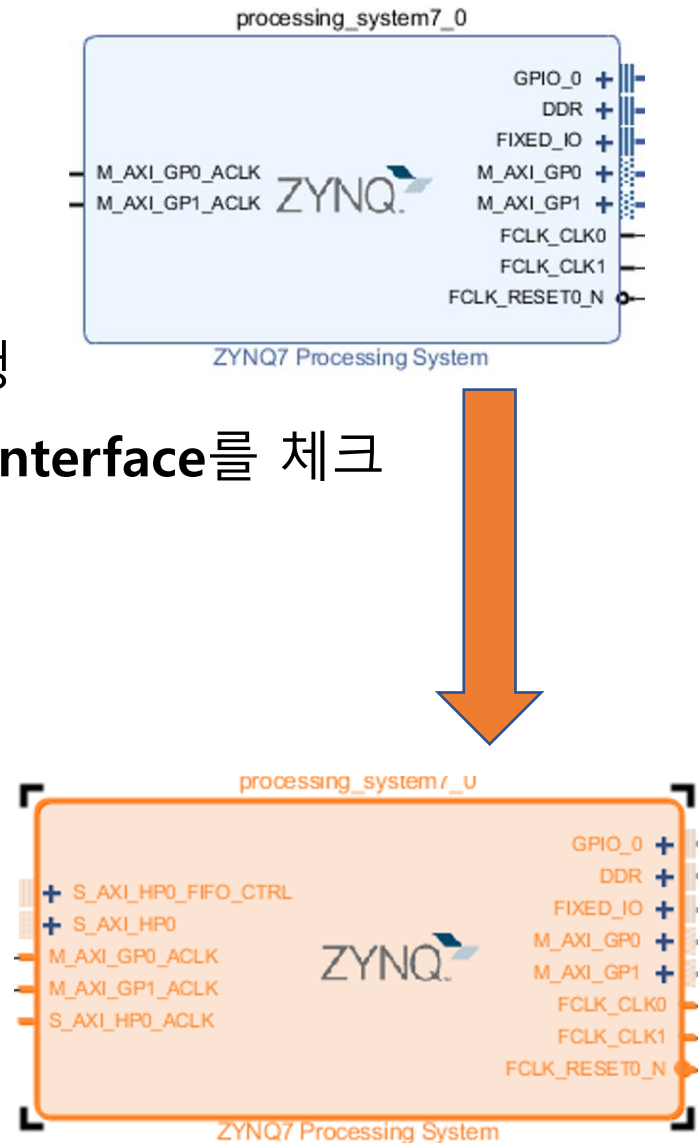
SMC Timing Calculation

Interrupts

PS-PL Configuration

Search: Q

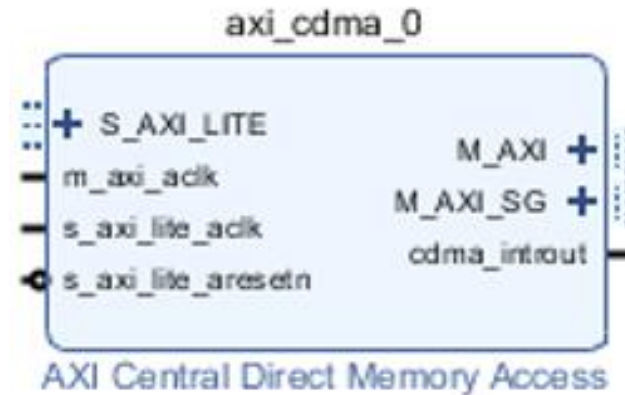
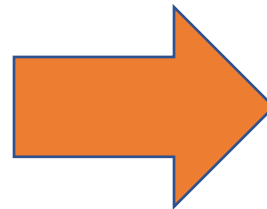
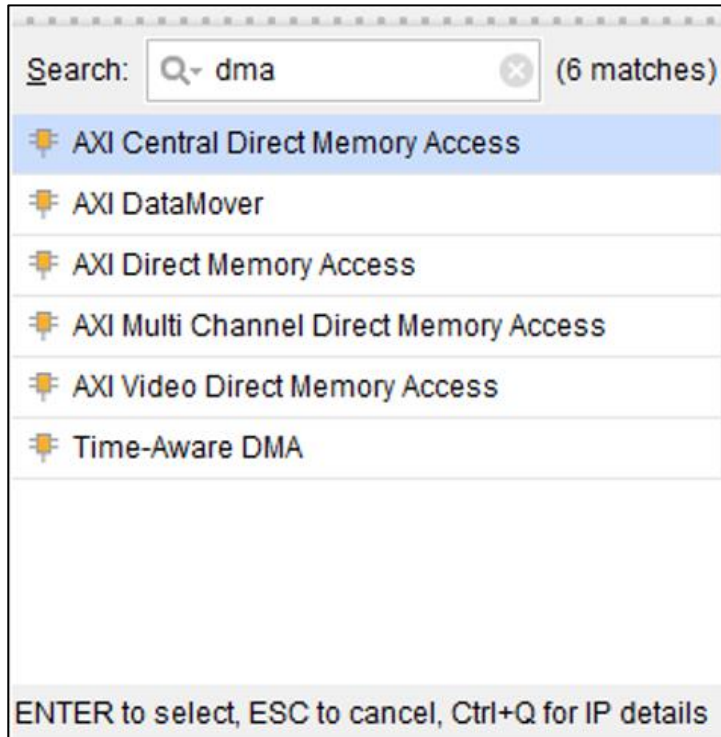
Name	Select	Description
> General		
> AXI Non Secure Enablement	0	Enable AXI Non Secure Transaction
> GP Slave AXI Interface		
> HP Slave AXI Interface		
> S AXI HP0 interface	<input checked="" type="checkbox"/>	Enables AXI high performance slave interface 0
> S AXI HP1 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 1
> S AXI HP2 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 2
> S AXI HP3 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 3
> ACP Slave AXI Interface		
> DMA Controller		
> PS-PL Cross Trigger interface	<input type="checkbox"/>	Enables PL cross trigger signals to PS and vice-versa



3. AXI central DMA controller (CDMA)추가하기

> CDMA IP블록 추가하기

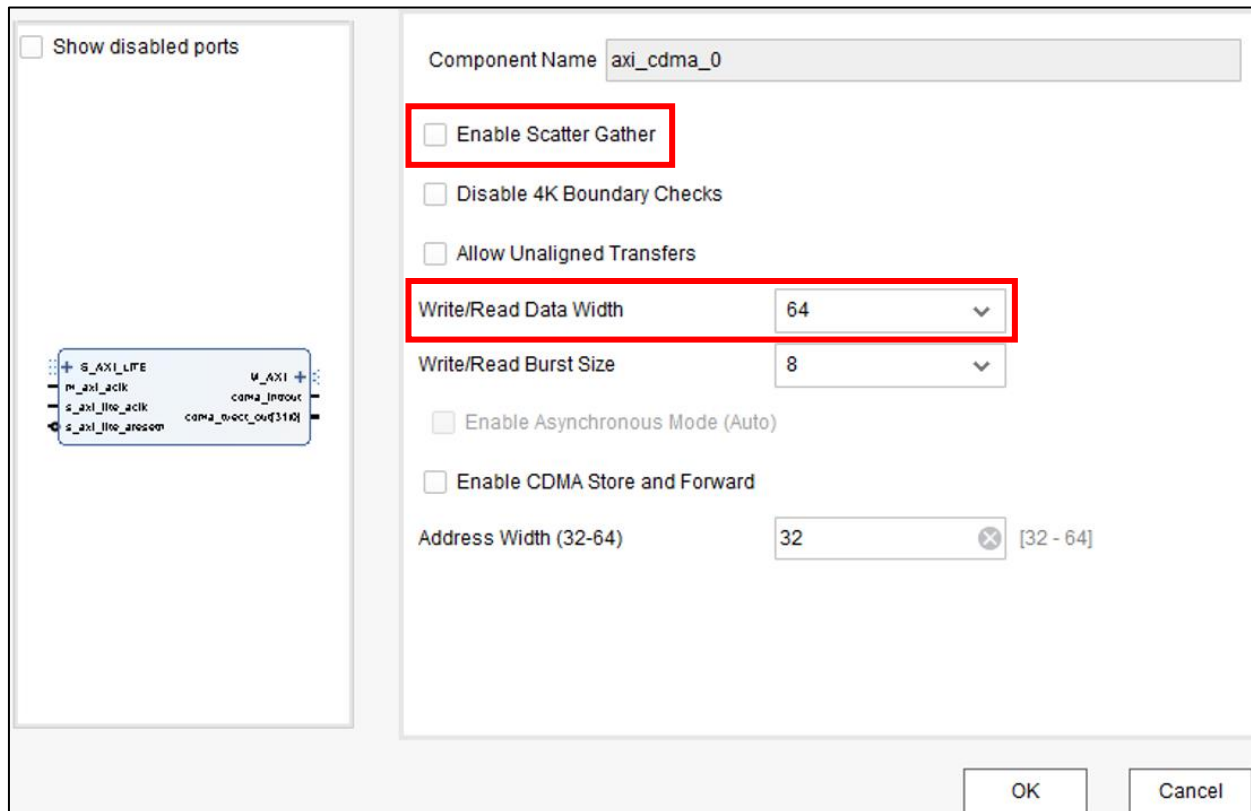
- Add IP에서 **AXI Central Direct Memory Access** 블록 추가



3. AXI central DMA controller (CDMA)추가하기

> CDMA IP블록 커스터마이징

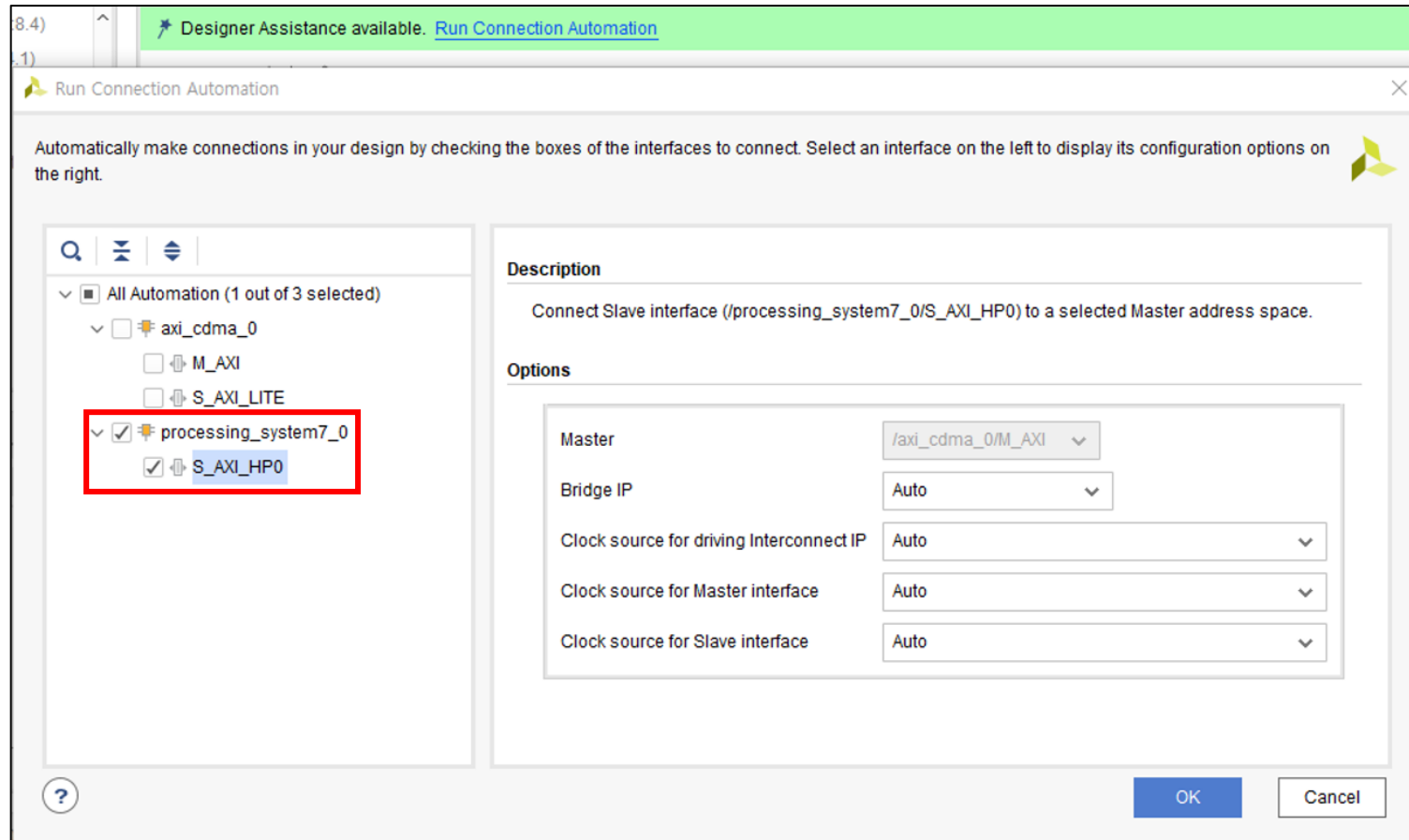
- 추가된 블록을 더블 클릭하여 커스텀 진행
- **Enable Scatter Gather 체크 해제**, Write/Read Data Width = **64**로 수정 -> OK 클릭



3. AXI central DMA controller (CDMA)추가하기

> Run Connection Automation 진행

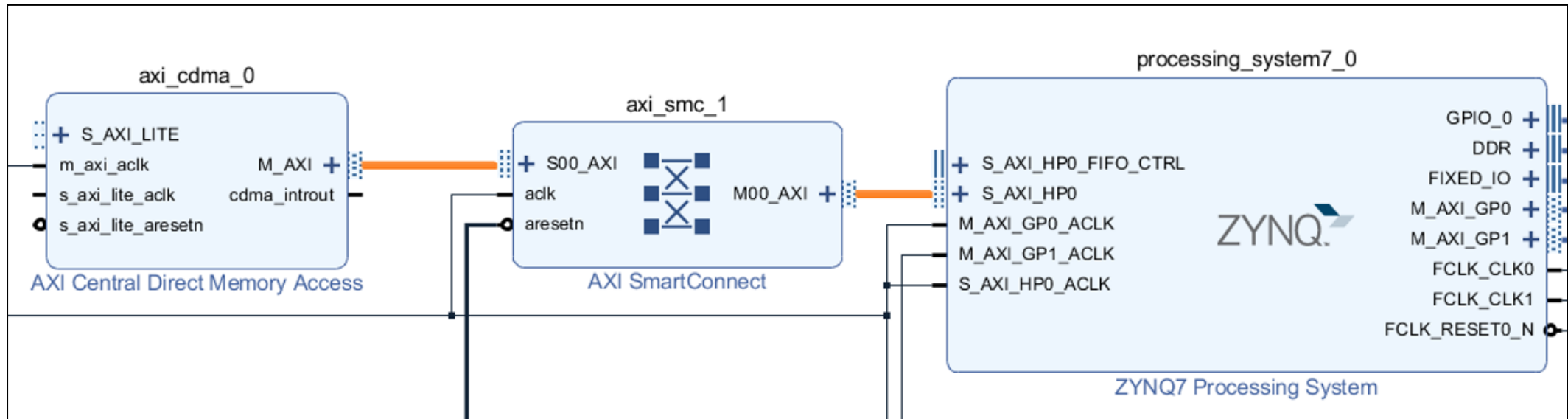
- processing_system7_0 > **S_AXI_HP0** 만 Connection 진행



3. AXI central DMA controller (CDMA)추가하기

> Run Connection Automation 진행

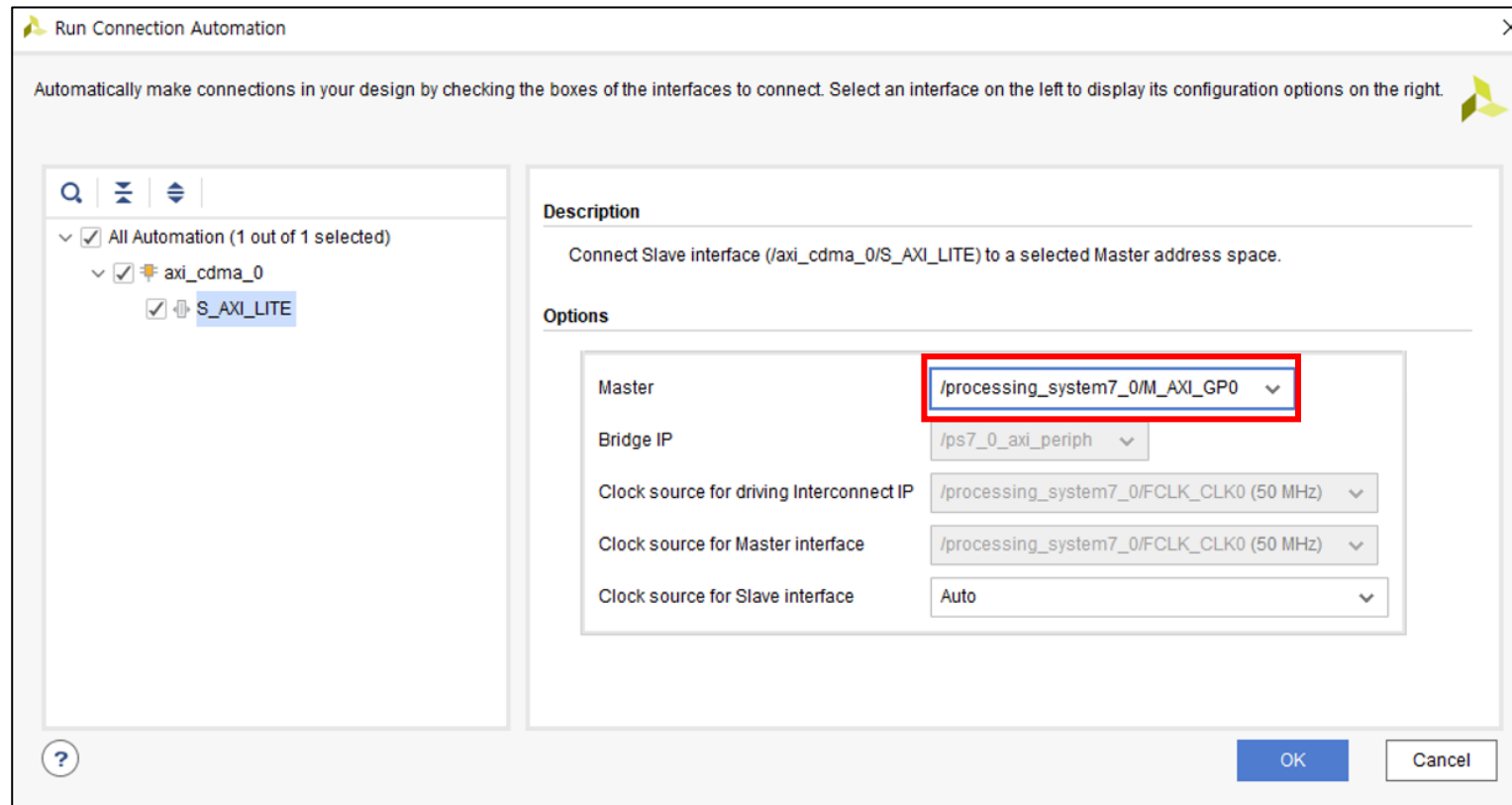
- processing_system7_0 > **S_AXI_HP0** 만 Connection 진행
- 연결 구조가 다음과 같이 이루어졌는지 확인하기



3. AXI central DMA controller (CDMA)추가하기

> Run Connection Automation 진행

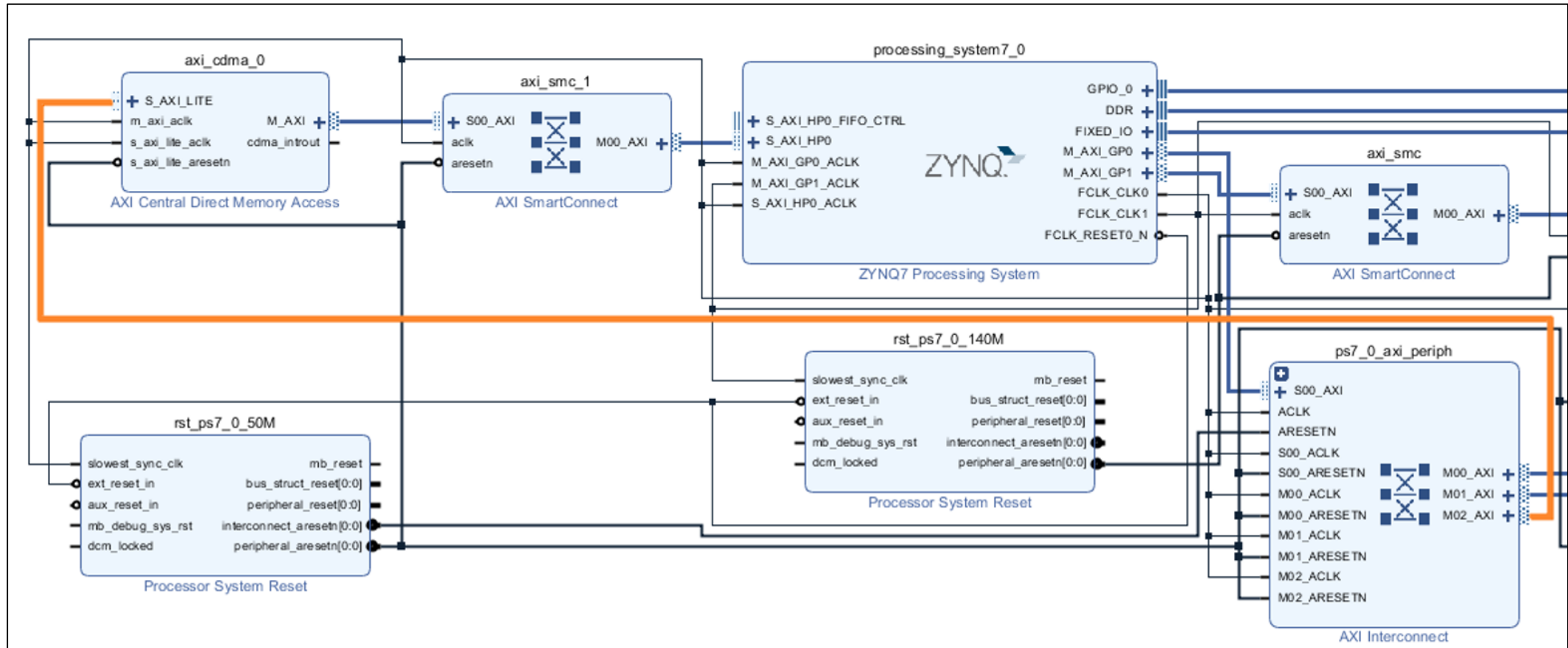
- 다시 한 번 Run Connection Automation 클릭
- axi_cdma_0 > **S_AXI_LITE** 연결 option에서 설정을 그림과 같이 변경



3. AXI central DMA controller (CDMA)추가하기

> Run Connection Automation 진행

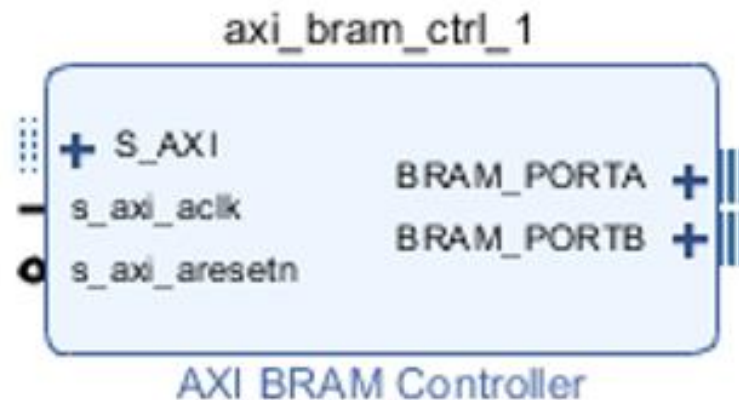
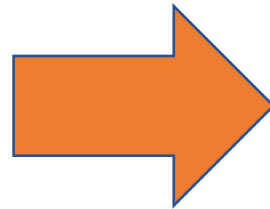
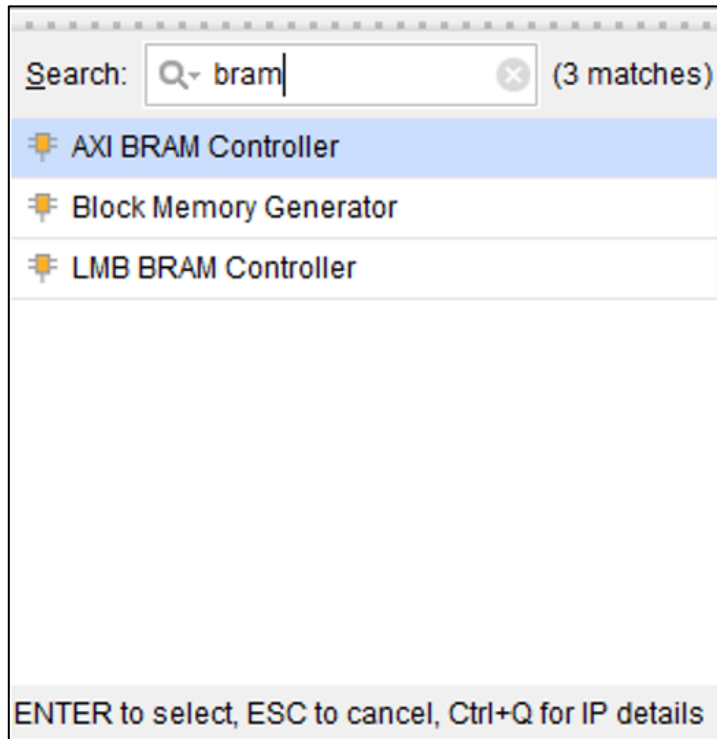
- 그림과 같이 연결되었는지 확인하기



4. 또 다른 BRAM 추가하기

> AXI BRAM Controller 추가

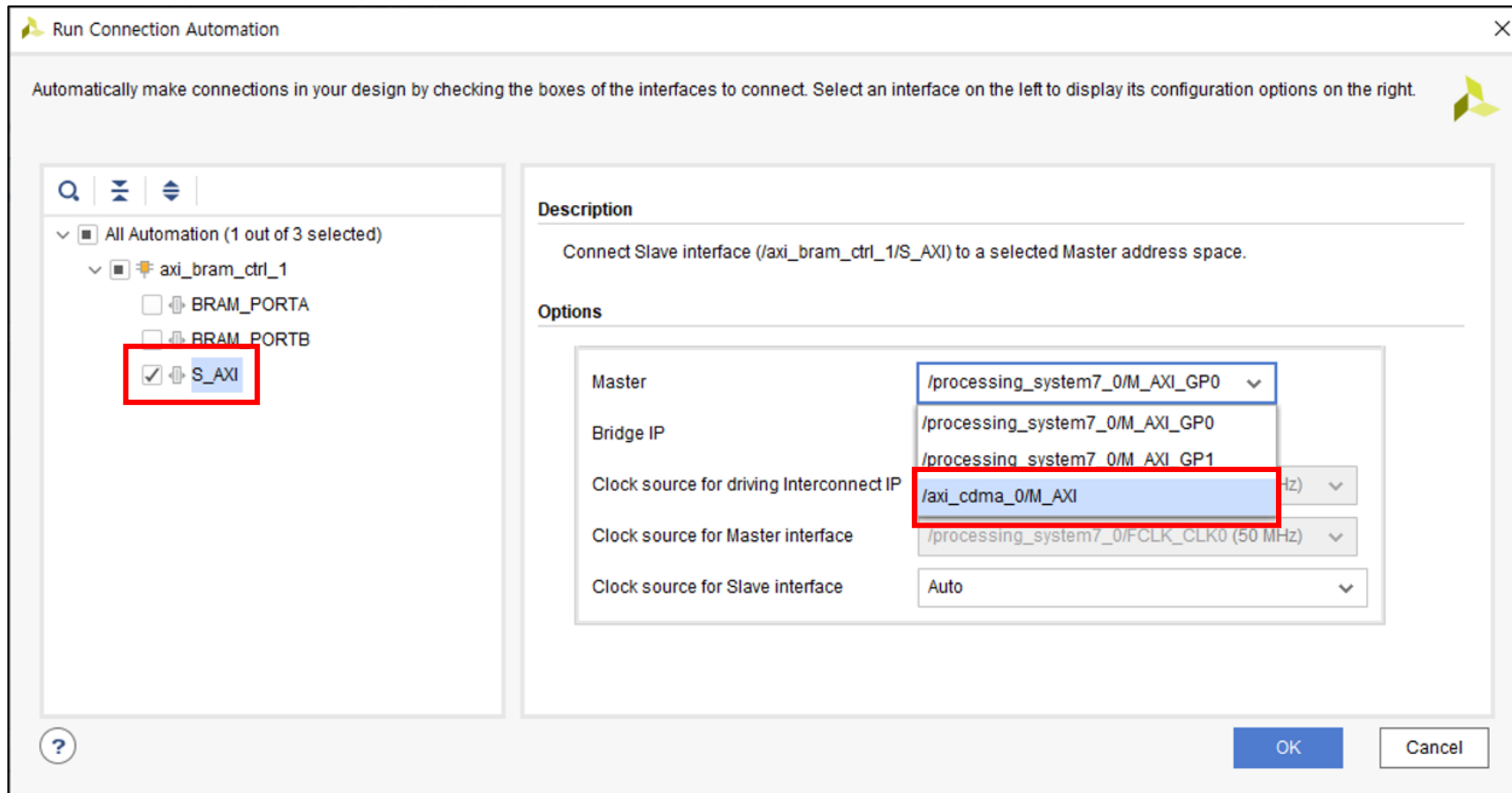
- Add IP에서 AXI BRAM Controller 블록 추가
- axi_bram_ctrl_0이 이미 있으므로(lab3) axi_bram_ctrl_1 블록이 추가됨



4. 또 다른 BRAM 추가하기

> AXI BRAM Controller 에 대해 Run Connection Automation 진행

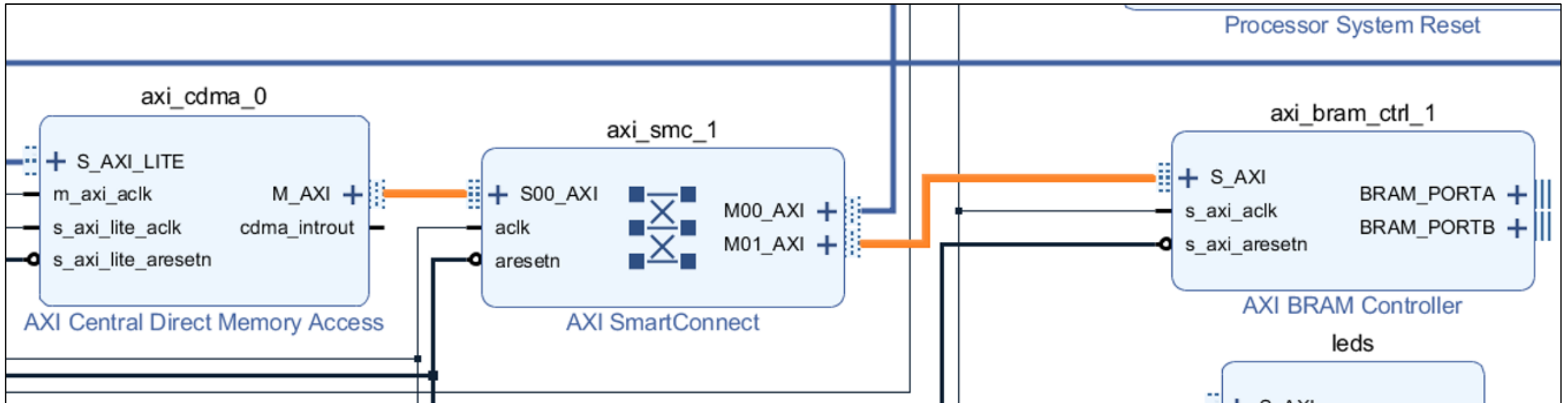
- S_AXI 만 선택하고 옵션에서 Master = /axi_cdma_0/M_AXI 로 변경 -> OK 클릭



4. 또 다른 BRAM 추가하기

> AXI BRAM Controller 에 대해 Run Connection Automation 진행

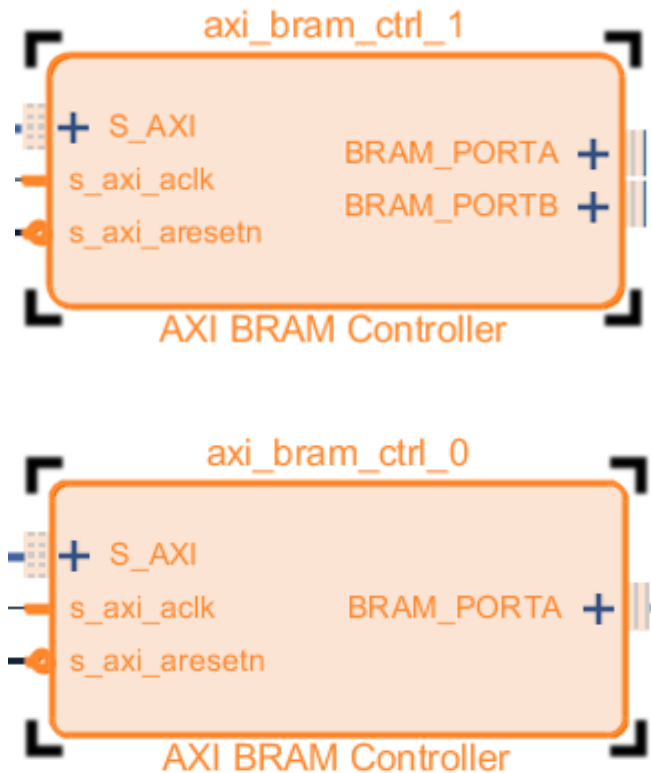
- 다음과 같이 연결되었는지 확인하기



4. 또 다른 BRAM 추가하기

> AXI BRAM Controller 커스텀 진행 (2개 블록에 대해 모두 수행)

- Data Width = 64 / Number of BRAM interface = 1 로 수정



The screenshot shows the configuration window for the component `axi_bram_ctrl_1`. The window is divided into two main sections: a left pane showing a block diagram of the component and a right pane showing configuration options.

Left Pane: Shows a block diagram of the `axi_bram_ctrl_1` component. It has inputs `S_AXI`, `s_axi_aclk`, `s_axi_aresetn`, and `BRAM_PORTA`.

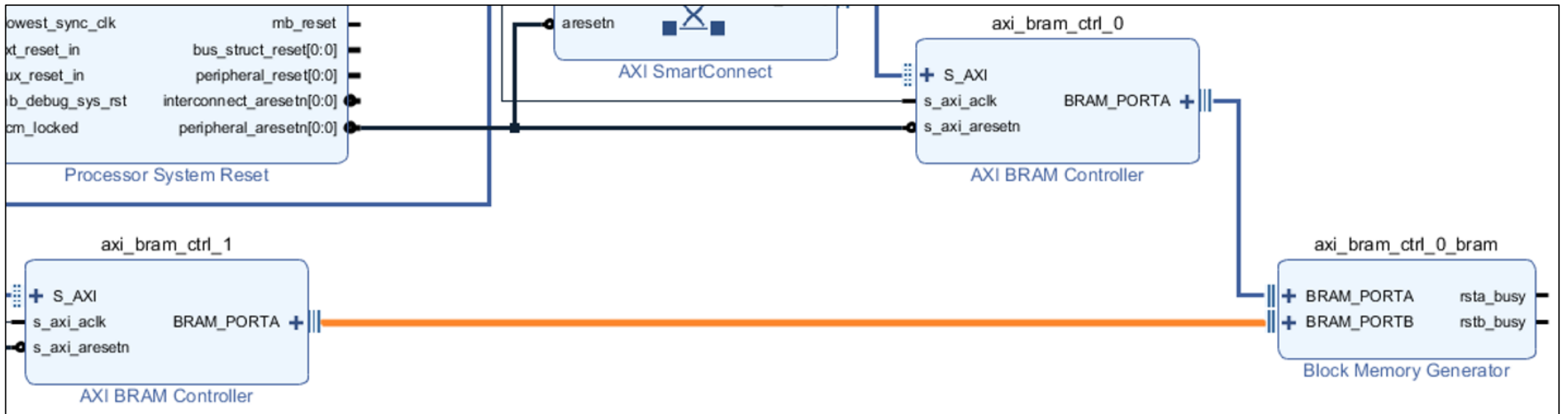
Right Pane: Contains configuration options for the component.

- Component Name:** `axi_bram_ctrl_1`
- AXI Protocol:** `AXI4`
- Data Width:** `64` (highlighted with a red box)
- Memory Depth (Auto):** `8192`
- ID Width (Auto):** `0`
- Support AXI Narrow Bursts:** `Yes`
- BRAM Options:**
 - BRAM_INSTANCE (Auto):** `External`
 - Number of BRAM interfaces:** `1` (highlighted with a red box)
- ECC Options:**
 - Enable ECC:** `No`
 - ECC TYPE:** `Hamming`
 - Enable Fault Injection:** `No`
 - ECC Reset Value:** `0`

Buttons at the bottom right: `OK` and `Cancel`.

4. 또 다른 BRAM 추가하기

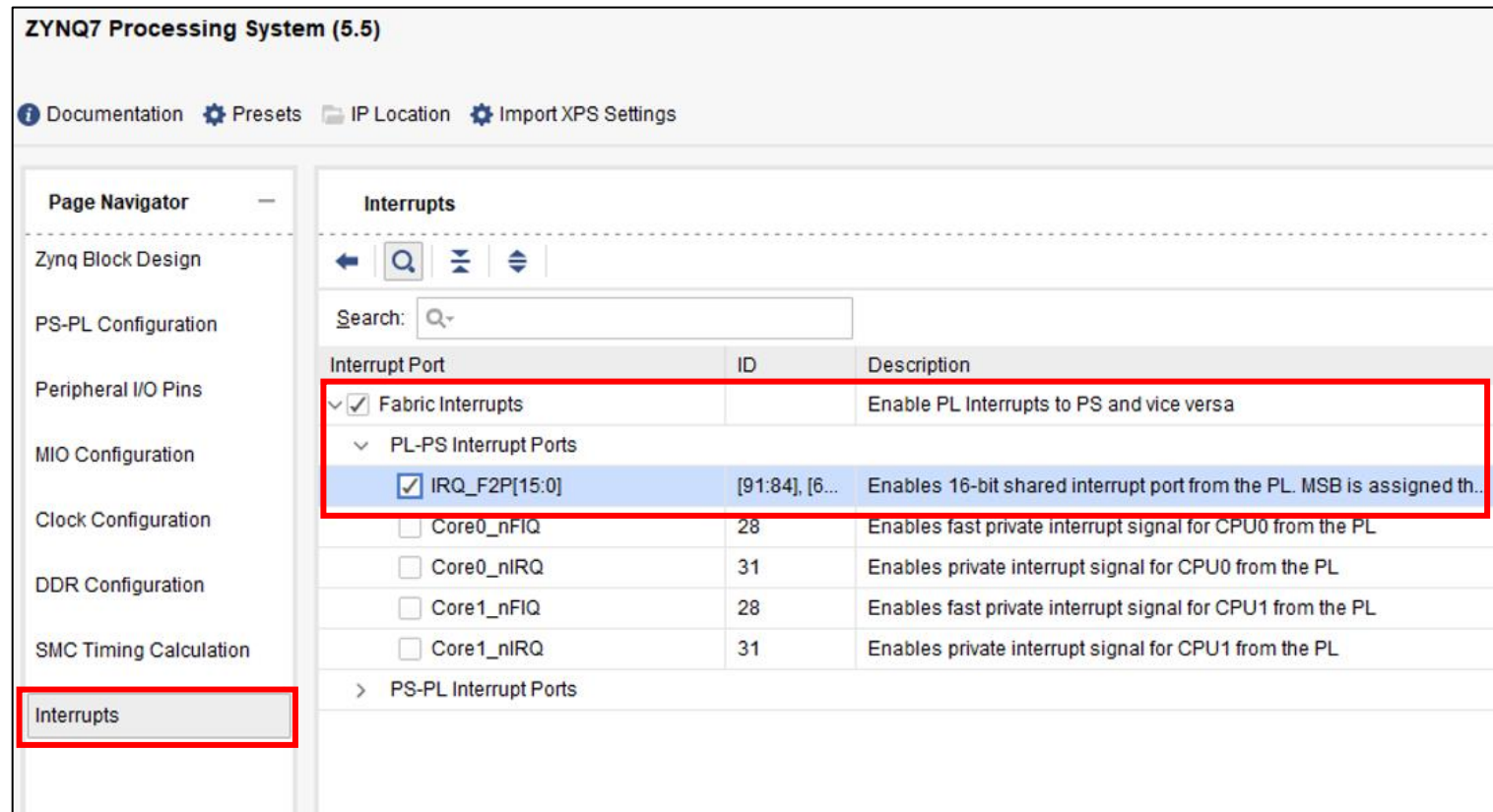
- > 두 AXI BRAM Controller를 Block Memory Generator에 연결
- 아래 그림과 같이 wire를 연결하기



5. Interrupt 추가하기

> ZYNQ7 PS 블록에서 interrupt 기능 추가

- 블록을 더블 클릭하여 생기는 커스텀 창에서 Interrupts 탭 클릭
- Fabric Interrupts > PL-PS Interrupt Ports > IRQ_F2P 체크



ZYNQ7 Processing System (5.5)

Documentation Presets IP Location Import XPS Settings

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts**

Interrupts

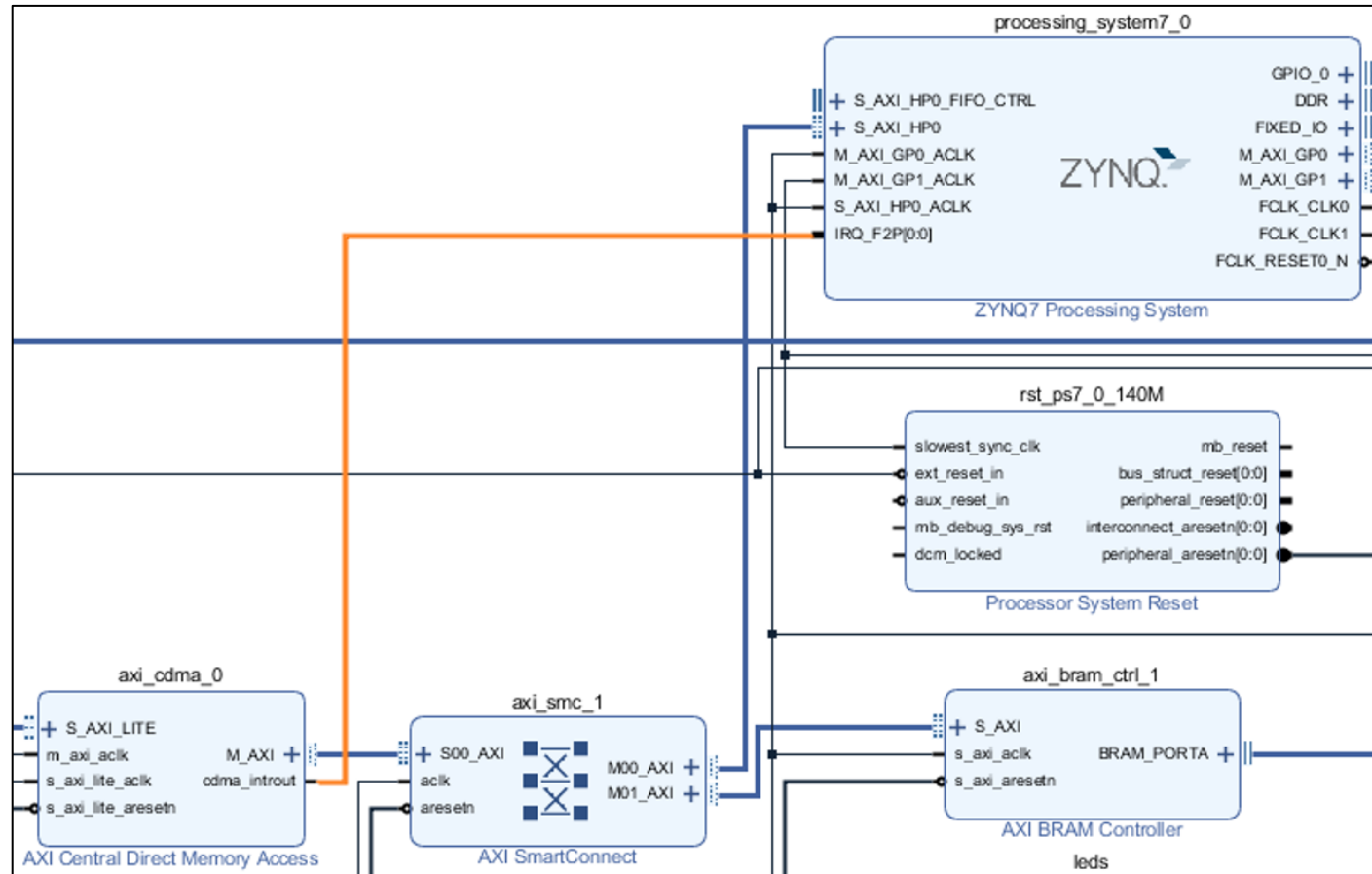
Search:

Interrupt Port	ID	Description
<input checked="" type="checkbox"/> Fabric Interrupts		Enable PL Interrupts to PS and vice versa
▼ PL-PS Interrupt Ports		
<input checked="" type="checkbox"/> IRQ_F2P[15:0]	[91:84], [6...	Enables 16-bit shared interrupt port from the PL. MSB is assigned th...
<input type="checkbox"/> Core0_nFIQ	28	Enables fast private interrupt signal for CPU0 from the PL
<input type="checkbox"/> Core0_nIRQ	31	Enables private interrupt signal for CPU0 from the PL
<input type="checkbox"/> Core1_nFIQ	28	Enables fast private interrupt signal for CPU1 from the PL
<input type="checkbox"/> Core1_nIRQ	31	Enables private interrupt signal for CPU1 from the PL
> PS-PL Interrupt Ports		

5. Interrupt 추가하기

> ZYNQ7 PS 블록에서 interrupt 기능 추가

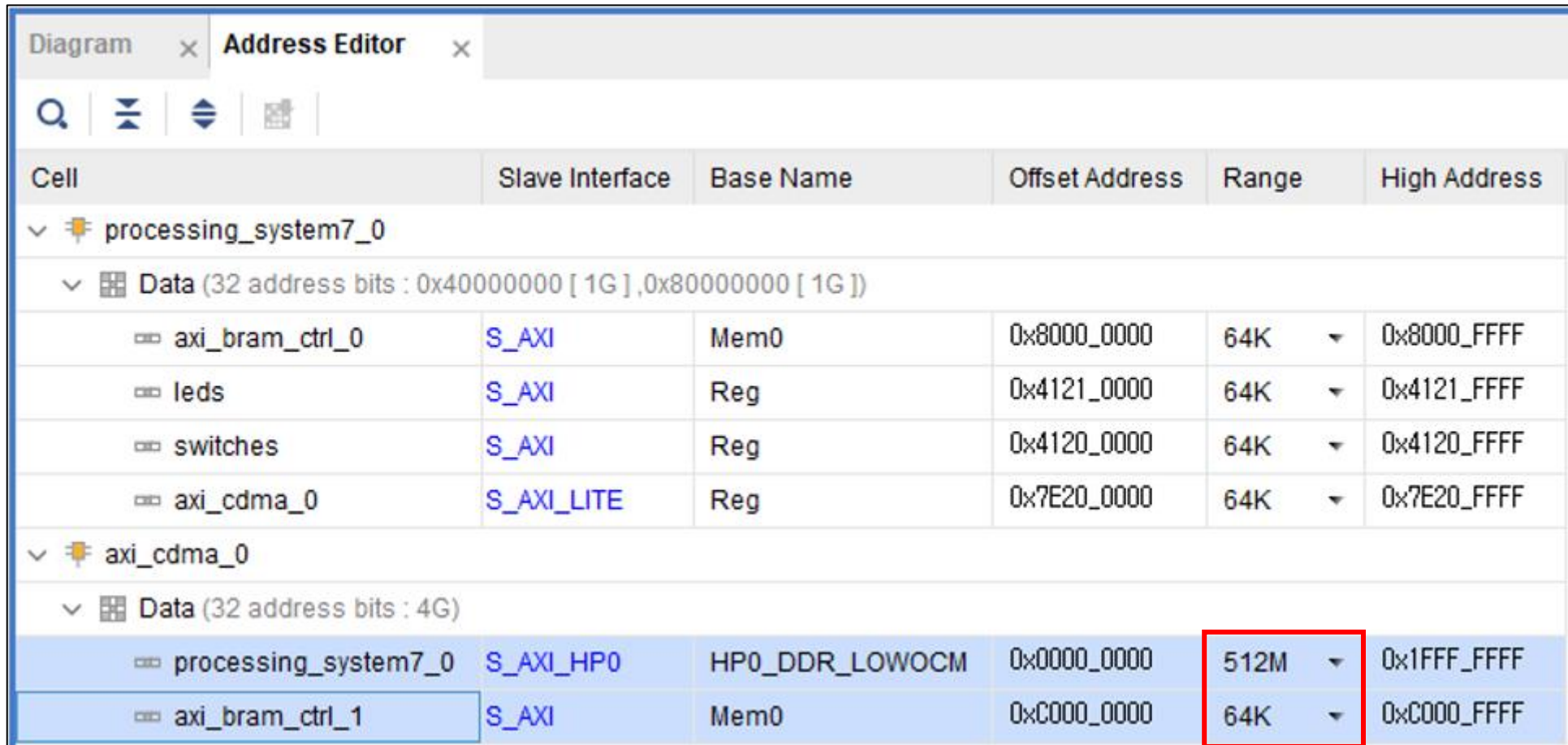
- ZYNQ7 PS의 **IRQ_F2P** 핀과 axi_cdma_0의 **cdma_introut** 핀을 wire로 연결



6. Address Editor 수정하기

> Address Editor 탭에서 다음과 같이 Range 값을 수정

- processing_system7_0 -> **512M** / axi_bram_ctrl_1 -> **64K** 로 수정

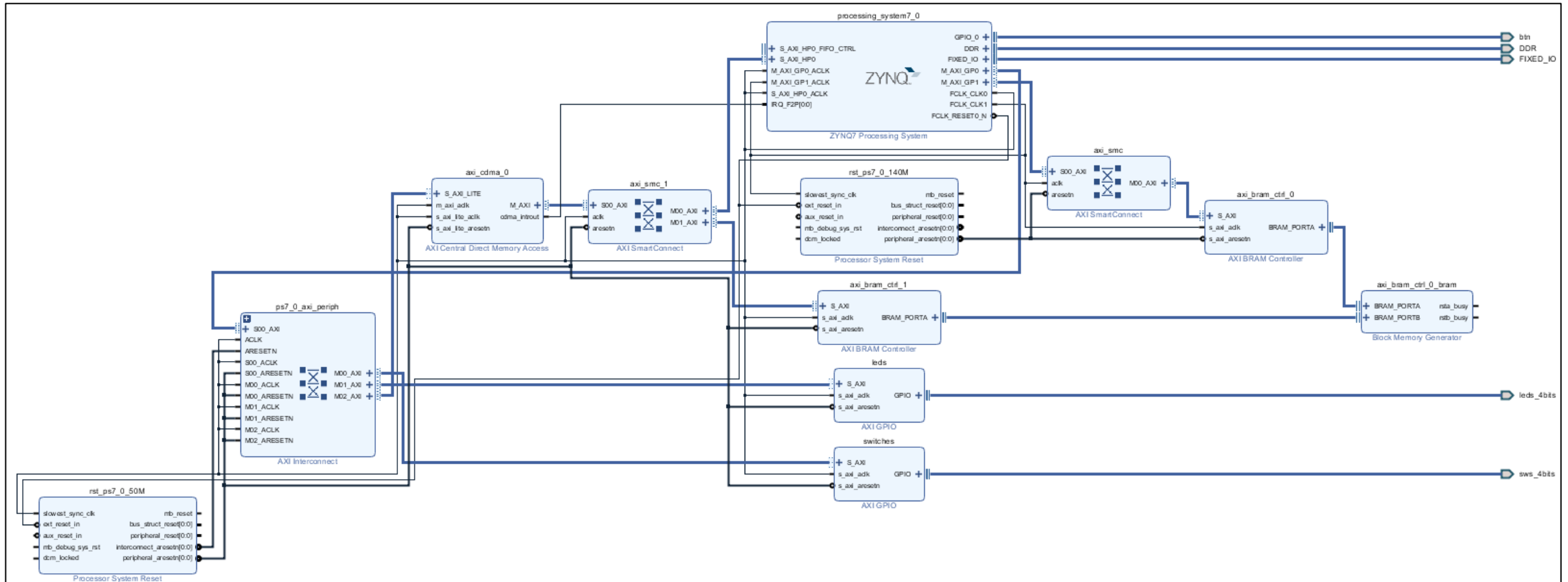


Cell	Slave Interface	Base Name	Offset Address	Range	High Address
▼ processing_system7_0					
▼ Data (32 address bits : 0x40000000 [1G], 0x80000000 [1G])					
axi_bram_ctrl_0	S_AXI	Mem0	0x8000_0000	64K ▼	0x8000_FFFF
leds	S_AXI	Reg	0x4121_0000	64K ▼	0x4121_FFFF
switches	S_AXI	Reg	0x4120_0000	64K ▼	0x4120_FFFF
axi_cdma_0	S_AXI_LITE	Reg	0x7E20_0000	64K ▼	0x7E20_FFFF
▼ axi_cdma_0					
▼ Data (32 address bits : 4G)					
processing_system7_0	S_AXI_HP0	HP0_DDR_LOWOCM	0x0000_0000	512M ▼	0x1FFF_FFFF
axi_bram_ctrl_1	S_AXI	Mem0	0xC000_0000	64K ▼	0xC000_FFFF

7. Bitstream 생성 및 SDK 실행

> Bitstream 생성하기

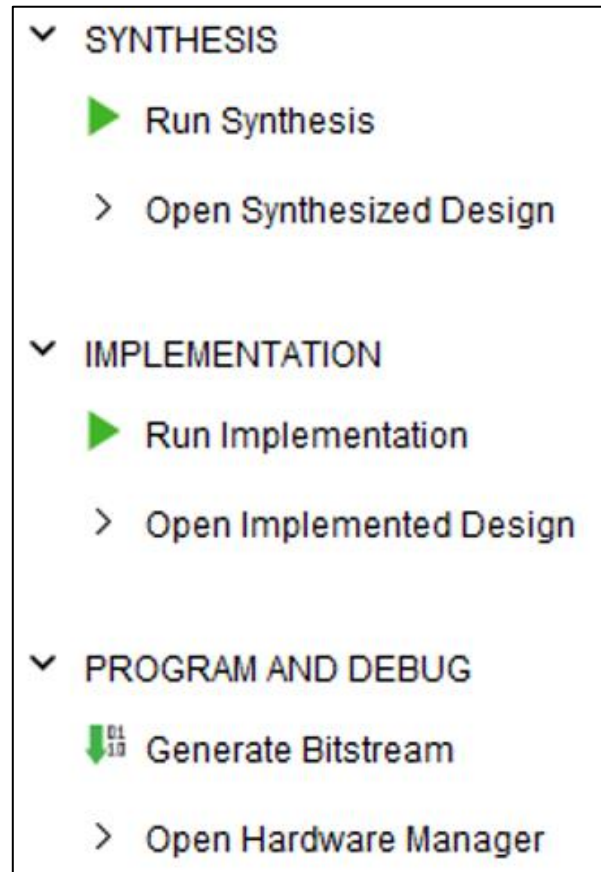
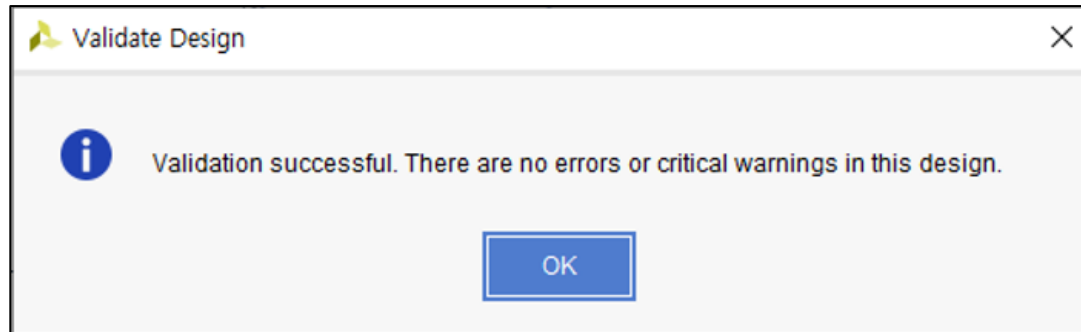
- 완성된 디자인 모습



7. Bitstream 생성 및 SDK 실행

> Bitstream 생성하기

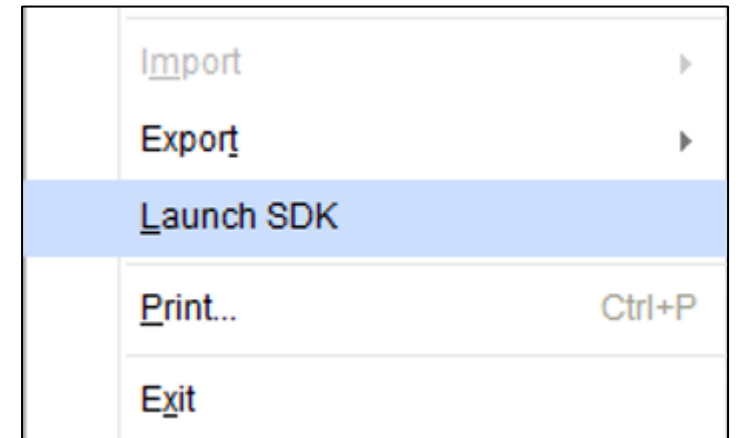
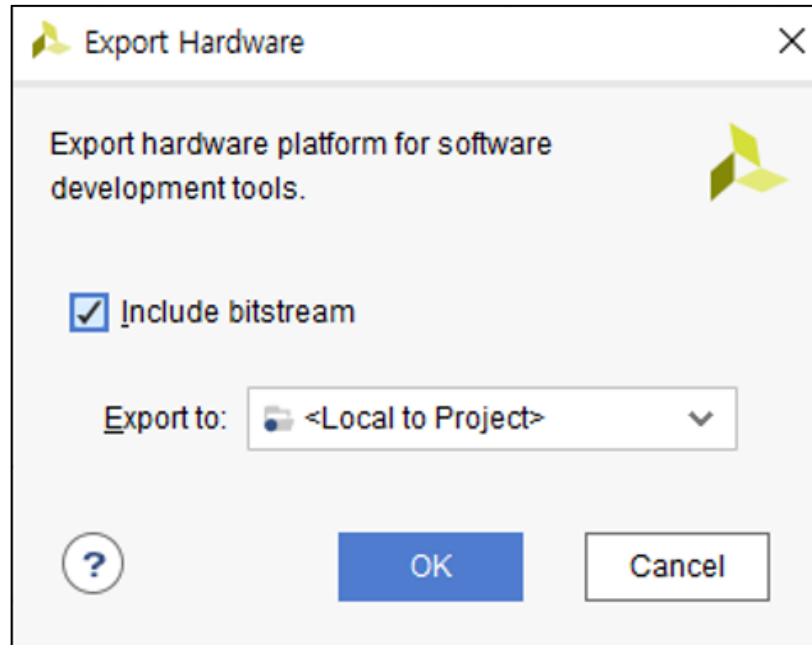
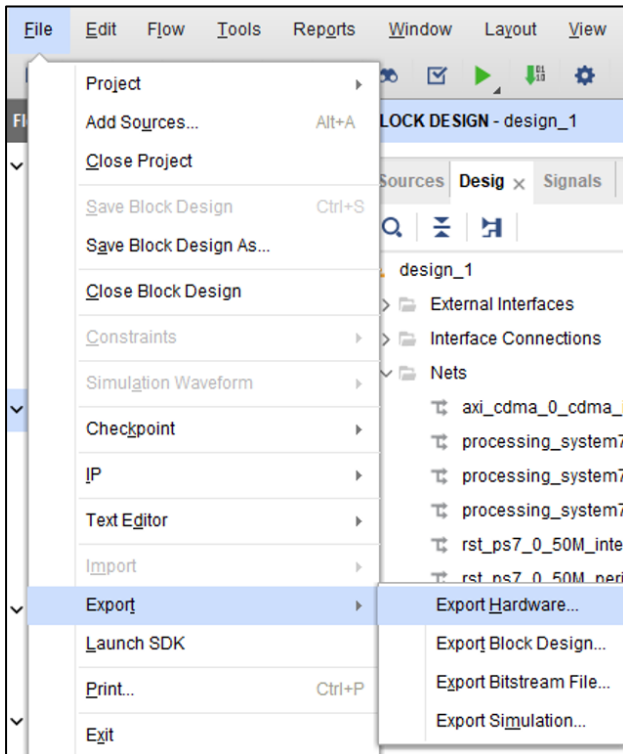
- Validation 후 Error가 없으면 design 저장한 후 Generate Bitstream 클릭



7. Bitstream 생성 및 SDK 실행

> SDK 실행하기

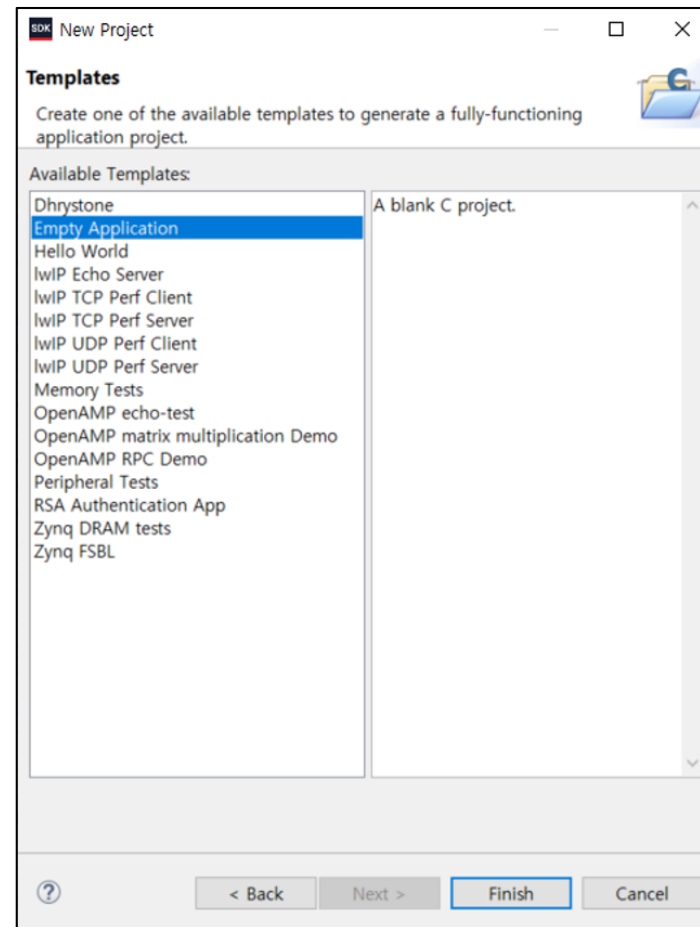
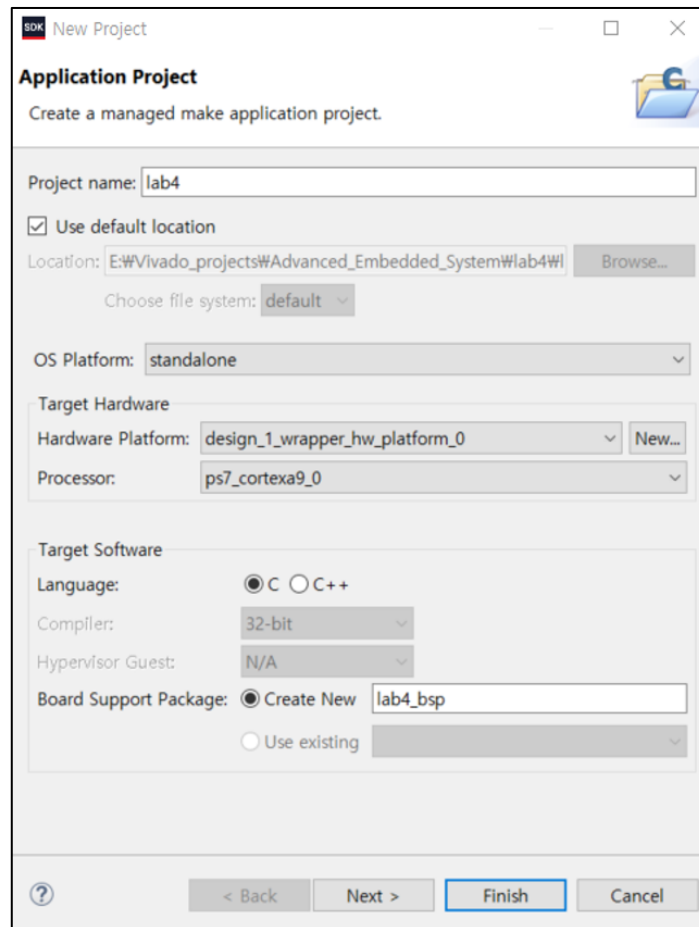
- File > Export > **Export Hardware** 클릭
- **Include bitstream** 반드시 체크
- File > Export > **Launch SDK** 클릭



8. SDK로 테스트하기

> New Application Project 생성

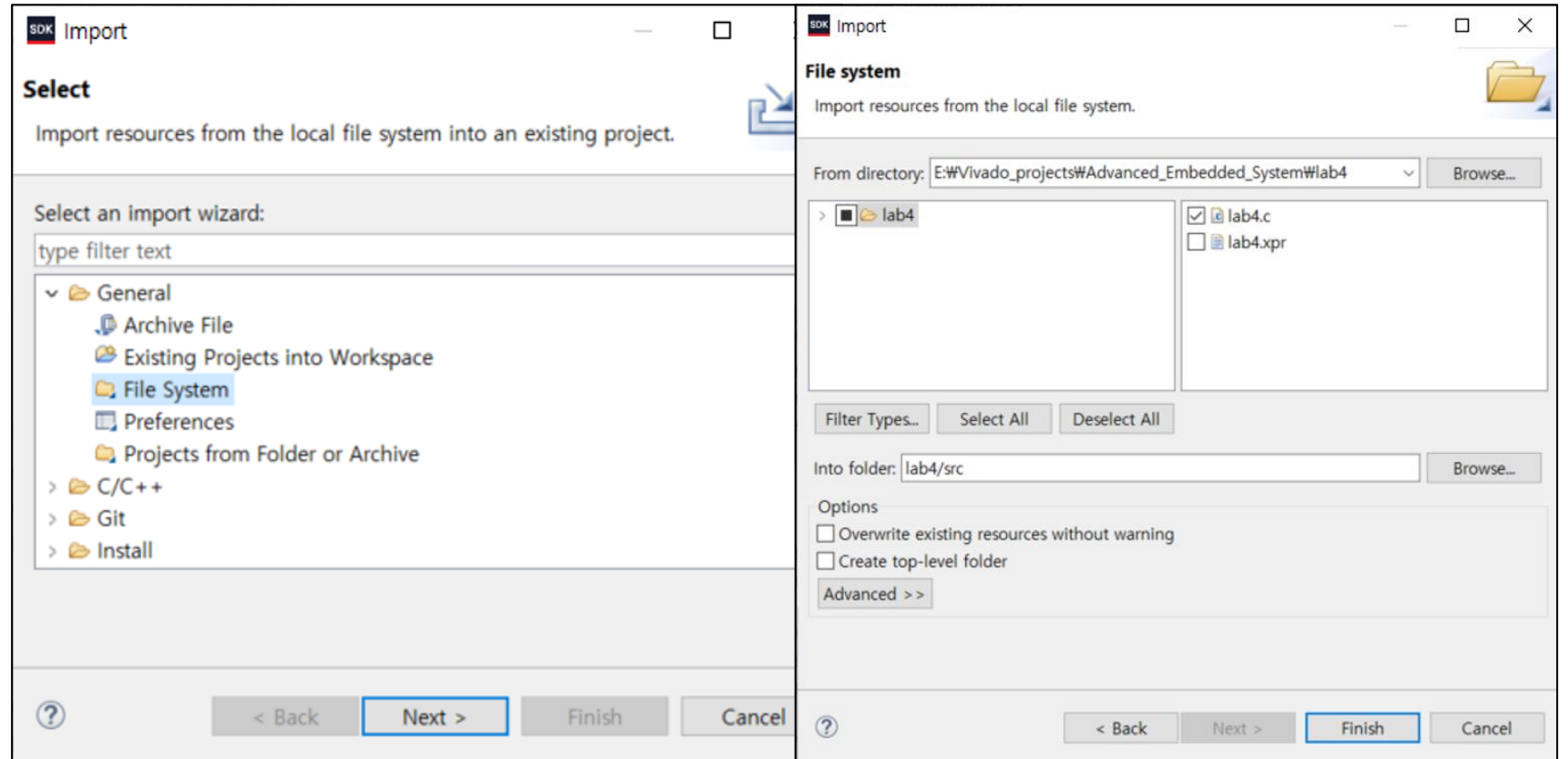
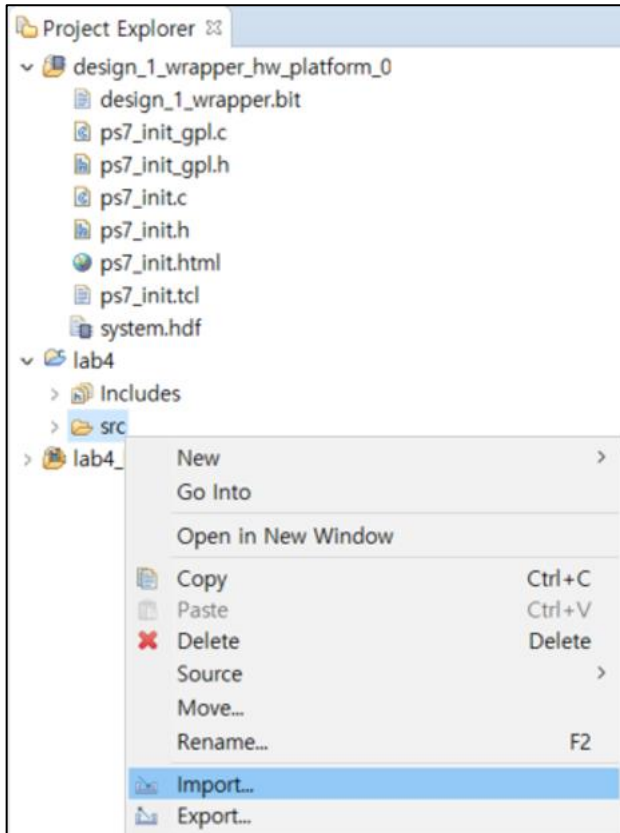
- Name: **lab4** 로 입력 -> Next 클릭 -> **Empty Application** 선택 -> Finish 클릭



8. SDK로 테스트하기

> lab4.c를 import하기

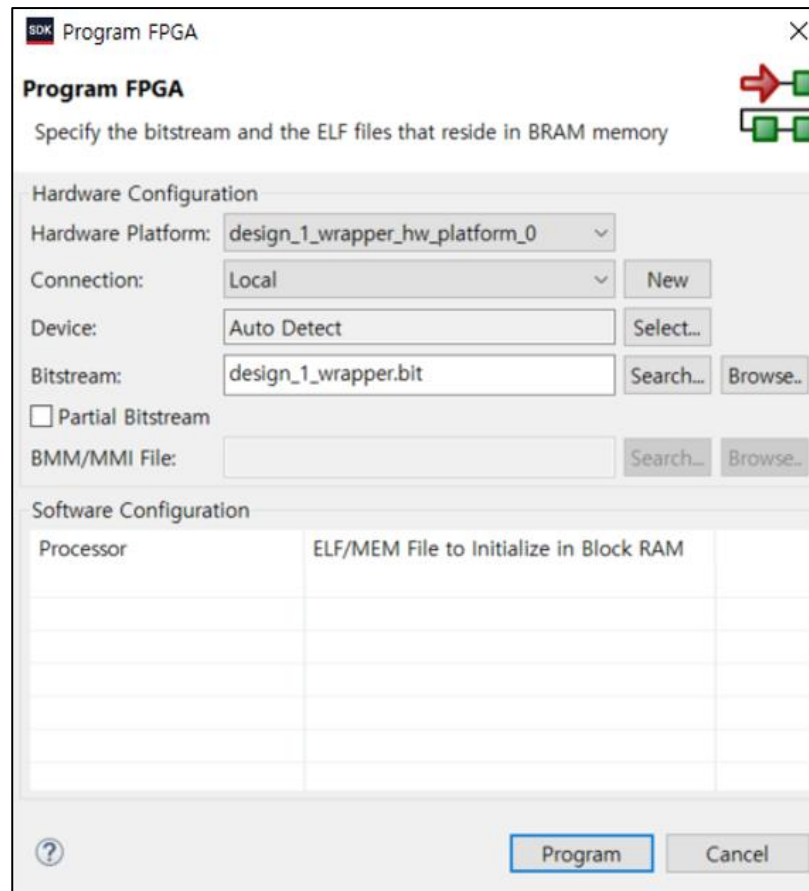
- lab4/src 폴더에서 마우스 우클릭 -> Import -> General/**File System** -> lab4.c가 위치한 폴더를 Browse -> lab4.c 선택 -> Finish



8. SDK로 테스트하기

> Program FPGA 진행

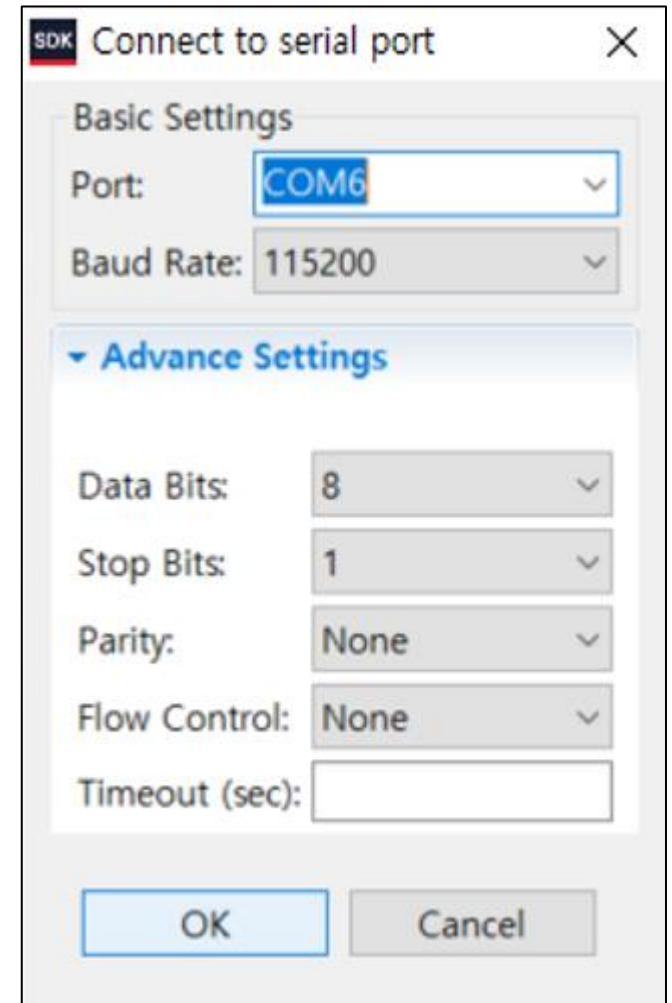
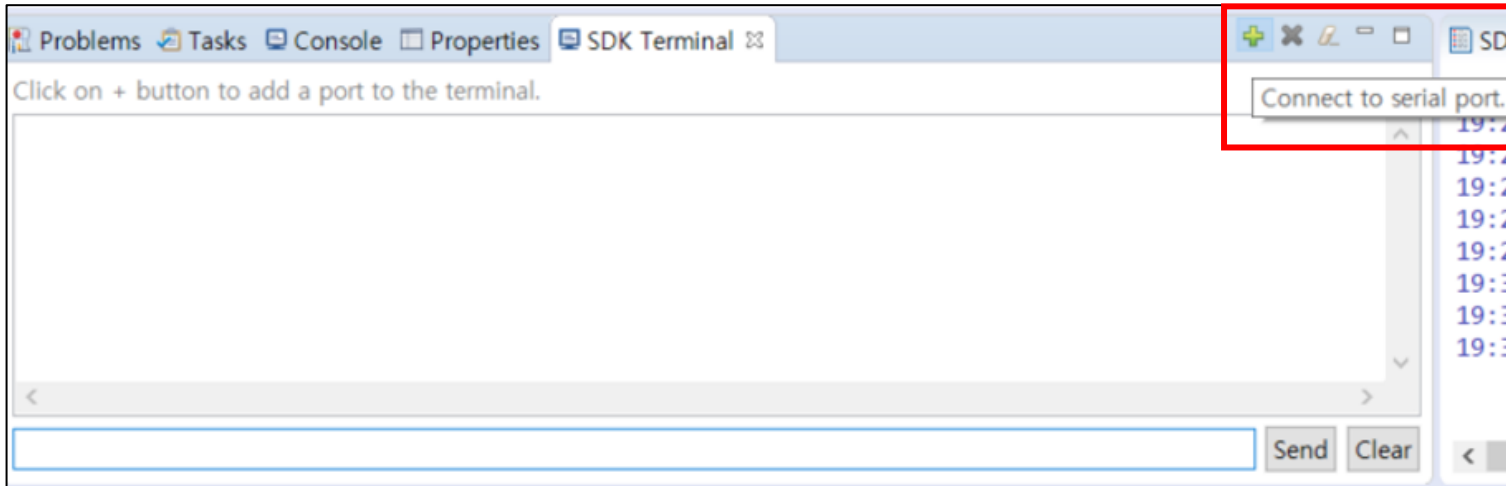
- 보드를 PC에 연결한 후 Program FPGA 진행



8. SDK로 테스트하기

> SDK 터미널 열기

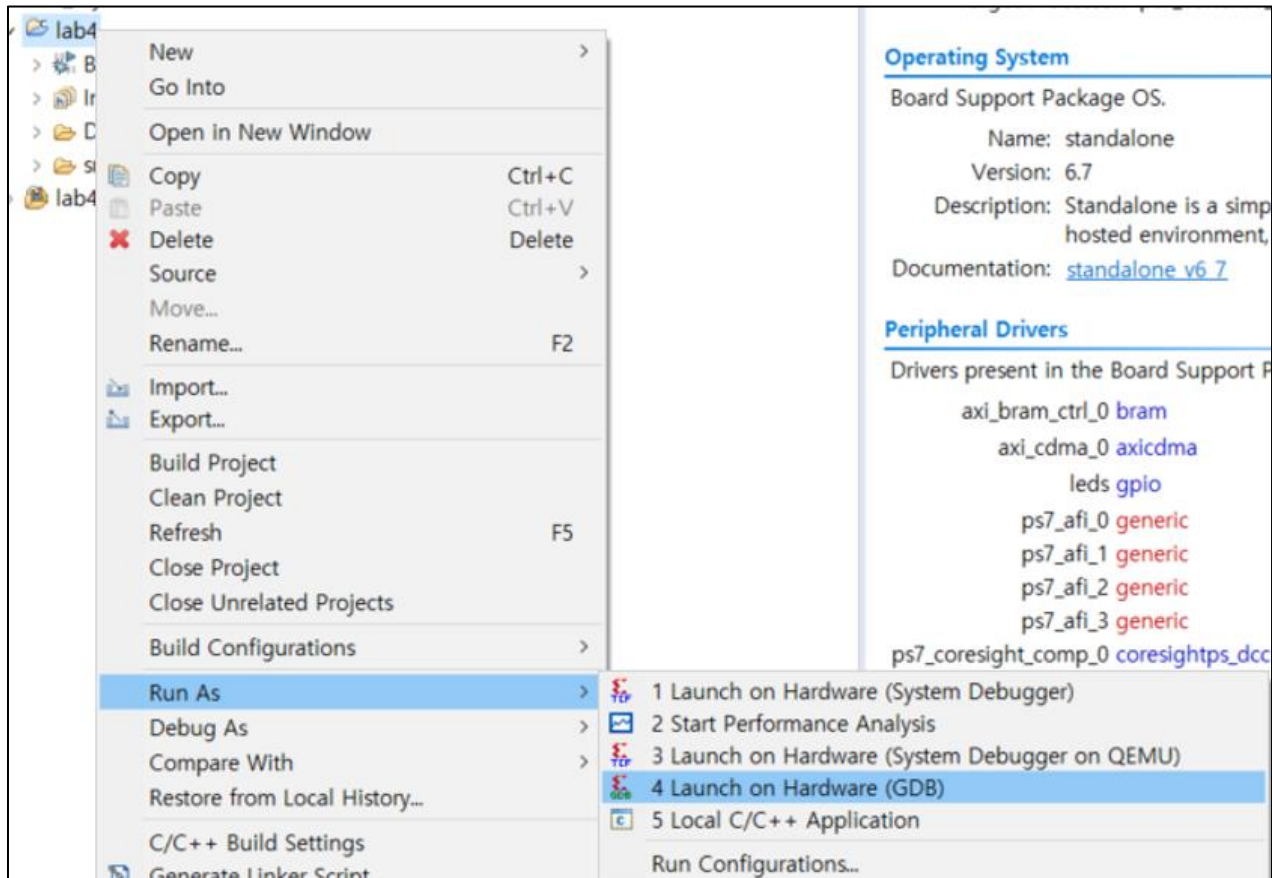
- 보드를 PC에 연결된 USB 포트를 터미널에 연결하도록 등록
- Baud rate : 115200



8. SDK로 테스트하기

> 하드웨어 실행하기

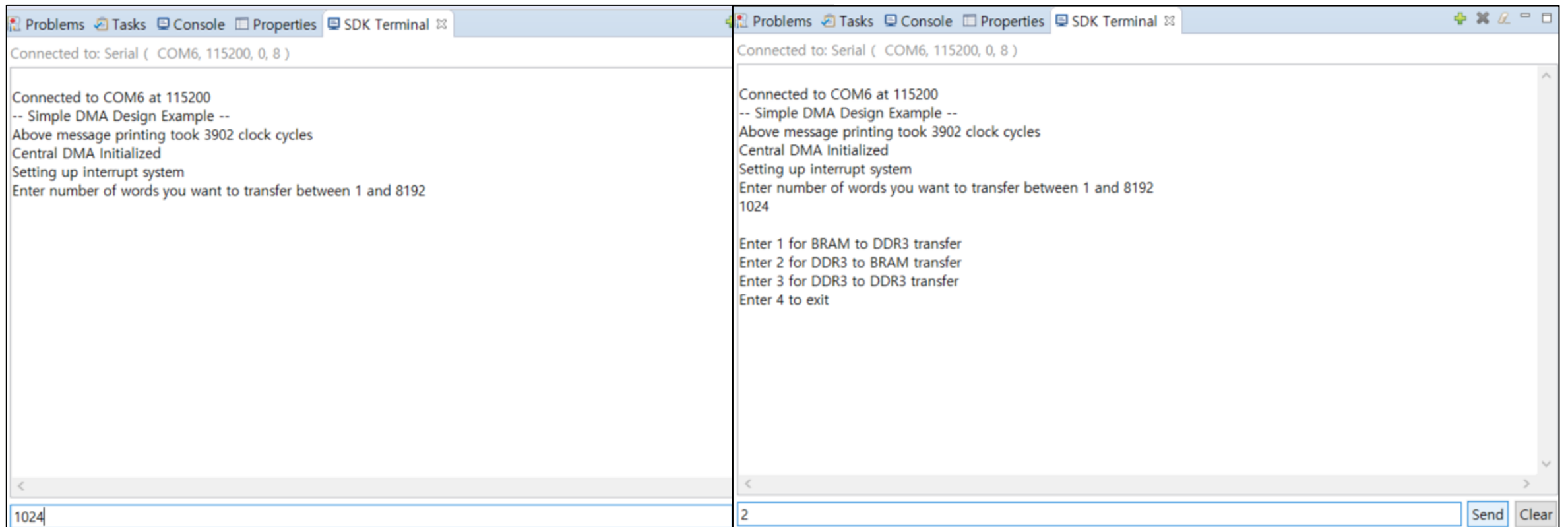
- lab4 폴더 우클릭 -> Run As > **4 Launch on Hardware (GDB)** 클릭



8. SDK로 테스트하기

> 하드웨어 실행하기

- 다음과 같이 SDK 터미널에 내용이 나타남
- 예시로 1024 words(4096 bytes)를 DDR3에서 BRAM으로 transfer하도록 입력하였음



The image displays two side-by-side screenshots of the SDK Terminal window, showing the execution of a DMA transfer test. Both windows show the same initial output: "Connected to: Serial (COM6, 115200, 0, 8)", "Connected to COM6 at 115200", "-- Simple DMA Design Example --", "Above message printing took 3902 clock cycles", "Central DMA Initialized", "Setting up interrupt system", and "Enter number of words you want to transfer between 1 and 8192".

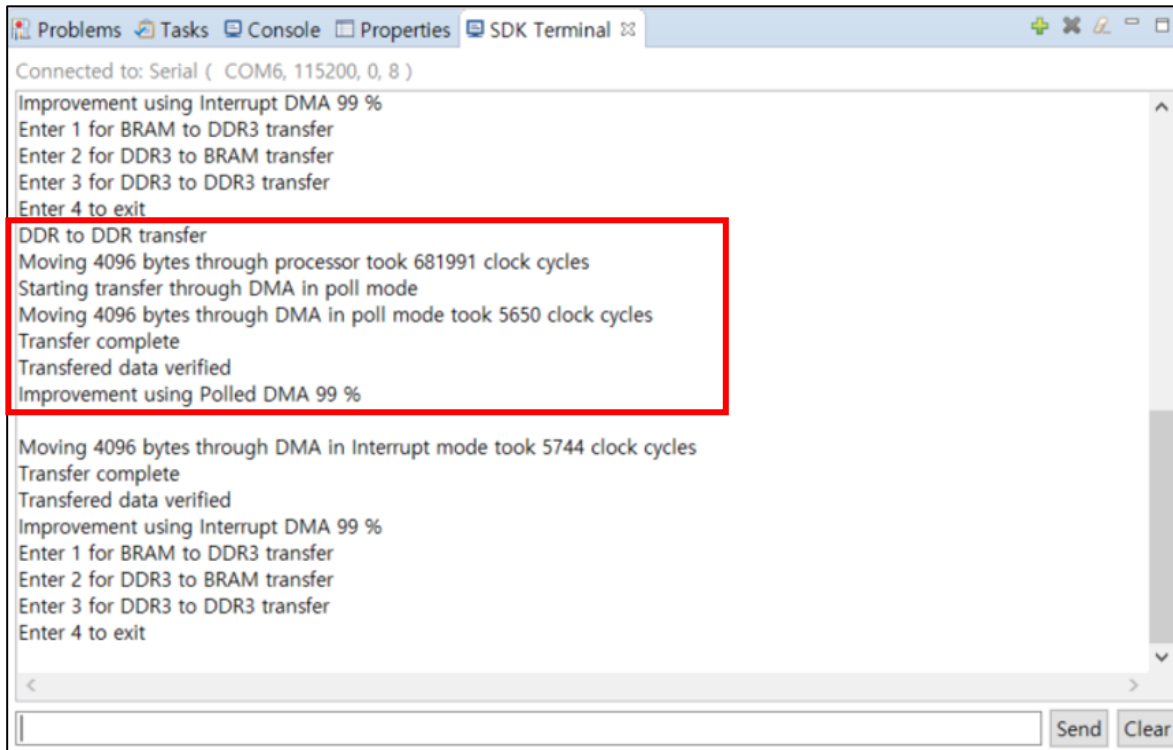
In the left screenshot, the input field at the bottom contains the number "1024".

In the right screenshot, the input field at the bottom contains the number "2". The terminal output below the input shows the menu options: "Enter 1 for BRAM to DDR3 transfer", "Enter 2 for DDR3 to BRAM transfer", "Enter 3 for DDR3 to DDR3 transfer", and "Enter 4 to exit".

8. SDK로 테스트하기

> 하드웨어 실행하기

- 1024 words(4096 bytes)를 DDR3에서 BRAM으로 transfer하도록 입력하였음
- Processor 자체에서 transfer를 수행하면 680000 cycle 이상이 걸리지만 interrupt를 사용해 DMA를 동작시켜 transfer를 수행하였더니 5600 cycle 정도가 걸리는 것을 볼 수 있음



```
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( COM6, 115200, 0, 8 )
Improvement using Interrupt DMA 99 %
Enter 1 for BRAM to DDR3 transfer
Enter 2 for DDR3 to BRAM transfer
Enter 3 for DDR3 to DDR3 transfer
Enter 4 to exit
DDR to DDR transfer
Moving 4096 bytes through processor took 681991 clock cycles
Starting transfer through DMA in poll mode
Moving 4096 bytes through DMA in poll mode took 5650 clock cycles
Transfer complete
Transferred data verified
Improvement using Polled DMA 99 %

Moving 4096 bytes through DMA in Interrupt mode took 5744 clock cycles
Transfer complete
Transferred data verified
Improvement using Interrupt DMA 99 %
Enter 1 for BRAM to DDR3 transfer
Enter 2 for DDR3 to BRAM transfer
Enter 3 for DDR3 to DDR3 transfer
Enter 4 to exit
```

감사합니다