AMBA and AXI Connect [Servo Motor] Example

Micro Servo Module

[*Reference*] ▪

Table of Contents

➢ *SoC Peripheral RTC Design Project* (프로젝트 최종 계획서 제출)

1. Project Exam
   1) Exam_Washing_Machine
   2) Exam_Microblaze_Washing_Machine_Block_Design
2. led_btn_Exam
3. AXI_My_IP_LED_Exam
4. AXI_My_IP_FND_Exam
5. AXI_RTC_Exam
6. AXI_PWM
7. AXI_LCD
8. Servo_Motor_Exam
9. DHT11_Module_Exam
10. Ultrasonic Module Exam

AMBA and AXI Connect [Servo Motor] Example

Micro Servo Module

[Create IP]

- ***AXI Connect ➔ AXI Template***

New IP Location

- AXI Connect ➔ AXI Template

VIVADO ML Editions

Create a New Customized IP Location

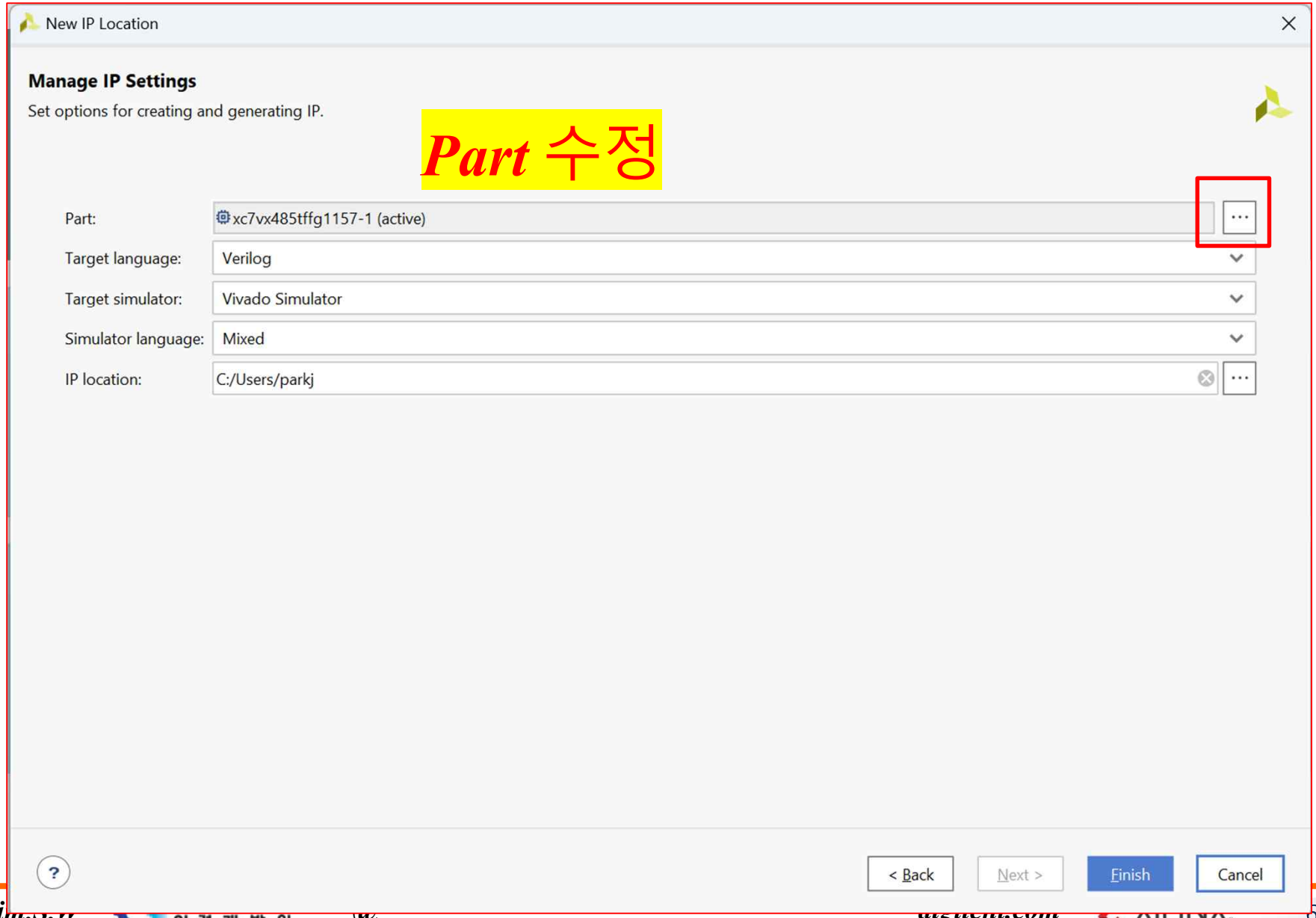
This wizard will guide you through the creation of a new customized IP location

The Manage IP flow requires a directory for your IP files to be specified. Additional options you will be able to specify include the targeted part, preferred language for IP files as well as supported simulator.

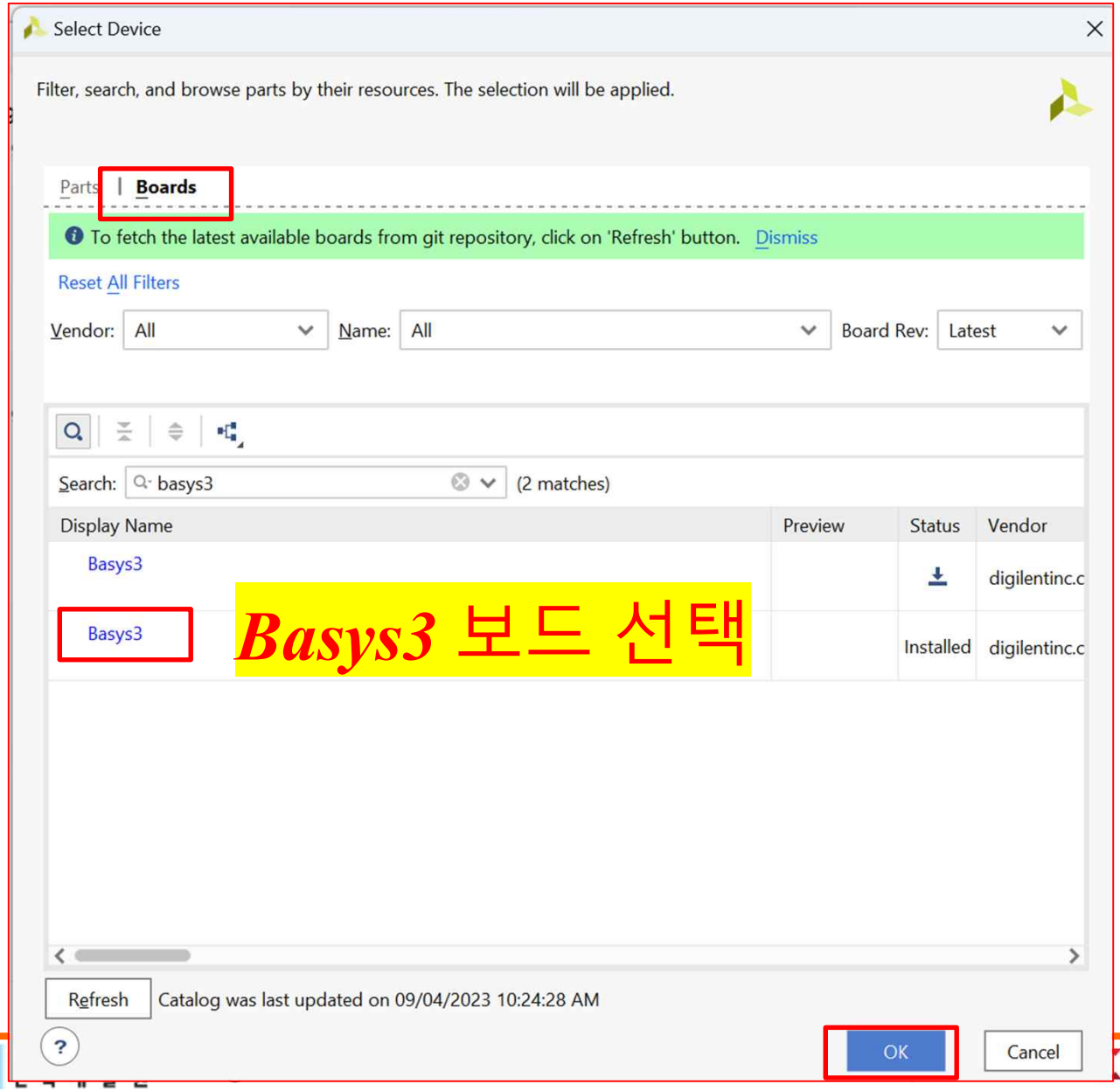
XILINX

? < Back | Next > | Finish | Cancel

- AXI Connect ➔ AXI Template



- ***AXI Connect ➔ AXI Template***



AMBA and AXI Connect [Servo Motor]

Axi_Servo_Exam.xpr

New IP Location

➢ AXI Connect ➔ AXI Template

Manage IP Settings

Set options for creating and generating IP.

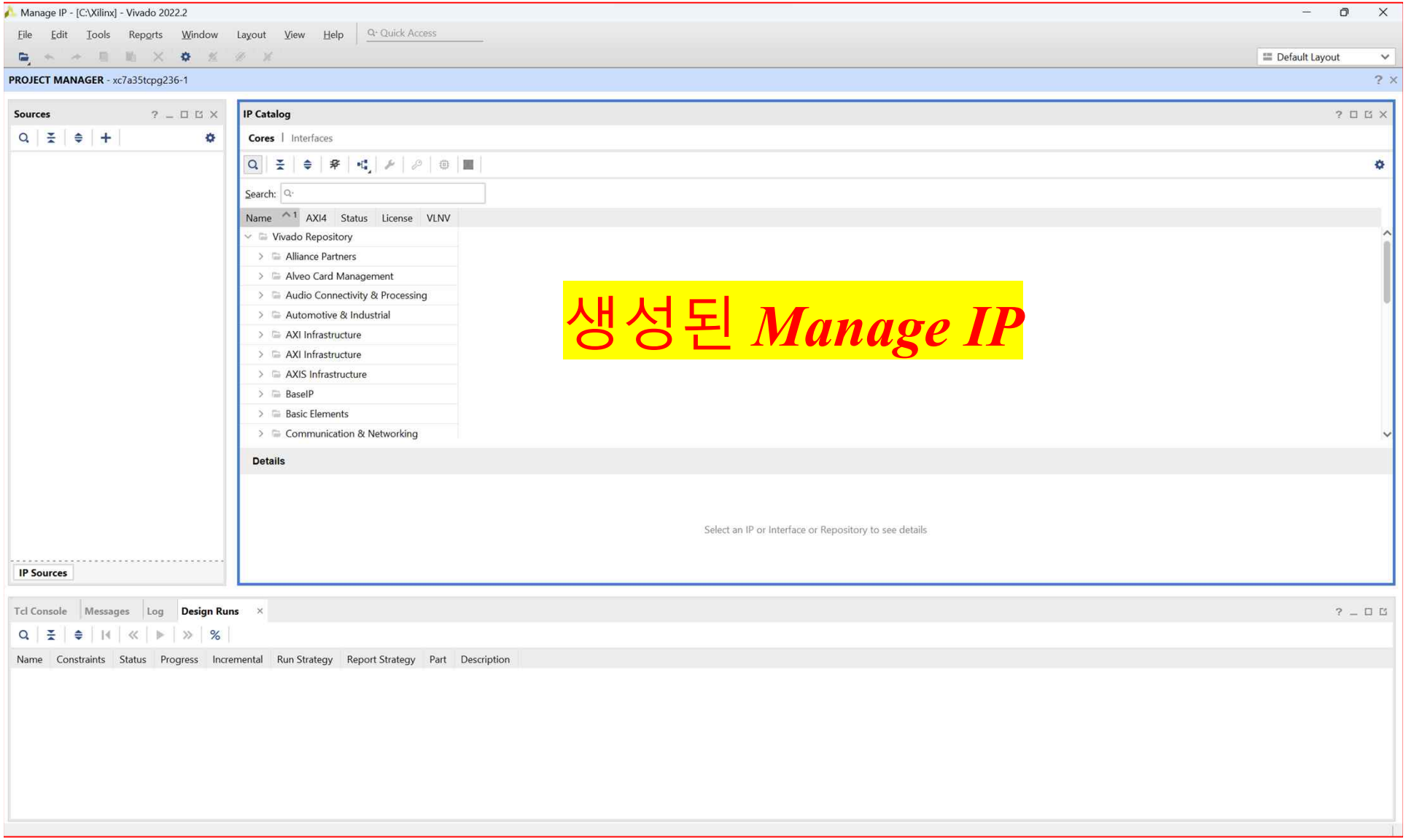
Part:	Basys3 (xc7a35tcpg236-1)
Target language:	Verilog
Target simulator:	Vivado Simulator
Simulator language:	Mixed
IP location:	C:/Users/parkj

원하는 IP location 지정

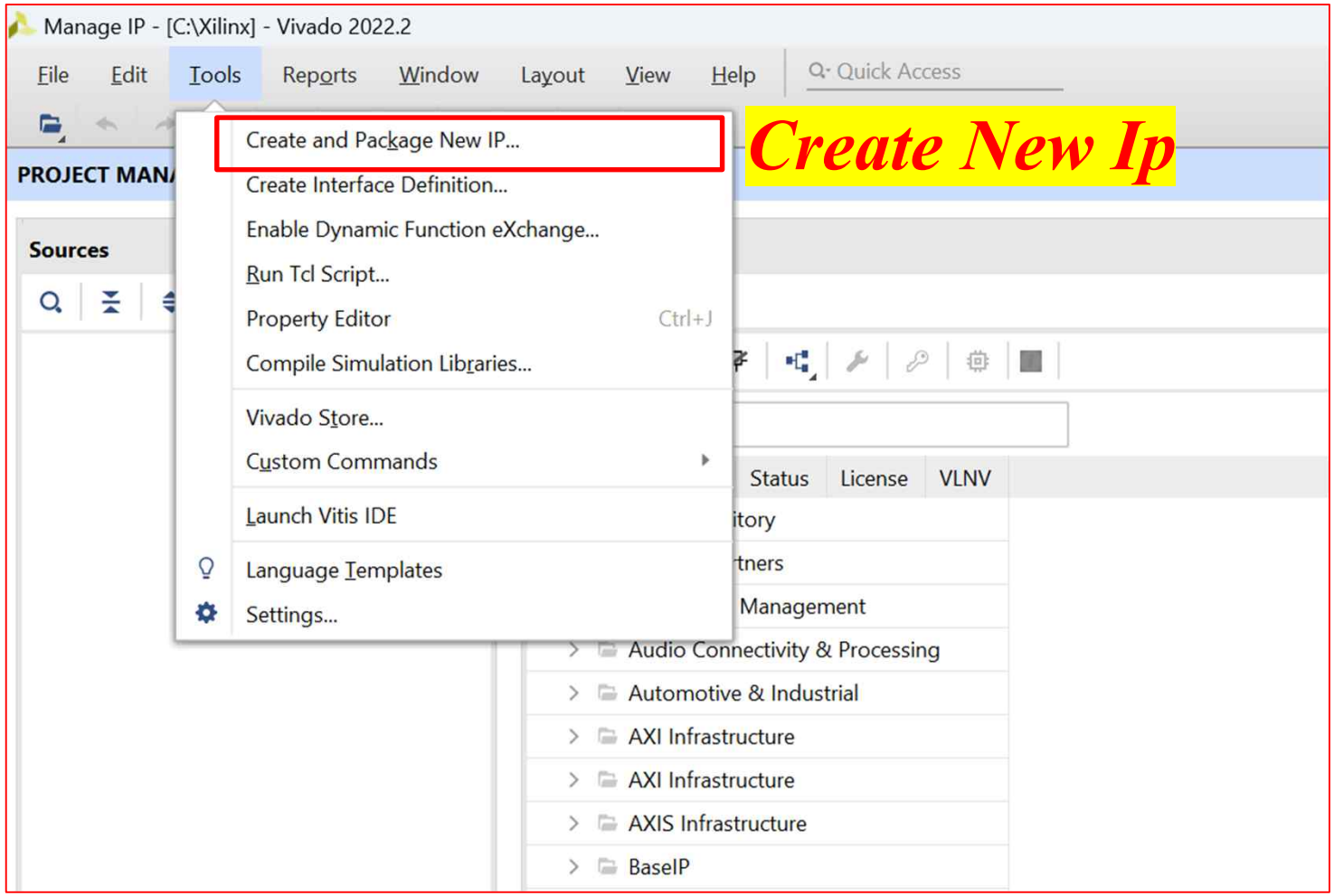
Finish

< Back | Next > | Finish | Cancel

- *AXI Connect ➔ AXI Template*



- *AXI Connect ➔ AXI Template*



- AXI Connect ➔ AXI Template

Create and Package New IP

Create Peripheral, Package IP or Package a Block Design

Please select one of the following tasks.

Packaging Options

- Package your current project
  Use the project as the source for creating a new IP Definition.
- Package a block design from the current project
  Choose a block design as the source for creating a new IP Definition.
  Select a block design: axi_connect_1
- Package a specified directory
  Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

- Create a new AXI4 peripheral
  Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.

Create New Ip

? < Back Next > Finish Cancel

➢ AXI Connect ➔ AXI Template

Create and Package New IP

Peripheral Details

Specify name, version and description for the new peripheral

Name: servo_axi

Version: 1.0

Display name: servo_axi_v1.0

Description: My new AXI IP

IP location: C:/Xilinx/managed_ip_project/../ip_repo

Overwrite existing

이름 변경 후 Next

< Back | Next > | Finish | Cancel

➢ AXI Connect ➔ AXI Template

Create and Package New IP

Add Interfaces

Add AXI4 interfaces supported by your peripheral

Enable Interrupt Support

S00_servo_AXI

servo_axi_v1.0

Interfaces

S00_serv

Name: S00_servo_AXI

Interface Type: Lite

Interface Mode: Slave

Data Width (Bits): 32

Memory Size (Bytes): 64

Number of Registers: 4 [4..512]

이름 변경 후 Next

< Back | Next > | Finish | Cancel

- *AXI Connect ➔ AXI Template*

Create and Package New IP

VIVADO ML Editions

Create Peripheral

Peripheral Generation Summary

1. IP (xilinx.com:user:myip_axi_led:1.0) with 1 interface(s)
2. Driver(v1_00_a) and testapp more info
3. AXI4 VIP Simulation demonstration design more info
4. AXI4 Debug Hardware Simulation demonstration design more info

Peripheral created will be available in the catalog :

F:/Xilnx/2022/axi_connet_exam/../ip_repo

Next Steps:

- (●) Add IP to the repository
- ( ) Edit IP
- ( ) Verify Peripheral IP using AXI4 VIP
- ( ) Verify peripheral IP using JTAG interface

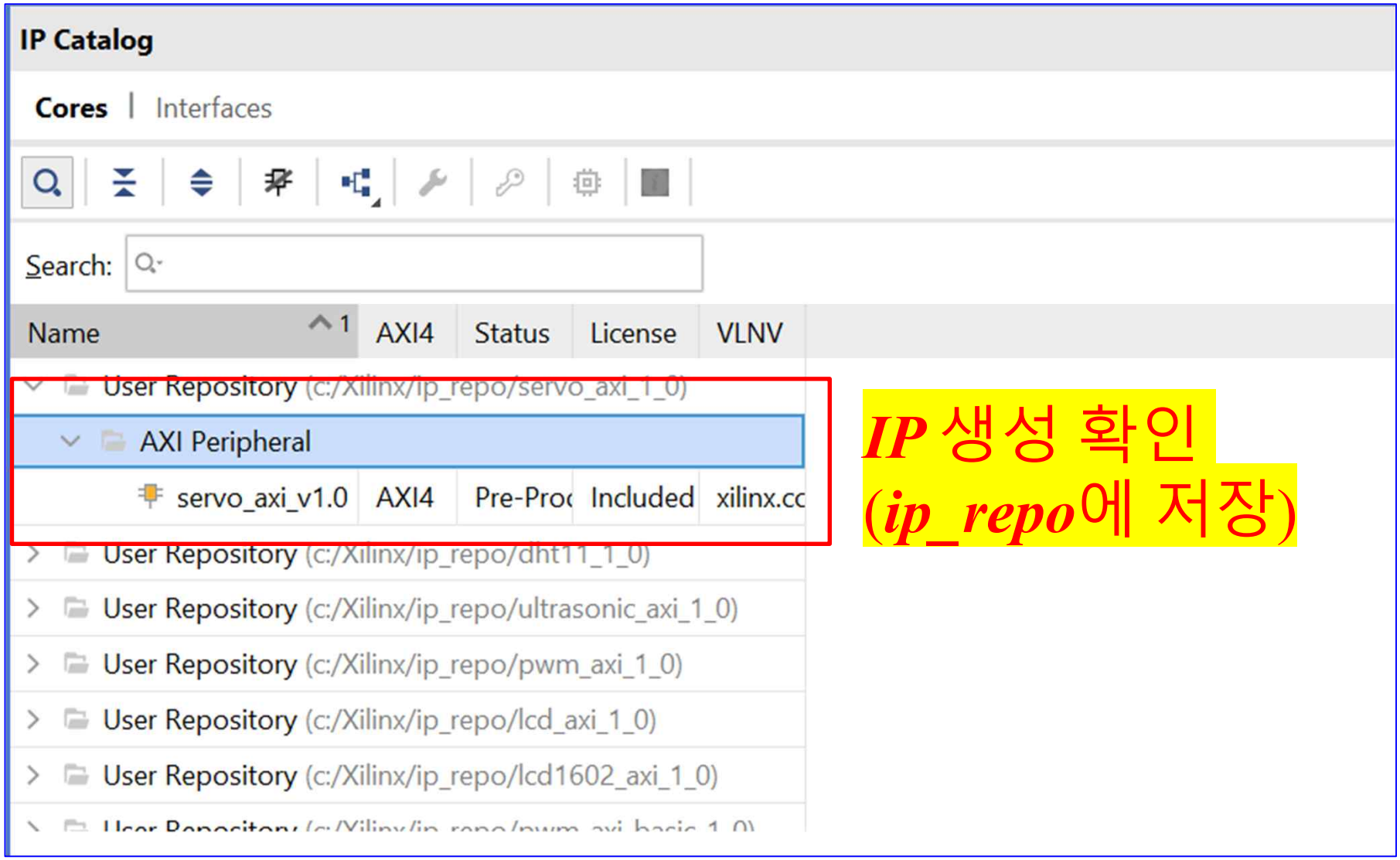
XILINX

Click Finish to continue

Finish

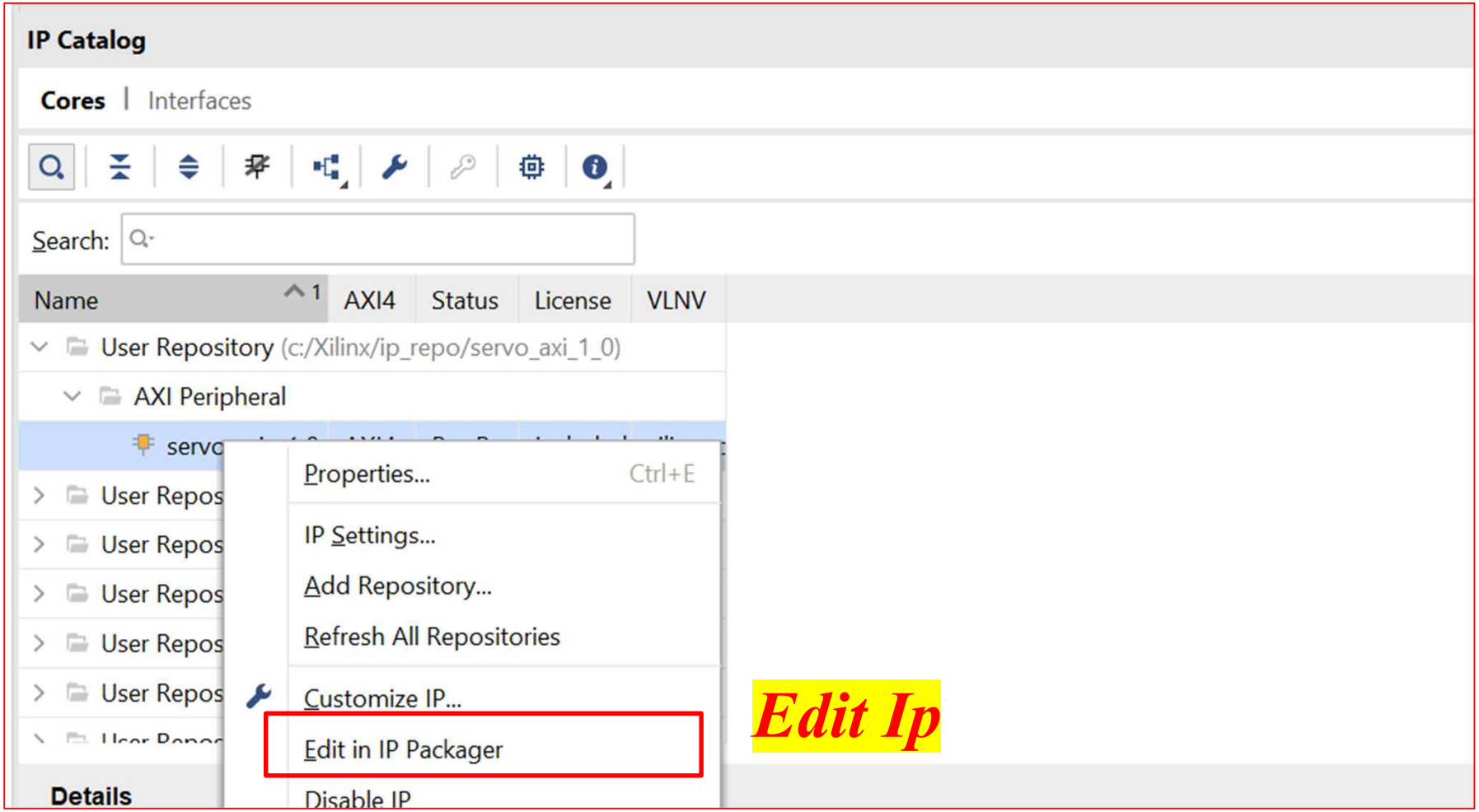
? < Back Next > Finish Cancel

➢ *AXI Connect ➔ AXI Template*

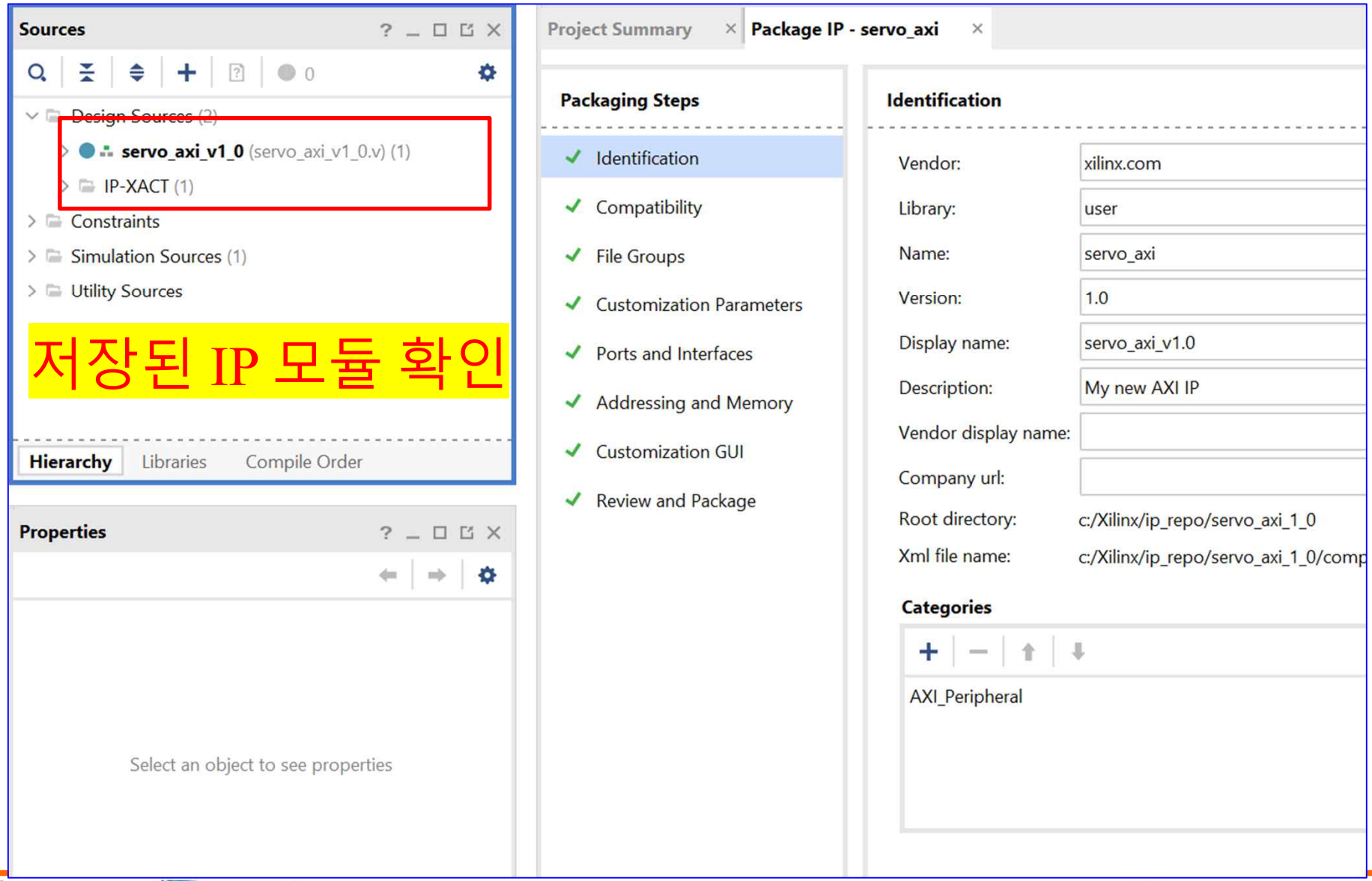


***IP* 생성 확인**
(*ip_repo*에 저장)

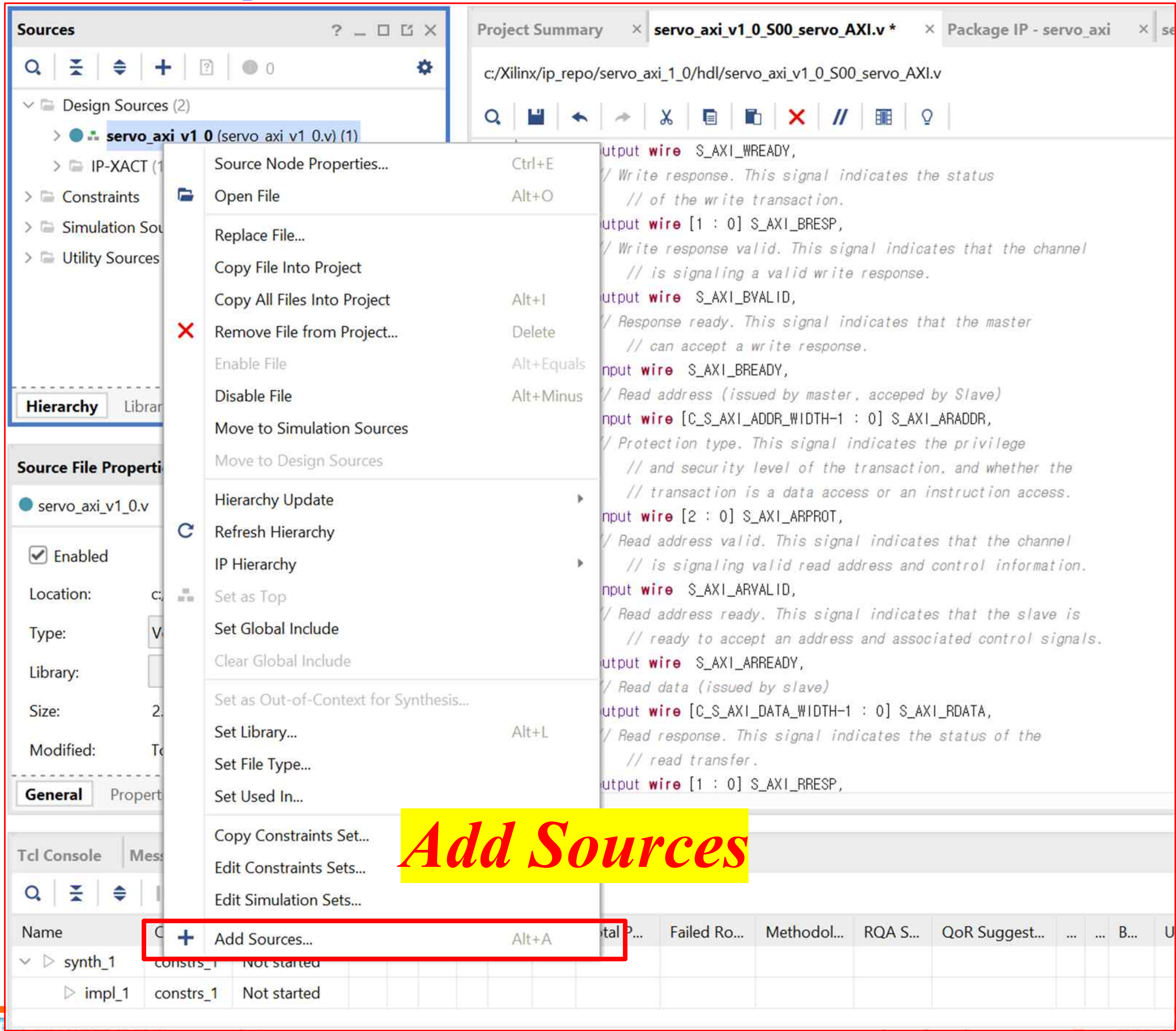
- *AXI Connect ➔ AXI Template*



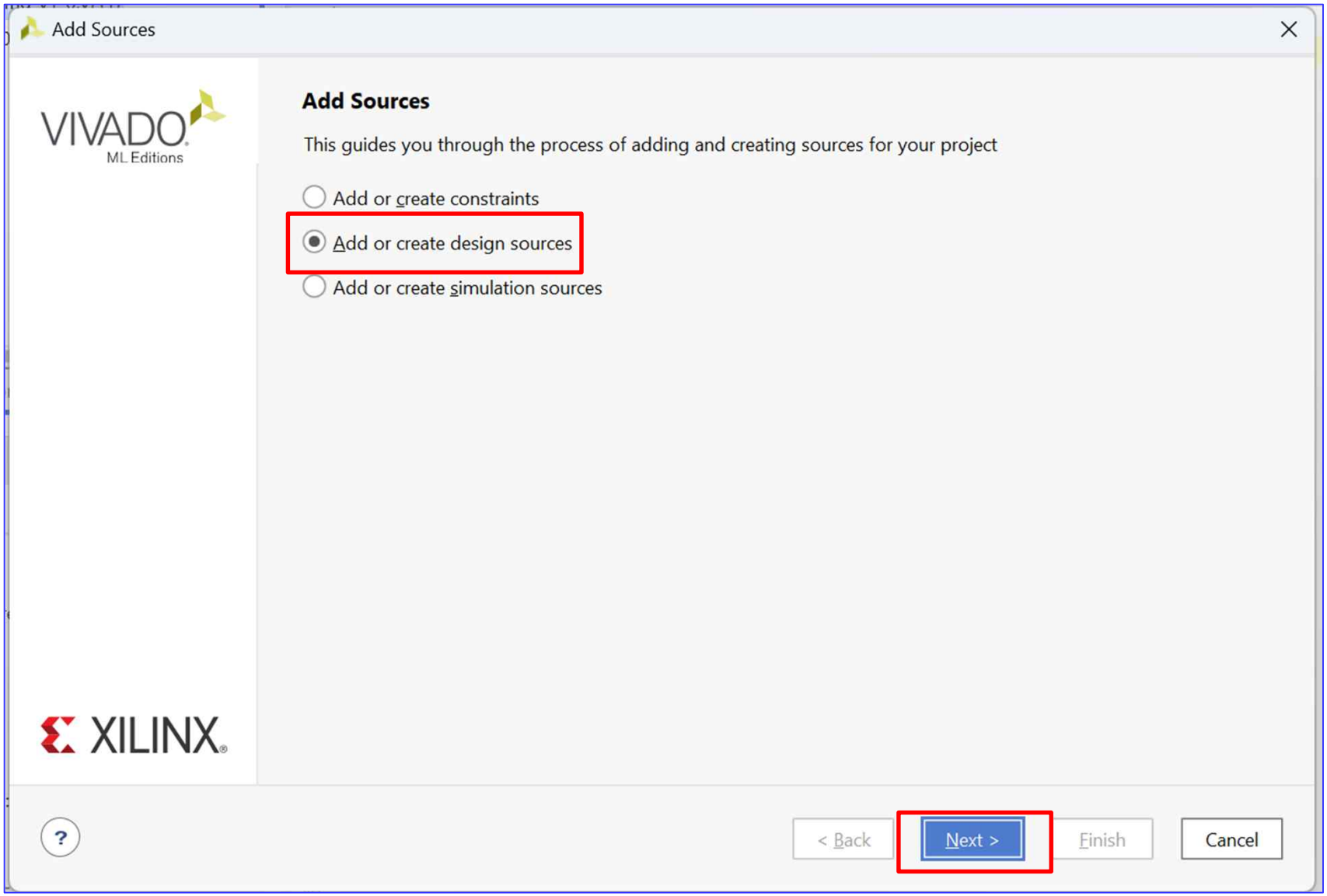
➢ *AXI Connect ➔ AXI Template*



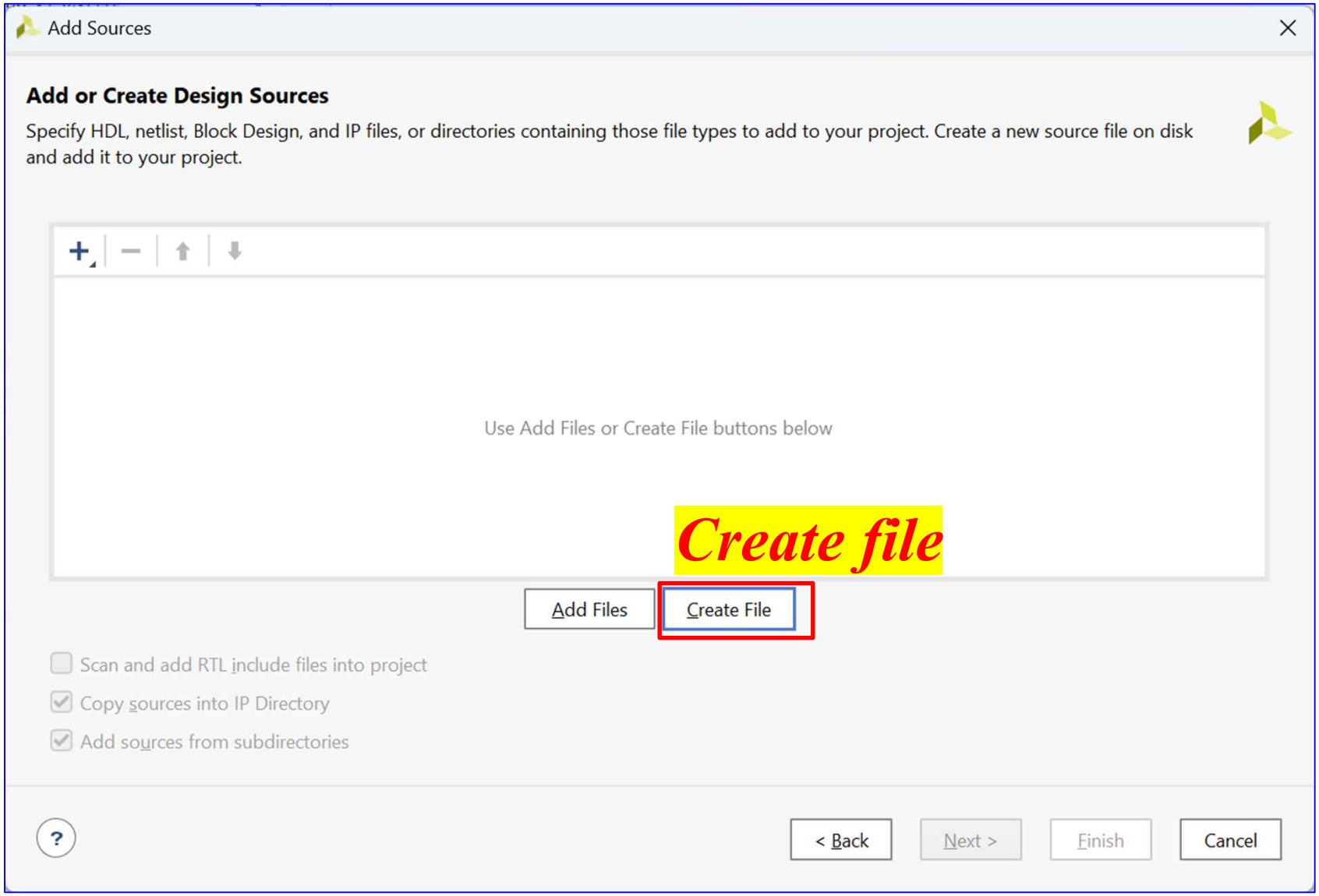
➢ *AXI Connect ➔ AXI Template*



- AXI Connect ➔ AXI Template



➢ AXI Connect ➔ AXI Template



➢ AXI Connect ➔ AXI Template

Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

Index	Name	Library	Location
1	servo.v	xil_defaultlib	C:/Users/parkj/231221_servo
2	clockdivider_10000hz.v	xil_defaultlib	C:/Users/parkj/231221_servo

Source 추가

Add Files | Create File

Scan and add RTL include files into project

Copy sources into IP Directory

Add sources from subdirectories

? < Back | Next > | Finish | Cancel

```
`timescale 1ns / 1ps
module servo(
 input clk, reset,
    input clk10000hz,
    input [7:0] data,
    output reg Servo
    );

    reg [1:0] mode;
    reg [7:0] duty;
    reg [9:0] cnt_duty;


always @(posedge clk or negedge reset) begin
        if(~reset) begin
            duty = 0;
        end else begin
       duty = data;
  end
end

always @(posedge clk10000hz or negedge reset) begin
        if(~reset) begin
            cnt_duty = 0;
            Servo = 0;
        end else begin

    if(cnt_duty >= 200) cnt_duty = 0;
        else cnt_duty = cnt_duty + 1;

    if(cnt_duty < duty) Servo = 1;
        else Servo = 0;
  end
end

endmodule
```

```
`timescale 1ns / 1ps
module clockdivider_10000hz (
 input  clk,
    input  reset,
    output reg  clk10000hz
);
  reg [25:0] cnt = 0;
  always @(posedge clk, negedge reset) begin
    if (~reset) begin
      cnt <= 0;
    end else begin
      if (cnt == (5_000 - 1)) begin
        cnt <= 0;
        clk10000hz <= ~clk10000hz;
      end else begin
        cnt <=  cnt + 1;
      end
    end
  end
endmodule
```

➢ AXI Connect ➔ AXI Template

Project Summary × Package IP - servo_axi × servo_axi_v1_0.v * ×

c:/Xilinx/ip_repo/servo_axi_1_0/hdl/servo_axi_v1_0.v

```
16      (
17          // Users to add ports here
18          output wire Servo,
19          // User ports ends
20          // Do not modify the ports beyond this line
21
22
23          // Ports of Axi Slave Bus Interface S00_servo_AXI
```

Project Summ 추가 rvo_axi_v1_0_S00_servo_AXI.v * × Package

c:/Xilinx/ip_repo/servo_axi_1_0/hdl/servo_axi_v1_0_S00_servo_AXI.v

```
13          // Width of S_AXI address bus
14          parameter integer C_S_AXI_ADDR_WIDTH    = 4
15      )
16      (
17          // Users to add ports here
18          output wire Servo,
```

Project Summary × servo_axi_v1_0.v * × servo_axi_v

c:/Xilinx/ip_repo/servo_axi_1_0/hdl/servo_axi_v1_0.v

```
55          .S_AXI_AWVALID(s00_servo_axi_awvalid),
56          .S_AXI_AWREADY(s00_servo_axi_awready),
57          .S_AXI_WDATA(s00_servo_axi_wdata),
58          .S_AXI_WSTRB(s00_servo_axi_wstrb),
59          .S_AXI_WVALID(s00_servo_axi_wvalid),
60          .S_AXI_WREADY(s00_servo_axi_wready),
61          .S_AXI_BRESP(s00_servo_axi_bresp),
62          .S_AXI_BVALID(s00_servo_axi_bvalid),
63          .S_AXI_BREADY(s00_servo_axi_bready),
64          .S_AXI_ARADDR(s00_servo_axi_araddr),
65          .S_AXI_ARPROT(s00_servo_axi_arprot),
66          .S_AXI_ARVALID(s00_servo_axi_arvalid),
67          .S_AXI_ARREADY(s00_servo_axi_arready),
68          .S_AXI_RDATA(s00_servo_axi_rdata),
69          .S_AXI_RRESP(s00_servo_axi_rresp),
70          .S_AXI_RVALID(s00_servo_axi_rvalid),
71          .S_AXI_RREADY(s00_servo_axi_rready),
72          .Servo(Servo)
73      );
74
75      // Add user logic here
76
77      // User logic ends
78
```

추가

AMBA and AXI Connect [Servo Motor]

➢ *AXI Connect ➔ AXI Template*

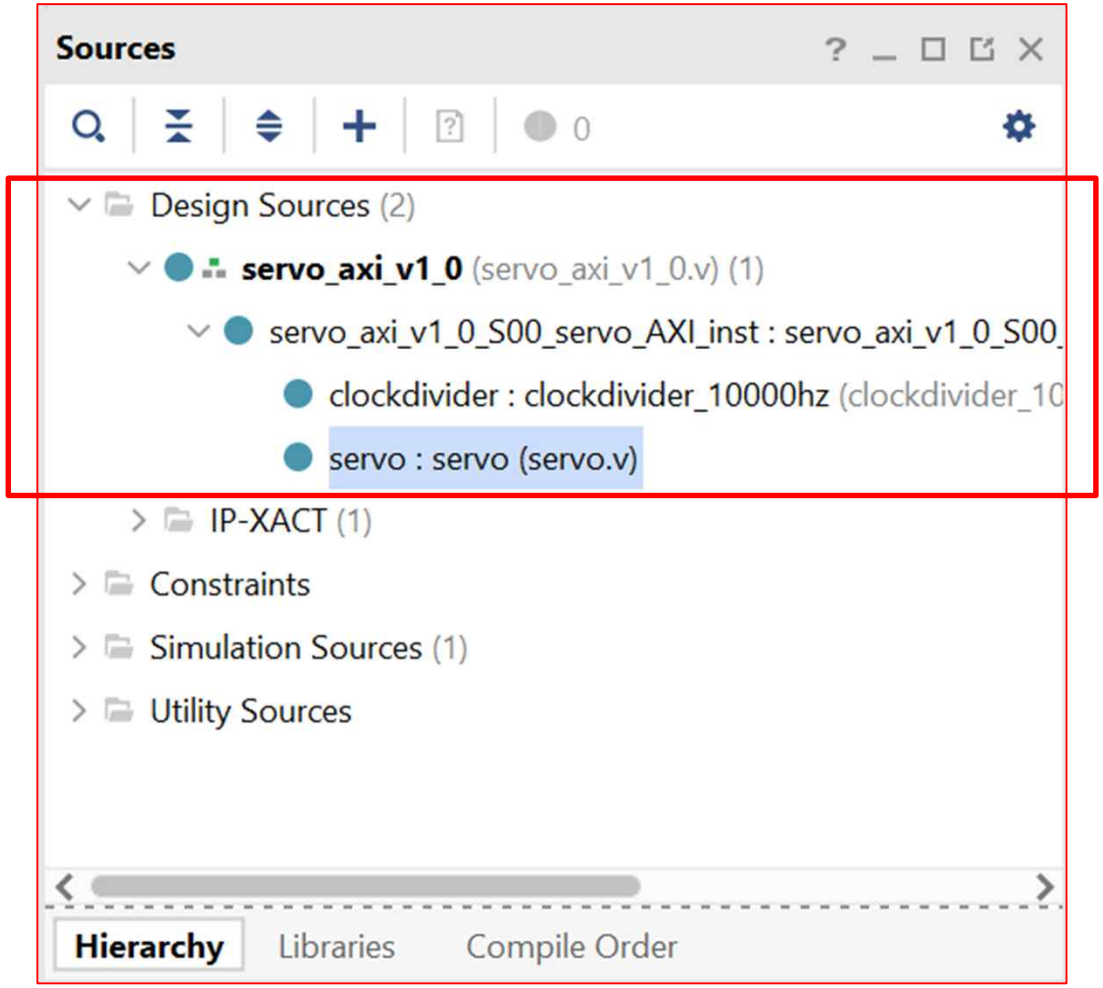
Project Summary × servo_axi_v1_0_S00_servo_AXI.v × P_Exam.xpr

c:/Xilinx/ip_repo/servo_axi_1_0/hdl/servo_axi_v1_0_S00_servo_AXI.v

```
391            // acceptance of read address by the slave
392            // output the read dada
393            if (slv_reg_rden)
394              begin
395                axi_rdata <= reg_data_out;     // regis
396              end
397          end
398      end
399
400      // Add user logic here
401
402  clockdivider_10000hz clockdivider(
403        .clk(S_AXI_ACLK),
404        .reset(S_AXI_ARESETN),
405        .clk10000hz(w_clkout)
406  );
407
408  servo servo(
409        .clk(S_AXI_ACLK),
410        .reset(S_AXI_ARESETN),
411        .clk10000hz(w_clkout),
412        .Servo(Servo),
413        .data(data)
414
415  );
416  wire [7:0] data;
417   assign data = slv_reg0[7:0];
418      // User logic ends
```

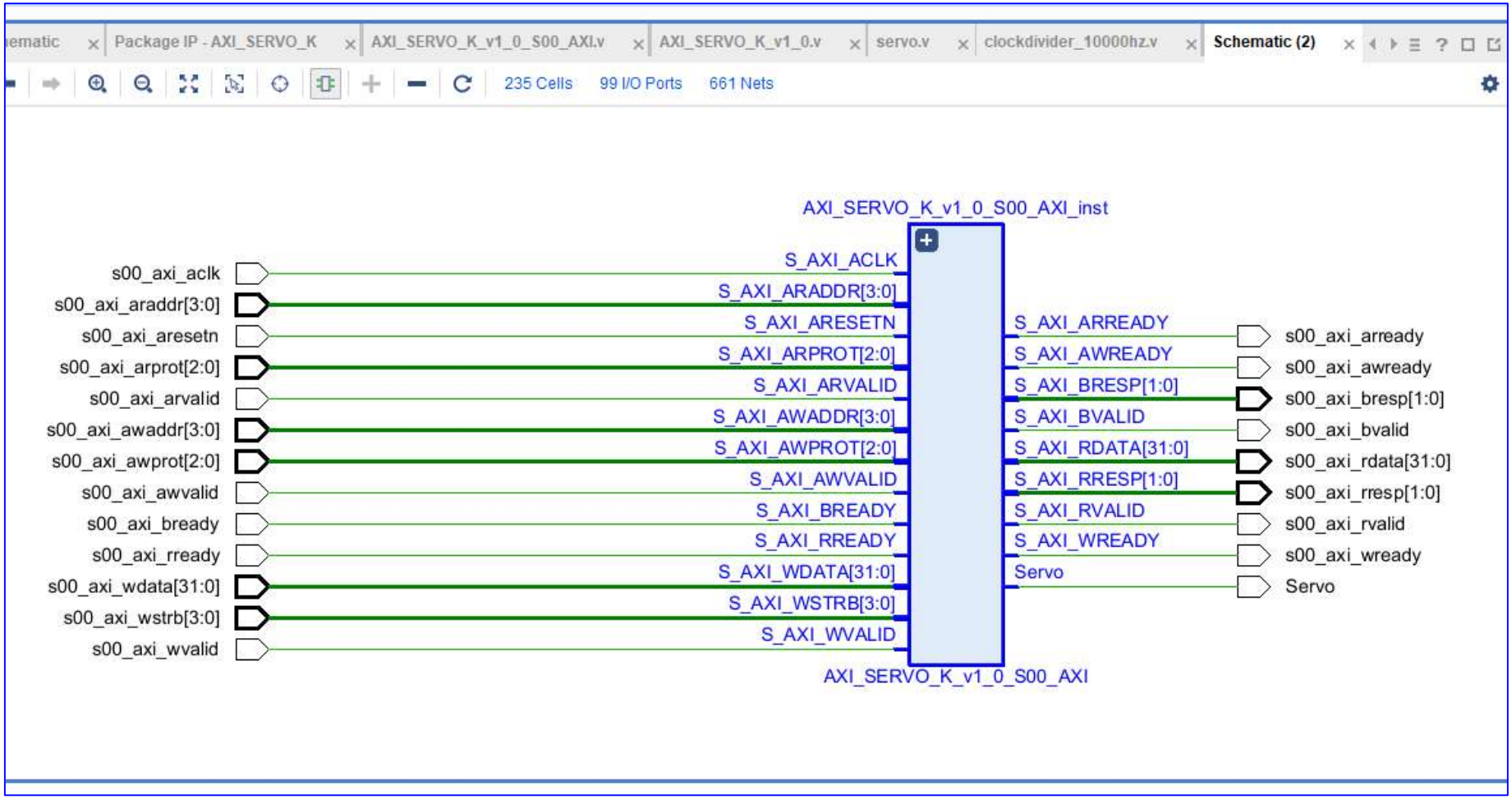
추가

- *AXI Connect ➔ AXI Template*



AMBA and AXI Connect [Servo Motor]

- ***AXI Connect ➔ AXI Template***



➢ AXI Connect ➔ AXI Template

axi-fnd_v1_0_project - [c:/xilinx/managed_ip_project/managed_ip_project.tmp/axi-fnd_v1_0_project/axi-fnd_v1_0_project.xpr] - Vivado 2022.2

File Edit Flow Tools Reports Window Layout View Help

Tools:
- Create and Package New IP...
- Create Interface Definition...
- Enable Dynamic Function eXchange...
- Run Tcl Script...
- Property Editor Ctrl+J
- Associate ELF Files...
- Generate Memory Configuration File...
- Compile Simulation Libraries...
- Vivado Store...
- Custom Commands
- Launch Vitis IDE
- Language Templates
- Settings...

Flow Navigator: PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, SYNTHESIS, IMPLEMENTATION, PROGRAM AND DEBUG

Project Summary | axi-fnd_v1_0.v | Package IP - axi-fnd | axi-fnd_v1_0_S00_fnd_AXI.v * | mux4x1.v | fnddecoder.v | counter.v | clockdivider.v

Packaging Steps: Identification, Compatibility, File Groups, Customization Parameters, Ports and Interfaces, Addressing and Memory, Customization GUI, Review and Package

Identification

Vendor:	xilinx.com
Library:	user
Name:	axi-fnd
Version:	1.0
Display name:	axi-fnd_v1.0
Description:	My new AXI IP
Vendor display name:	
Company url:	
Root directory:	c:/Xilinx/ip_repo/axi-fnd_1_0
Xml file name:	c:/Xilinx/ip_repo/axi-fnd_1_0/component.xml

Categories: AXI_Peripheral

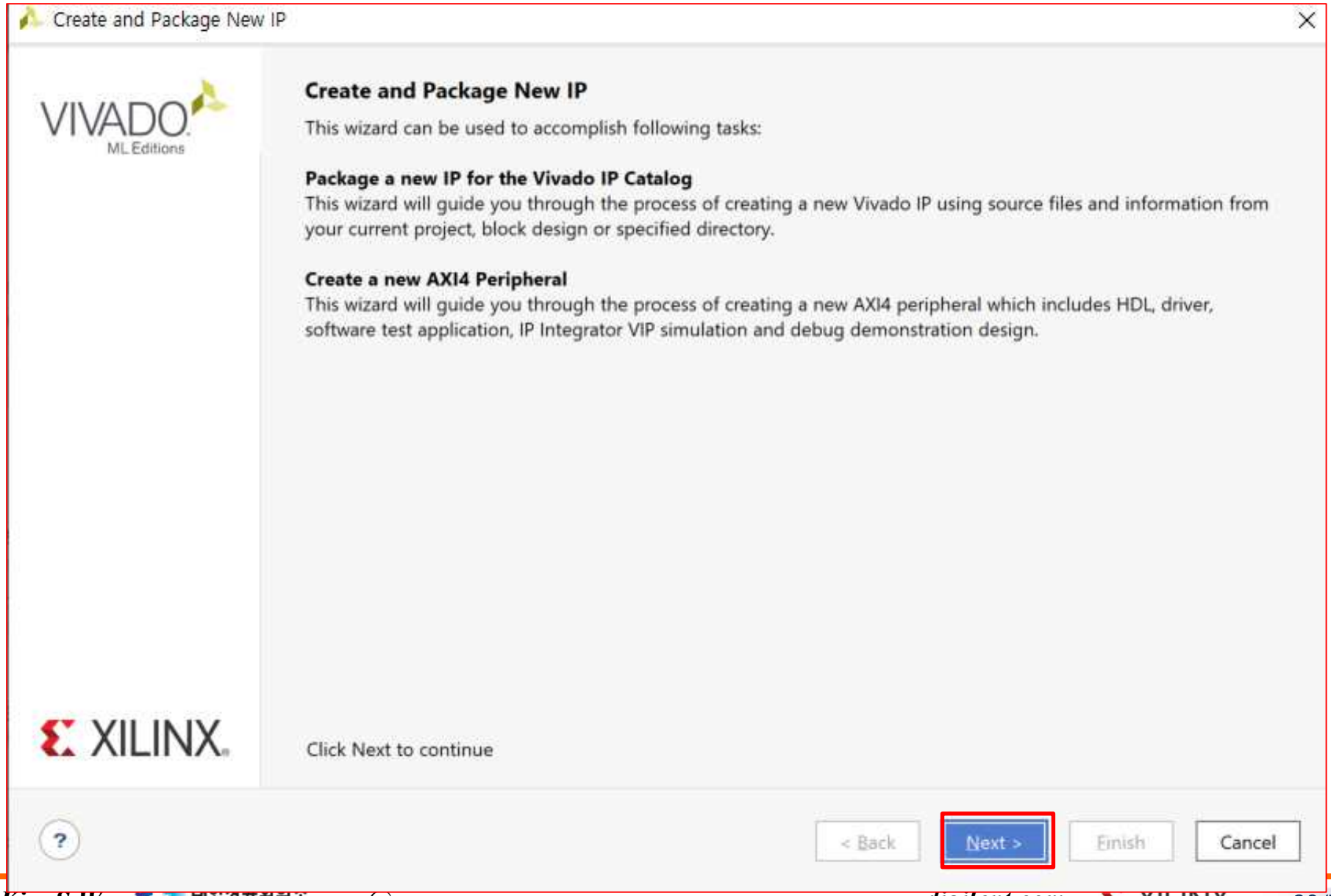
Source File Properties: axi-fnd_v1_0.v

Enabled	
Location:	c:/Xilinx/ip_repo/axi-fnd_1_0/hdl
Type:	Verilog
Library:	
Size:	3.3 KB
Modified:	Today at 10:08:32 AM
Read-only:	No

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constrai...	Status	Run Strategy	Report Stra
synth_1	constrs_1	Not started	Vivado Synthesis Defaults (Vivado Synthesis 2022)	Vivado Syn
impl_1	constrs_1	Not started	Vivado Implementation Defaults (Vivado Implementation 2022)	Vivado Imp

- ***AXI Connect ➔ AXI Template***



➢ AXI Connect ➔ AXI Template

Create and Package New IP

Create Peripheral, Package IP or Package a Block Design

Please select one of the following tasks.

Packaging Options

- (●) Package your current project
  Use the project as the source for creating a new IP Definition.
- ( ) Package a block design from the current project
  Choose a block design as the source for creating a new IP Definition.
- ( ) Package a specified directory
  Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

- ( ) Create a new AXI4 peripheral
  Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.

? < Back Next > Finish Cancel

➢ *AXI Connect ➔ AXI Template*

Create and Package New IP

VIVADO ML Editions

New IP Creation

The following pieces of information will be gathered:

- Identification information based on top module name
- Family compatibility based on part in the project
- File(s) from Synthesis and Simulation file sets
- Ports from the file containing the top module
- Parameters from the file containing the top module
- Bus Interfaces based on port names
- Address Spaces and Memory Maps based on inferred bus interfaces

Following file will be created on disk along with corresponding customization files:
f:/xilnx/2022/amba_axi_connect_test/amba_axi_connect_test.srcs/component.xml

XILINX

Click Finish to continue

? < Back Next > Finish Cancel

➢ *AXI Connect ➔ AXI Template*

MyIP 생성

PROJECT MANAGER - fnd_axi_v1_0_project

Sources

- Design Sources (2)
  - fnd_axi_v1_0 (fnd_axi_v1_0.v) (6)
  - IP-XACT (1)
- Constraints
- Simulation Sources (1)
- Utility Sources

Hierarchy | Libraries | Compile Order

Properties

Select an object to see properties

Project Summary | Package IP - fnd_axi

Packaging Steps

- Identification
- Compatibility
- File Groups
- Customization Parameters
- Ports and Interfaces
- Addressing and Memory
- Customization GUI
- Review and Package

Identification

Vendor:	xilinx.com
Library:	user
Name:	fnd_axi
Version:	1.0
Display name:	fnd_axi_v1.0
Description:	My new AXI IP
Vendor display name:	
Company url:	
Root directory:	c:/Xilinx/ip_repo/fnd_axi_1_0
Xml file name:	c:/Xilinx/ip_repo/fnd_axi_1_0/component.xml

Categories

AXI_Peripheral

AMBA and AXI Connect Example

Servo Motor Module

[Create Project ➔ Create Block Design]

➢ AXI Connect ➔ AXI Template

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

PROJECT MANAGER - project_1

PROJECT MANAGER

Settings **Settings 클릭**

Add Sources

Language Templates

IP Catalog

IP INTEGRATOR

Create Block Design

Open Block Design

Generate Block Design

SIMULATION

Run Simulation

RTL ANALYSIS

Open Elaborated Design

SYNTHESIS

Run Synthesis

Open Synthesized Design

IMPLEMENTATION

Run Implementation

Open Implemented Design

PROGRAM AND DEBUG

Generate Bitstream

Open Hardware Manager

Design Sources

Constraints

Simulation Sources

sim_1

Utility Sources

Hierarchy Libraries Compile Order

Properties

Select an object to see properties

Project Summary

Overview | Dashboard

Settings Edit

Project name:	project_1
Project location:	C:/Users/parkj/test231013/project_1
Product family:	Artix-7
Project part:	Basys3 (xc7a35tcpg236-1)
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Board Part

Display name:	Basys3
Board part name:	digilentinc.com:basys3:part0:1.2
Board revision:	C.0
Connectors:	No connections
Repository path:	E:/Xilinx/Vivado/2022.2/data/boards/board_files
URL:	https://digilent.com/reference/programmable-logic/basys-3/start
Board overview:	Basys3

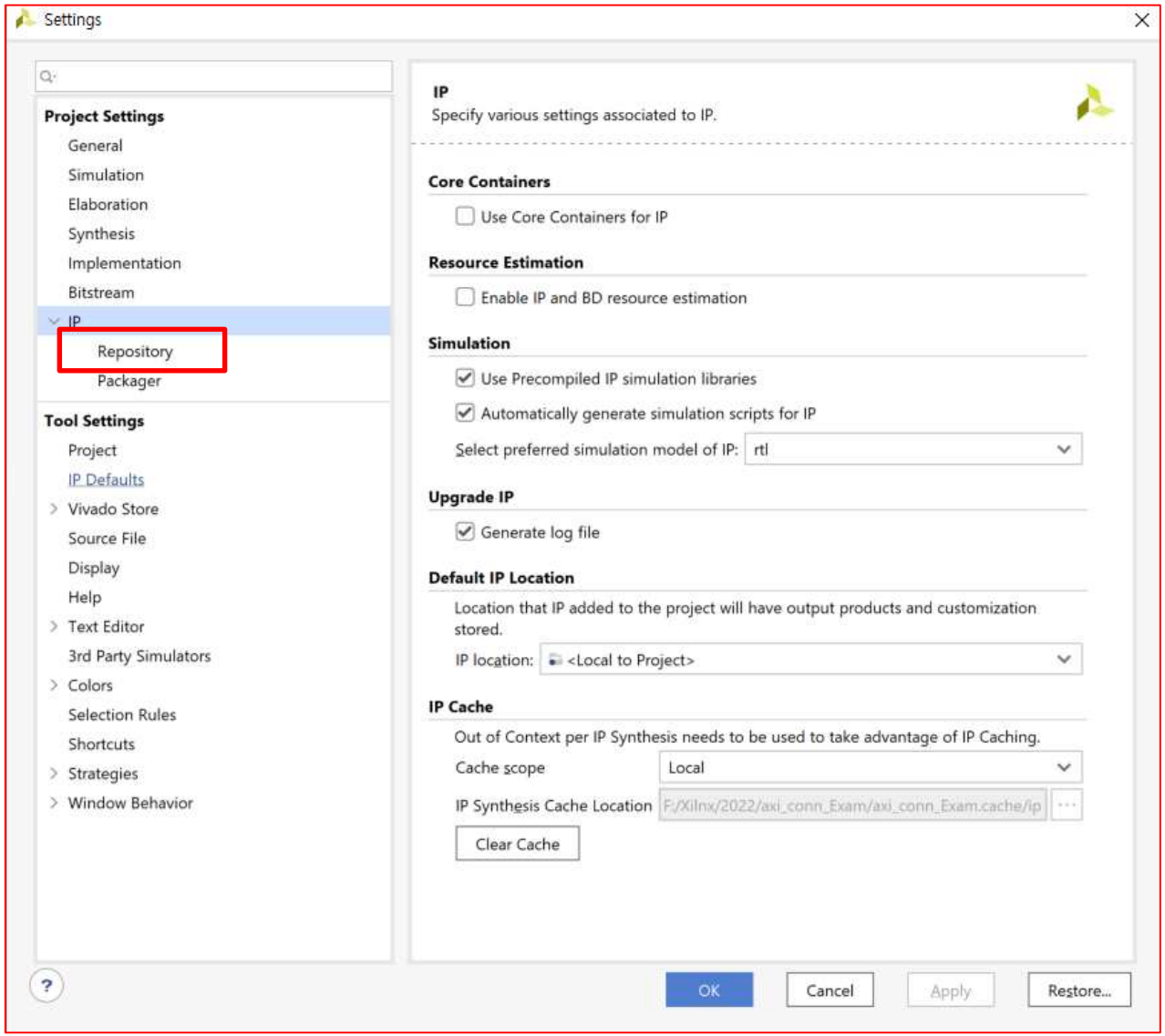
Synthesis		Implementation	
Status:	Not started	Status:	Not started
Messages:	No errors or warnings	Messages:	No errors or warnings

Tcl Console Messages Log Reports Design Runs

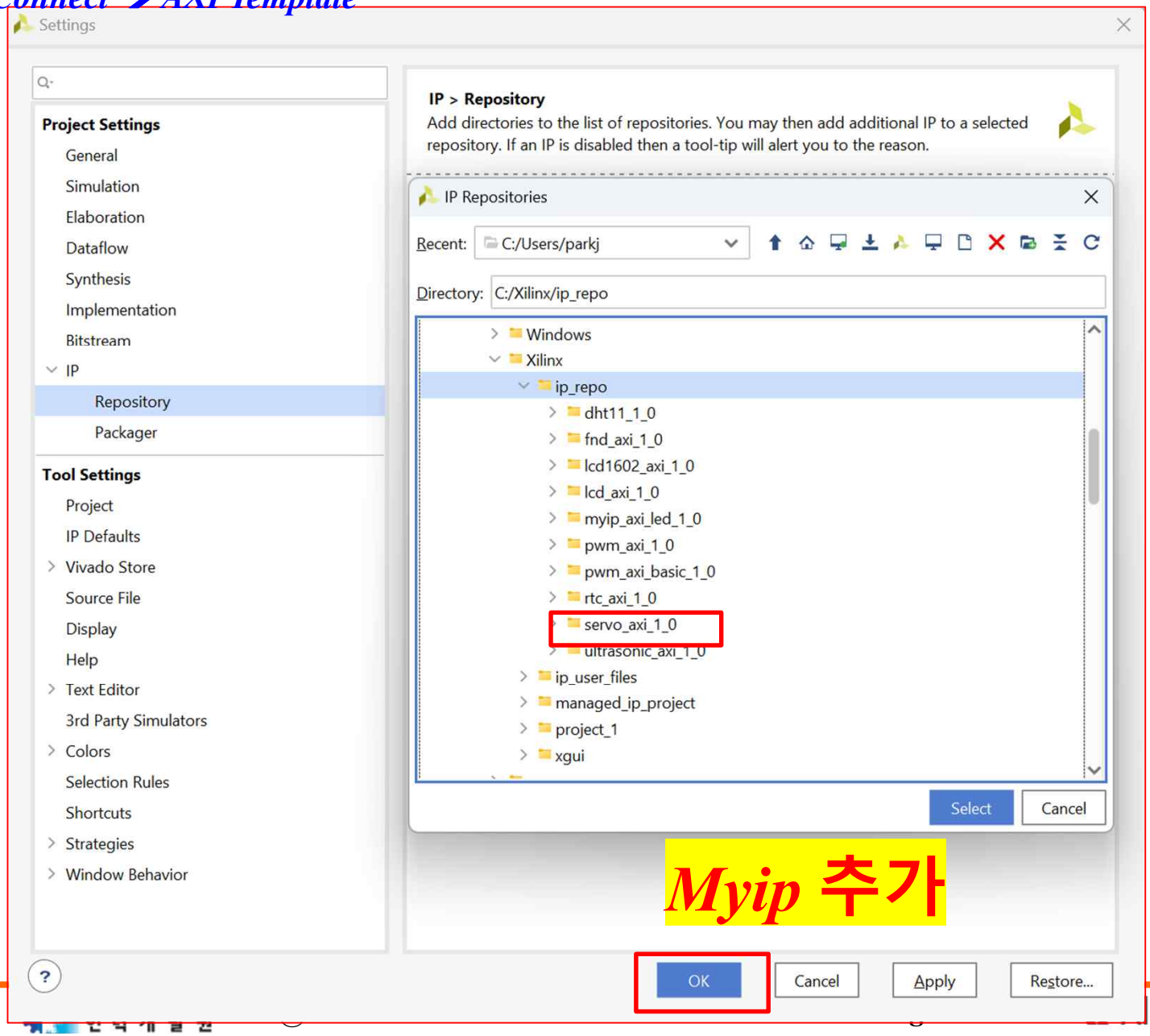
Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started																			Vivado Synthesis Defaults (Viva
impl_1	constrs_1	Not started																			Vivado Implementation Default

➢ AXI Connect ➔ AXI Template

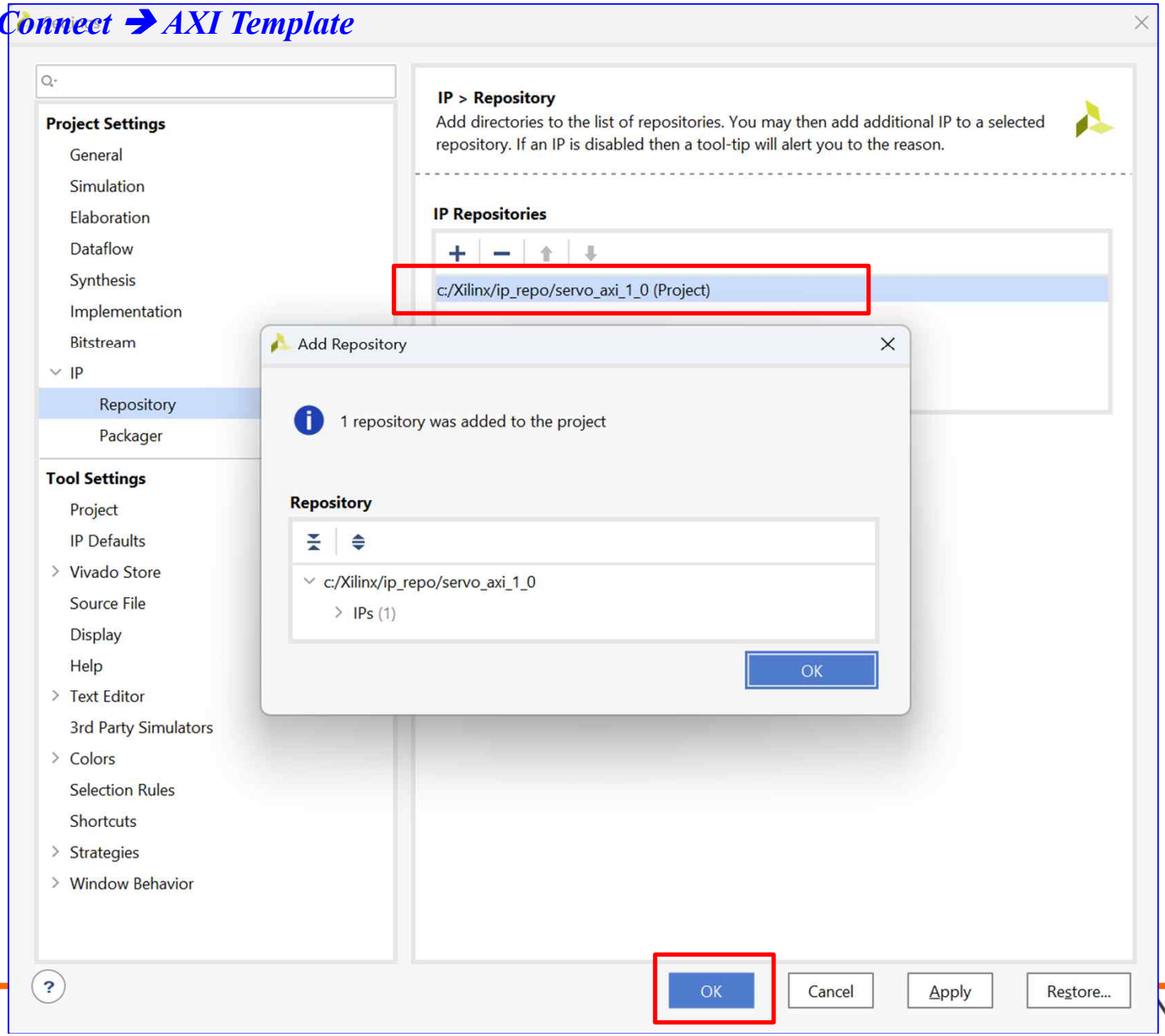
Repository



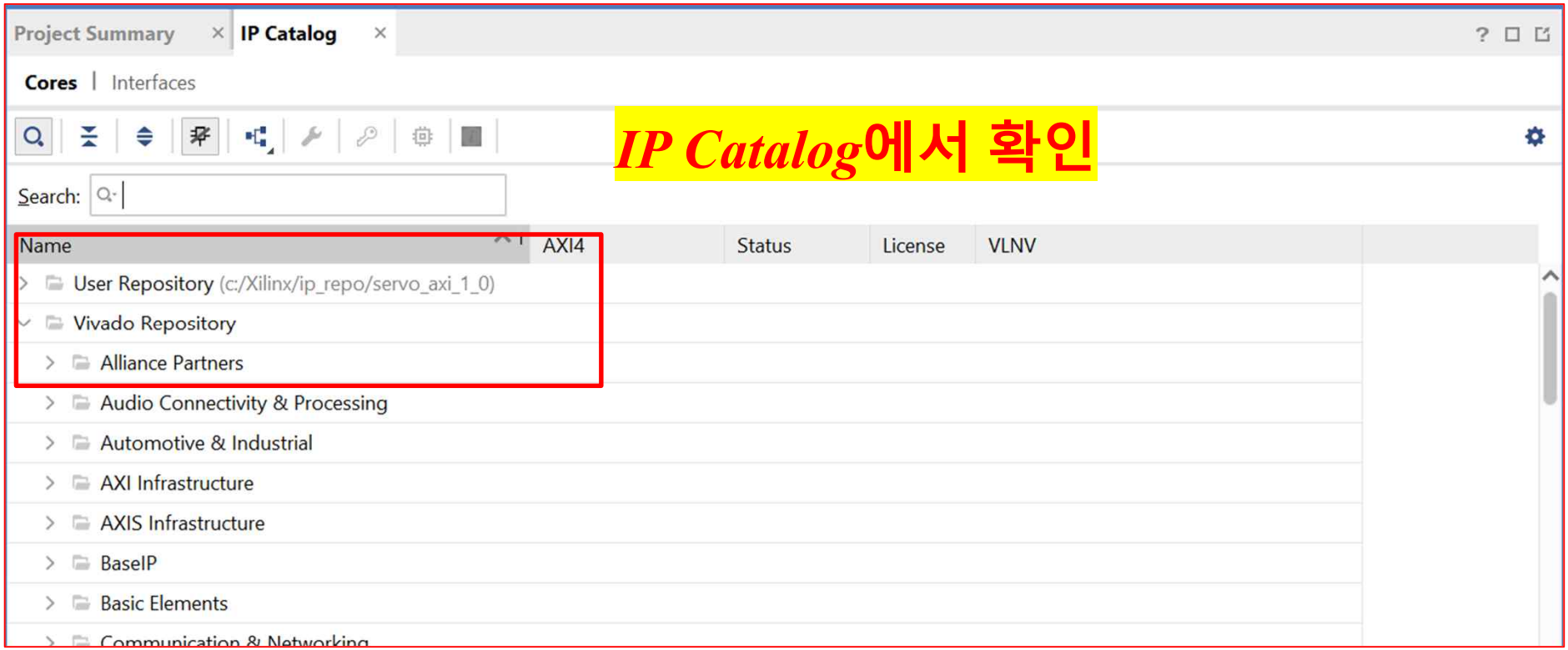
- *AXI Connect ➔ AXI Template*



➢ AXI Connect ➔ AXI Template



- *AXI Connect ➔ AXI Template*



➢ *AXI Connect ➔ AXI Template*

File Edit Flow Tools Reports Window Layout View Help Quick Access Ready

Default Layout

Flow Navigator

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

PROJECT MANAGER - project_1

Sources

- Design Sources
- Constraints
- Simulation Sources
  - sim_1
- Utility Sources

Hierarchy | Libraries | Compile Order

Create Block Design

Repository Properties

User Repository

Path :	c:/Xilinx/ip_repo/axi_fnd_1_0
Number of IPs :	1
Number of interfaces :	0

General | IPs

Project Summary | IP Catalog

Overview | Dashboard

Settings Edit

Project name:	project_1
Project location:	C:/Users/parkj/test231013/project_1
Product family:	Artix-7
Project part:	Basys3 (xc7a35tcpg236-1)
Top module name:	Not defined
Target language:	Verilog
Simulator language:	Mixed

Board Part

Display name:	Basys3
Board part name:	digilentinc.com:basys3:part0:1.2
Board revision:	C.0
Connectors:	No connections
Repository path:	E:/Xilinx/Vivado/2022.2/data/boards/board_files
URL:	https://digilent.com/reference/programmable-logic/basys-3/start
Board overview:	Basys3

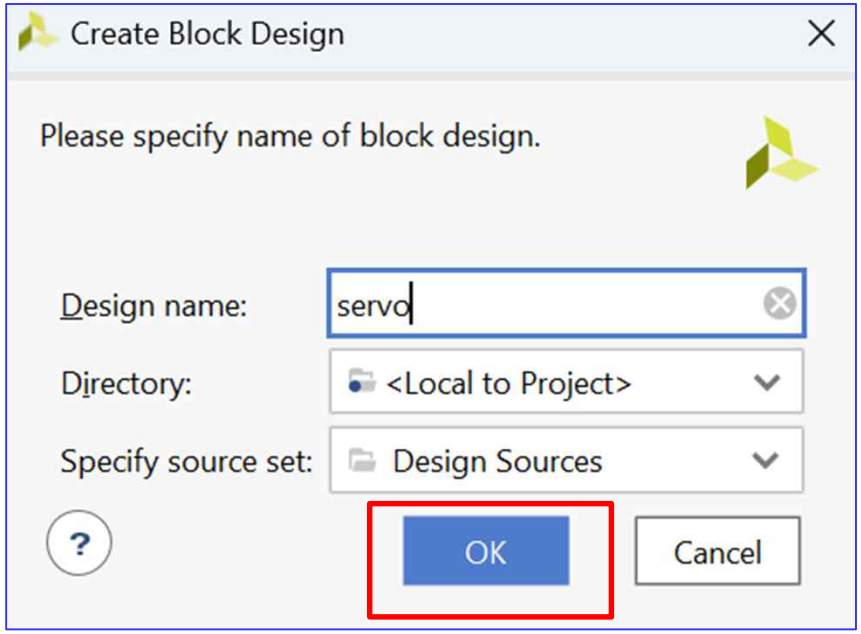
Synthesis		Implementation	
Status:	Not started	Status:	Not started
Messages:	No errors or warnings	Messages:	No errors or warnings

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started																			Vivado Synthesis Defaults (Viva
impl_1	constrs_1	Not started																			Vivado Implementation Default

Create and add an IP subsystem to the project

- AXI Connect ➔ AXI Template

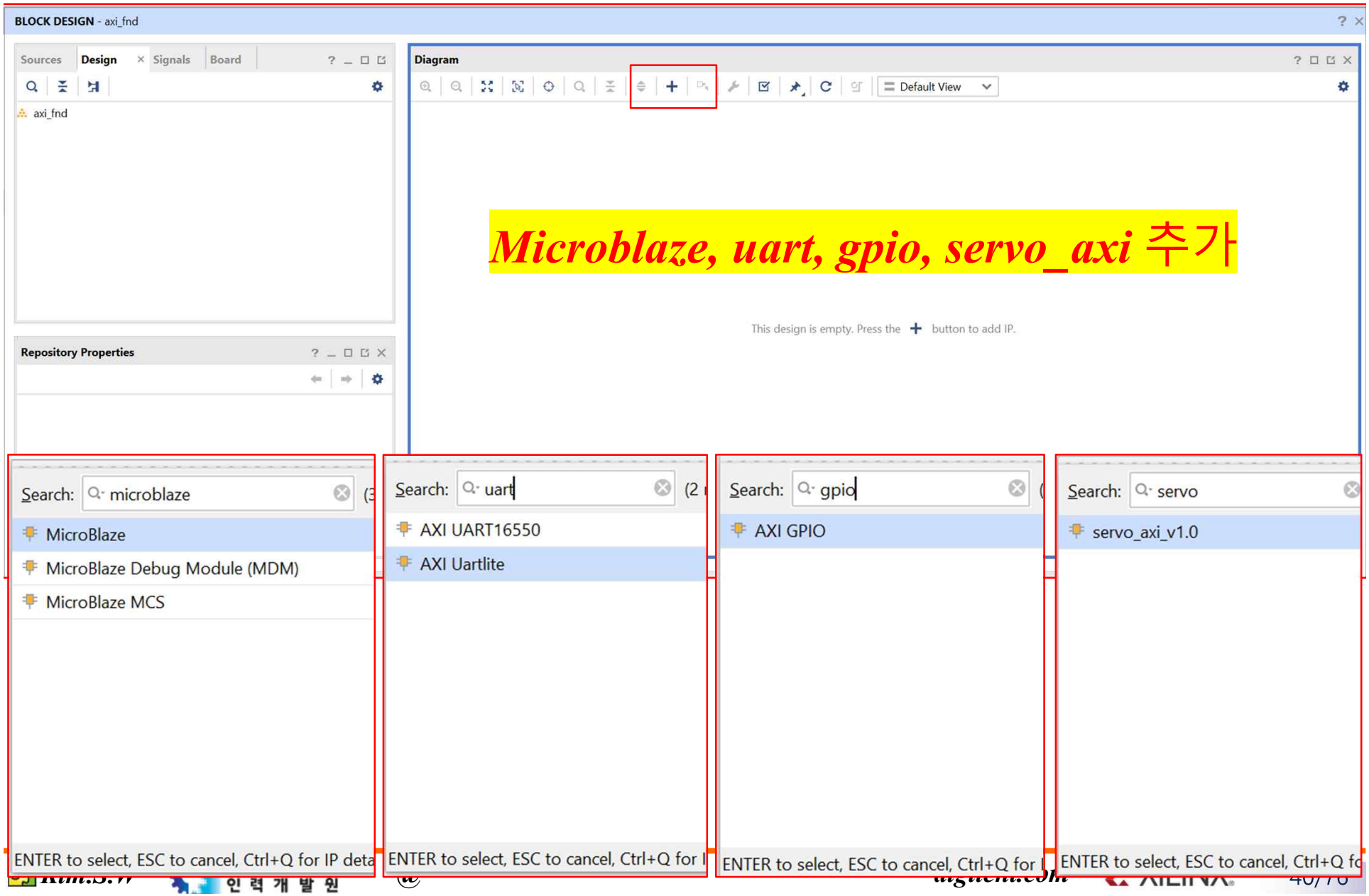


이름 설정 후 OK

AMBA and AXI Connect [Servo Motor]

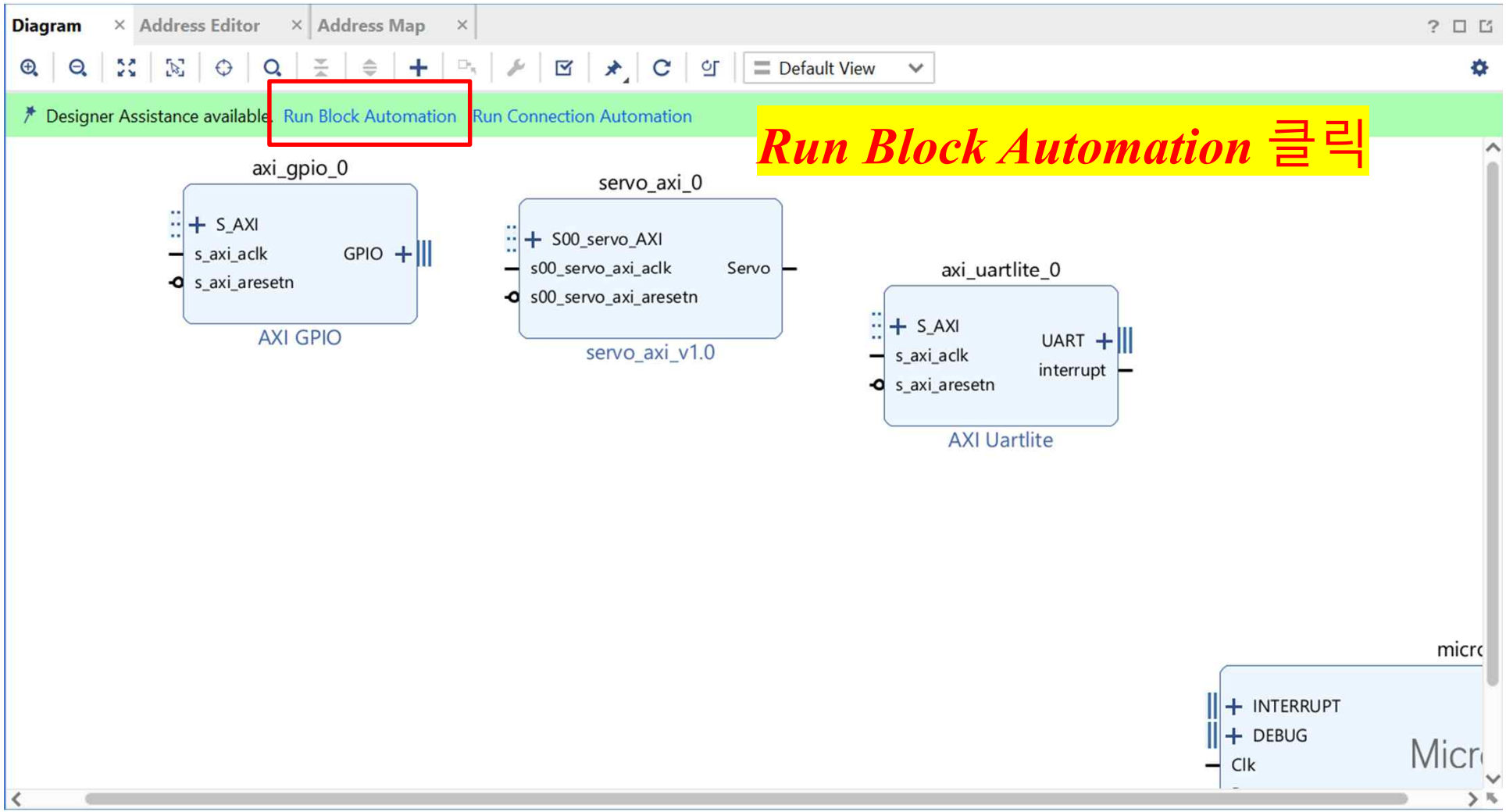
Axi_Servo_Exam.xpr

➢ *AXI Connect ➔ AXI Template*



AMBA and AXI Connect [Servo Motor]

➢ ***AXI Connect ➔ AXI Template***



AMBA and AXI Connect [Servo Motor]

Axi_Servo_Exam.xpr

Run Block Automation

➢ *AXI Connect ➔ AXI Template*

Automatically make connections in your design by checking the boxes of the blocks to connect. Select a block on the left to display its configuration options on the right.

All Automation (1 out of 1 selected)

microblaze_0

Description

MicroBlaze connection automation generates local memory of selected size, and caches can be configured. MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset are added and connected as needed. A preset MicroBlaze configuration can also be selected.

Information about the options can be found in the tooltips.

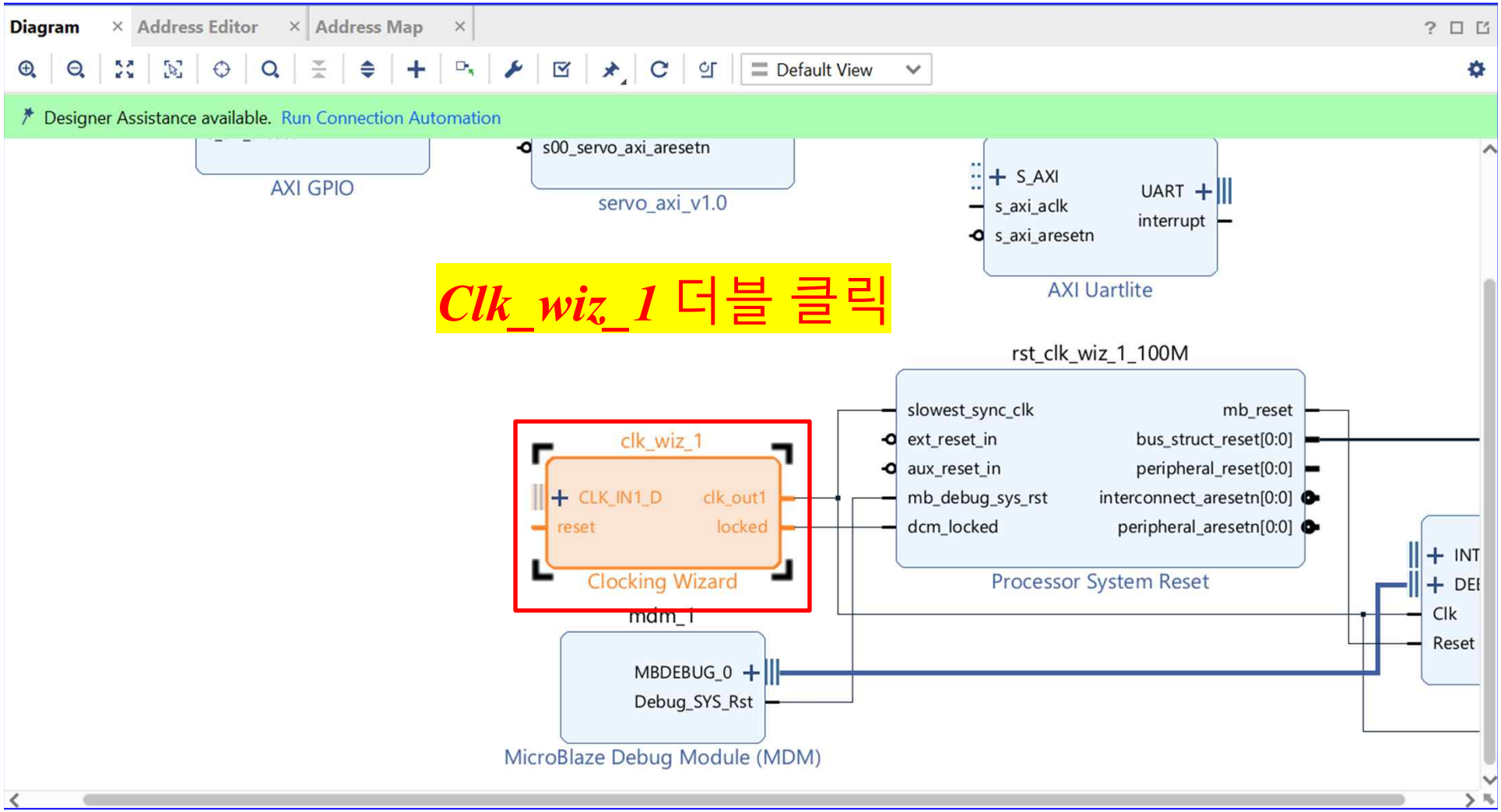
Options

Preset	None
Local Memory	32KB
Local Memory ECC	None
Cache Configuration	None
Debug Module	Debug Only
Peripheral AXI Port	Enabled
Interrupt Controller	
Clock Connection	New Clocking Wizard

OK Cancel

AMBA and AXI Connect [Servo Motor]

➢ AXI Connect ➔ AXI Template



➢ AXI Connect ➔ AXI Template

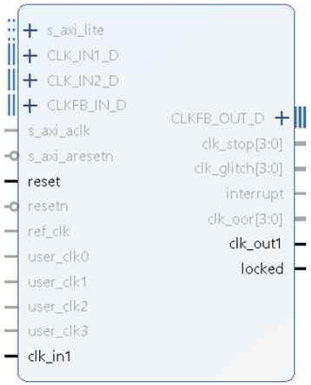
Re-customize IP

Clocking Wizard (6.0)

Documentation | IP Location

IP Symbol | Resource

Show disabled ports



Component Name clk_wiz_1

Board | Clocking Options | Output Clocks | MMCM Settings | Summary

Clock Monitor

☐ Enable Clock Monitoring

Primitive

◉ MMCM ○ PLL

Clocking Features

☑ Frequency Synthesis ☐ Minimize Power
☑ Phase Alignment ☐ Spread Spectrum
☐ Dynamic Reconfig ☐ Dynamic Phase Shift
☐ Safe Clock Startup

Jitter Optimization

◉ Balanced
○ Minimize Output Jitter
○ Maximize Input Jitter filtering

Dynamic Reconfig Interface Options

◉ AXI4Lite ○ DRP ☐ Phase Duty Cycle Config ☐ Write DRP registers

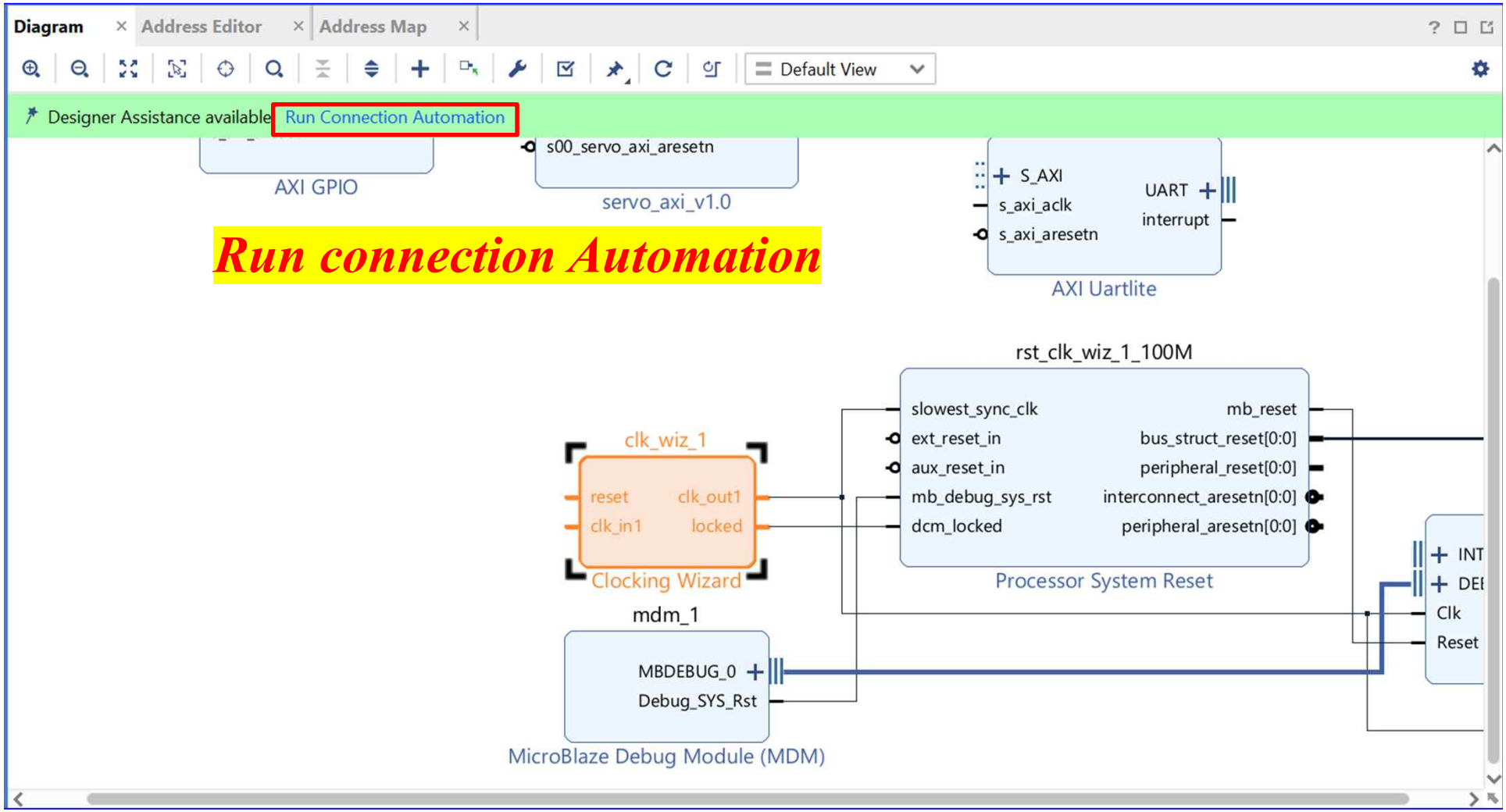
Input Clock Information

Input Clock	Port Name	Input Frequency(MHz)			Jitter Options	Input Jitter	Source
Primary	clk_in1	AUTO	100.000	10.000 - 800.000	UI	0.010	Single ended clock capable p
Secondary	clk_in2	AUTO	100.000	60.000 - 120.000		0.010	Single ended clock capable p

Single ended clock 변경 후 OK

OK | Cancel

- ***AXI Connect ➔ AXI Template***



AMBA and AXI Connect [Servo Motor]

Axi_Servo_Exam.xpr

Run Connection Automation

➢ AXI Connect ➔ AXI Template

Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.

- All Automation (8 out of 8 selected)
  - axi_gpio_0
    - GPIO
    - S_AXI
  - axi_uartlite_0
    - S_AXI
    - UART
  - clk_wiz_1
    - clk_in1
    - reset
  - rst_clk_wiz_1_100M
    - ext_reset_in
  - servo_axi_0
    - S00_servo_AXI

Select an interface pin on the left panel to view its options

All Automation 선택 후 OK

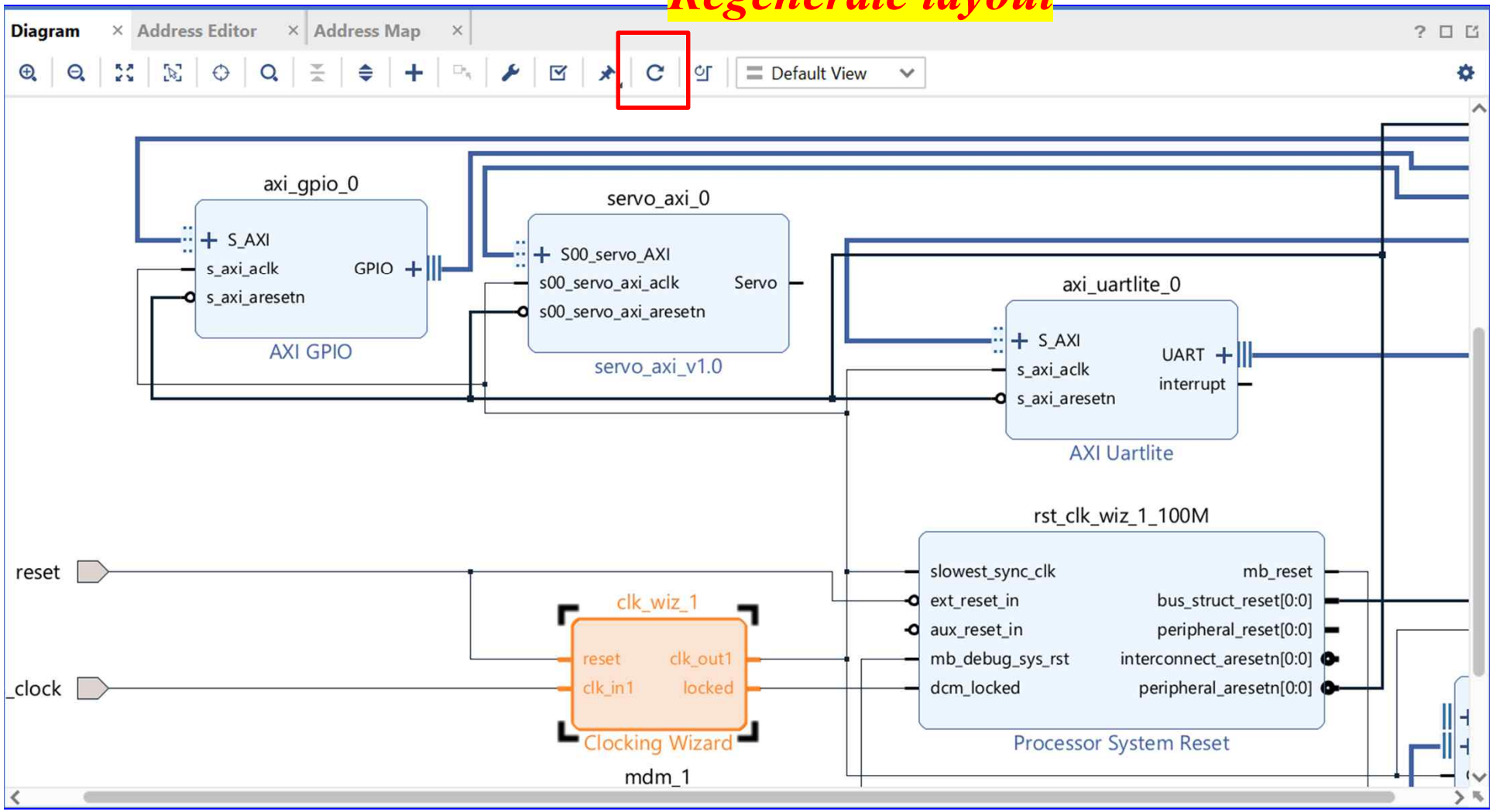
OK

Cancel

AMBA and AXI Connect [Servo Motor]

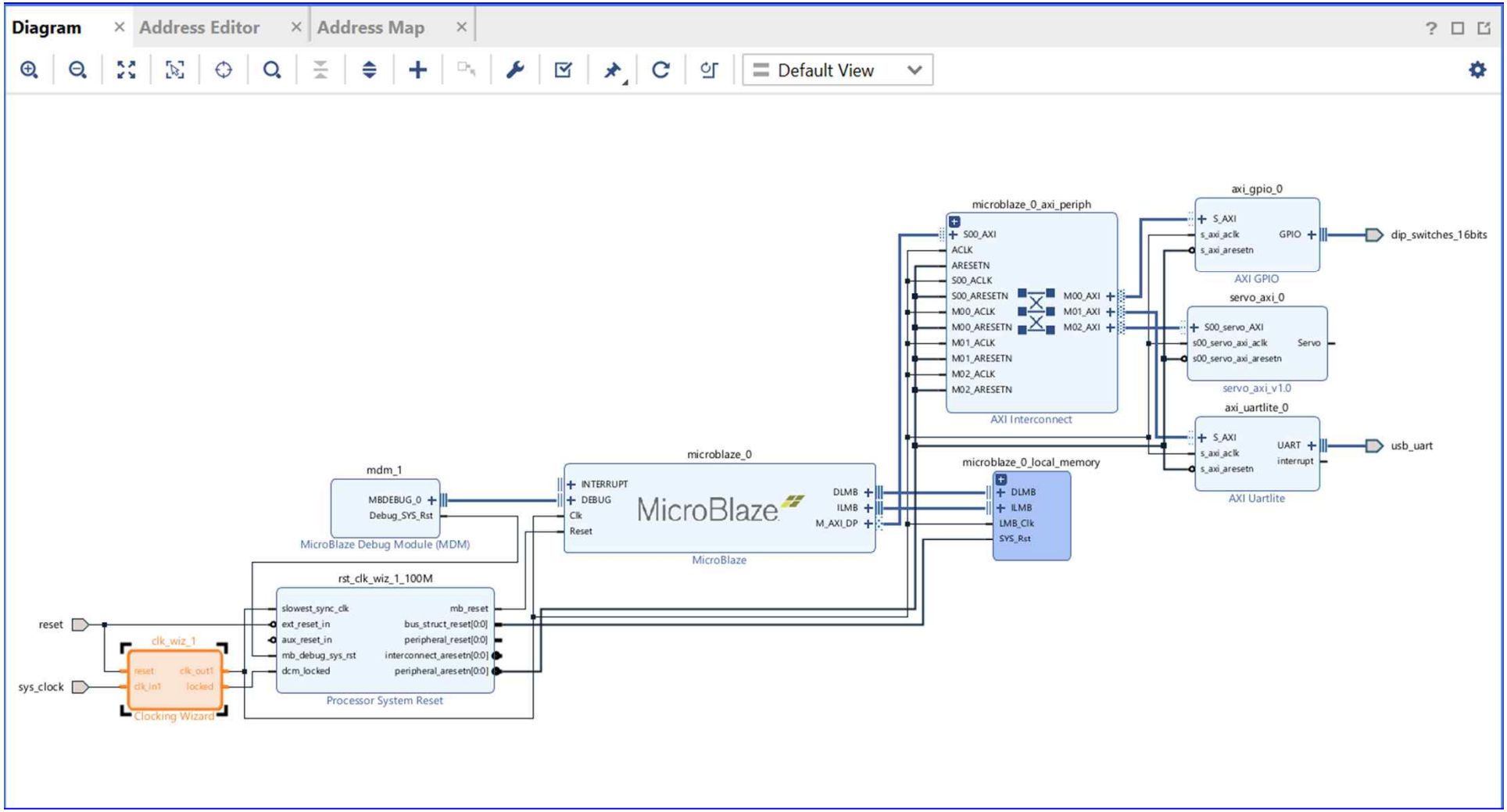
➢ AXI Connect ➔ AXI Template

Regenerate layout

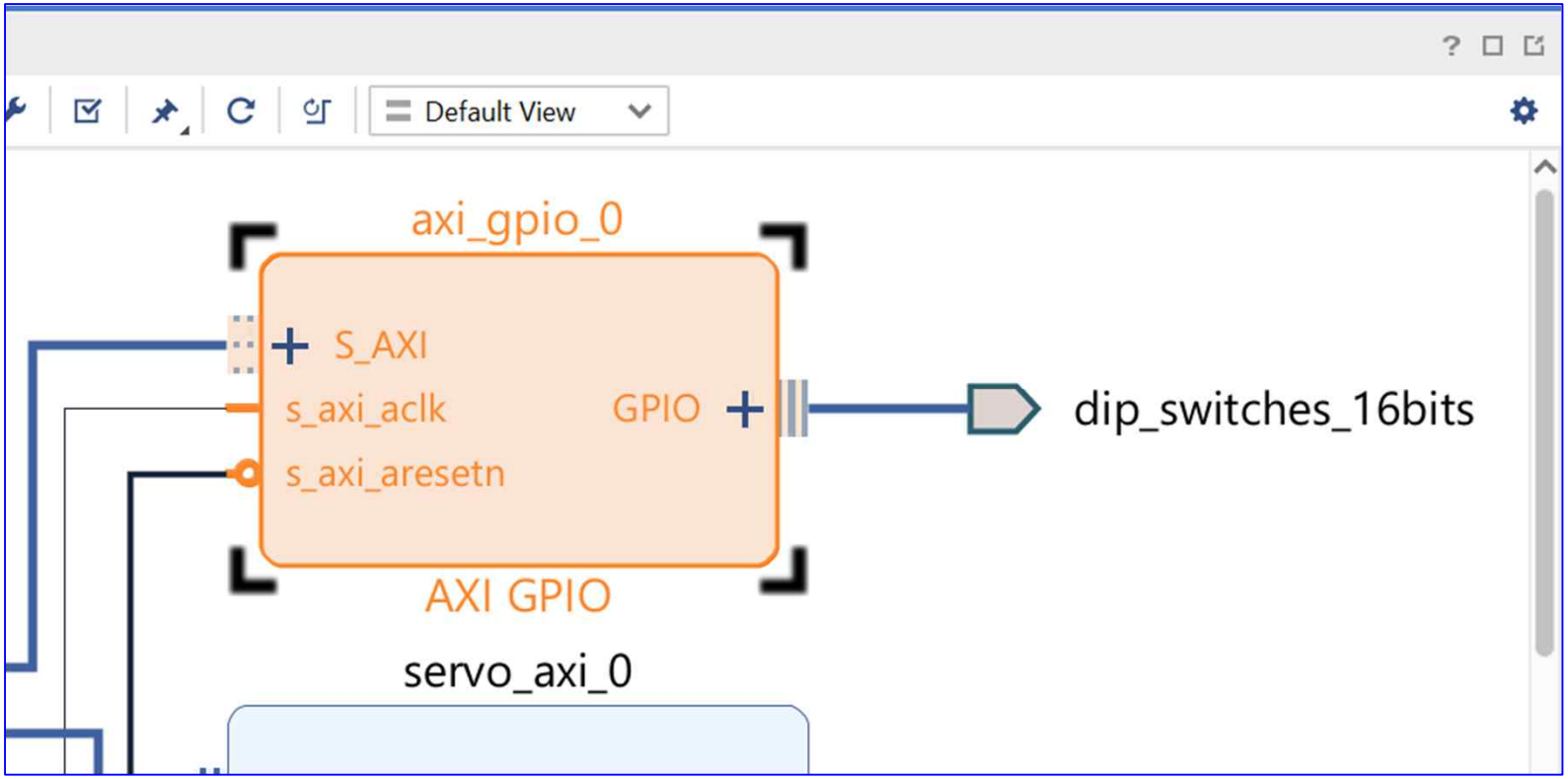


AMBA and AXI Connect [Servo Motor]

➢ AXI Connect ➔ AXI Template

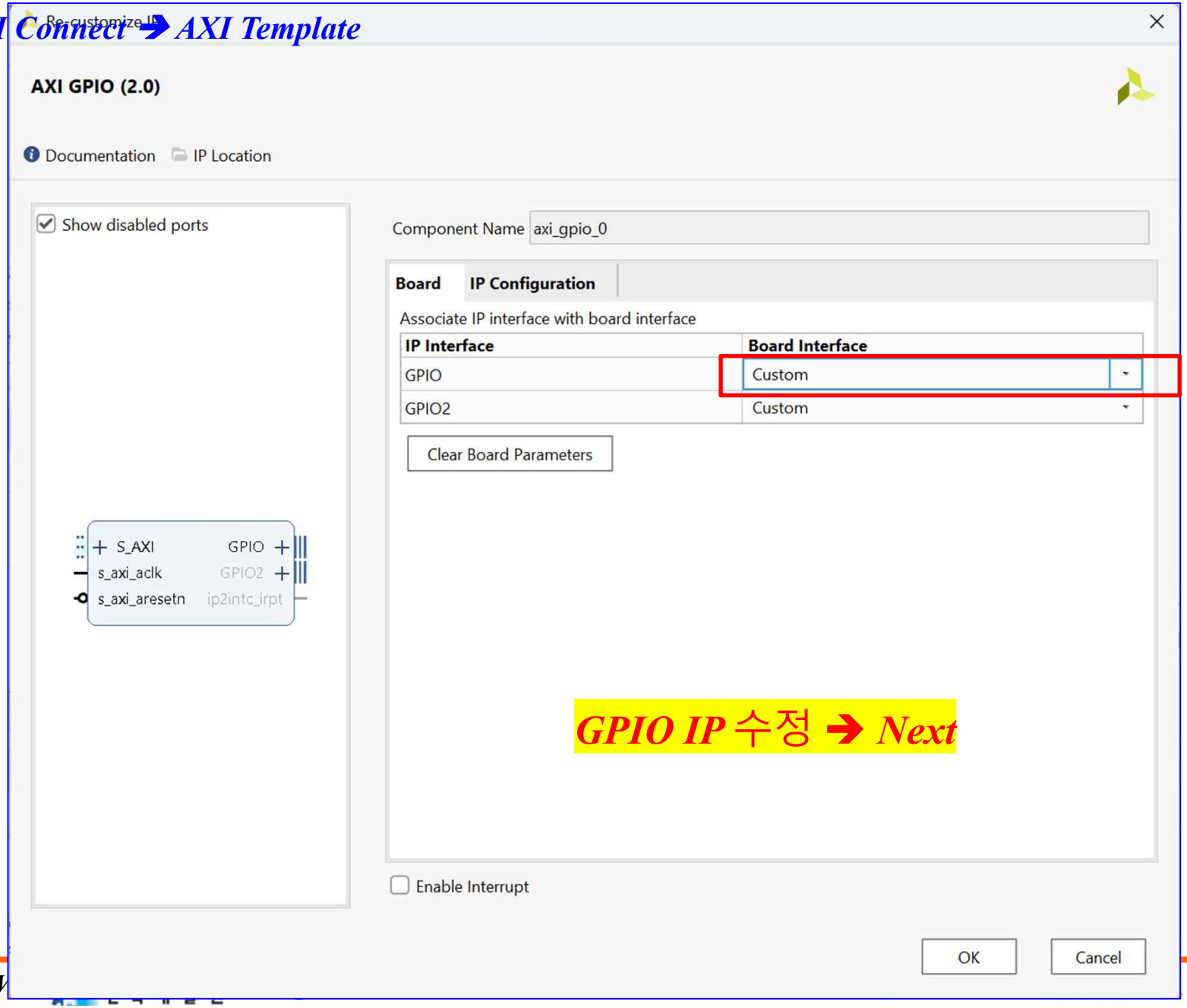


- *AXI Connect ➔ AXI Template*



GPIO IP 수정 ➔ Next

➢ *AXI Connect ➔ AXI Template*



GPIO IP 수정 ➔ Next

AMBA and AXI Connect [Servo Motor]

- AXI Connect ➔ AXI Template

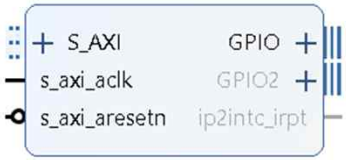
GPIO IP 수정 ➔ Next

Re-customize IP

AXI GPIO (2.0)

Documentation | IP Location

Show disabled ports



Component Name: axi_gpio_0

Board | IP Configuration

GPIO

- [x] All Inputs
- [ ] All Outputs

input, Width 4

GPIO Width	4	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

- [ ] Enable Dual Channel

GPIO 2

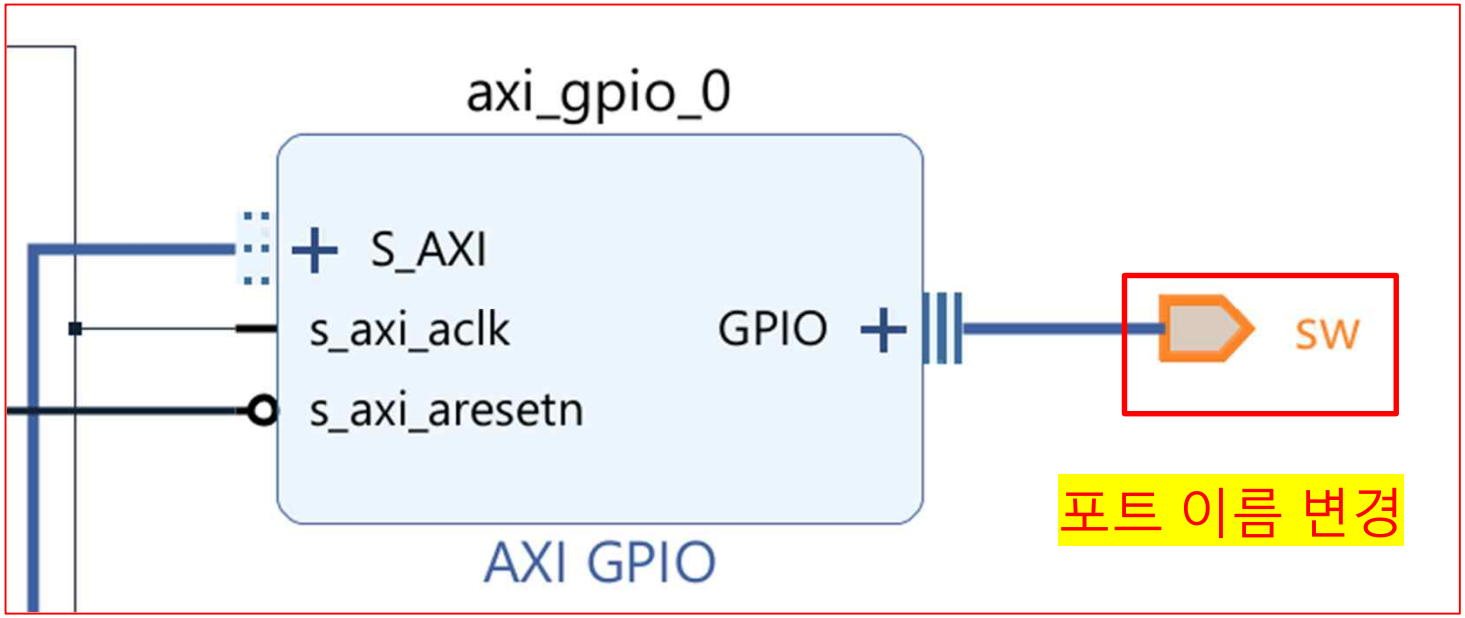
- [ ] All Inputs
- [ ] All Outputs

GPIO Width	32	[1 - 32]
Default Output Value	0x00000000	[0x00000000,0xFFFFFFFF]
Default Tri State Value	0xFFFFFFFF	[0x00000000,0xFFFFFFFF]

- [ ] Enable Interrupt

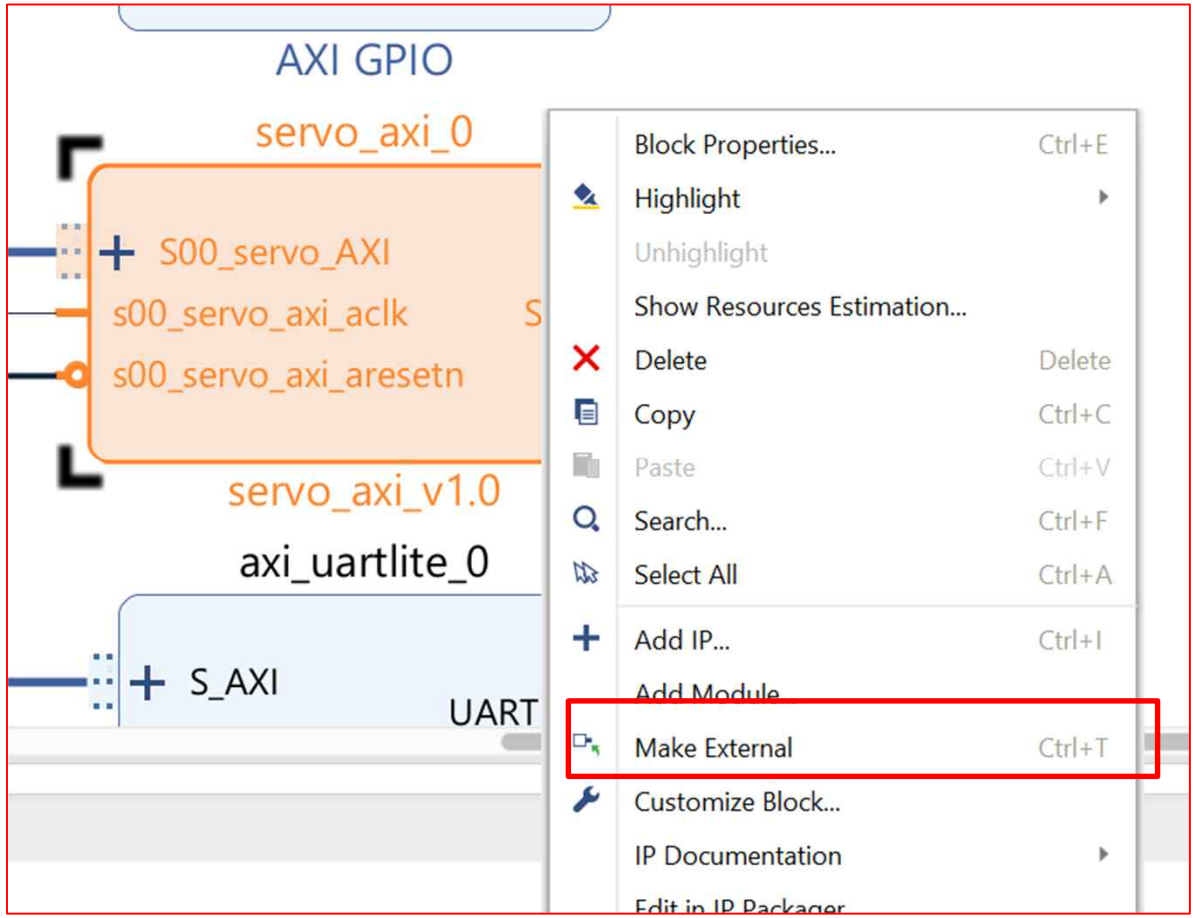
OK | Cancel

➢ AXI Connect ➔ AXI Template



AMBA and AXI Connect [Servo Motor]

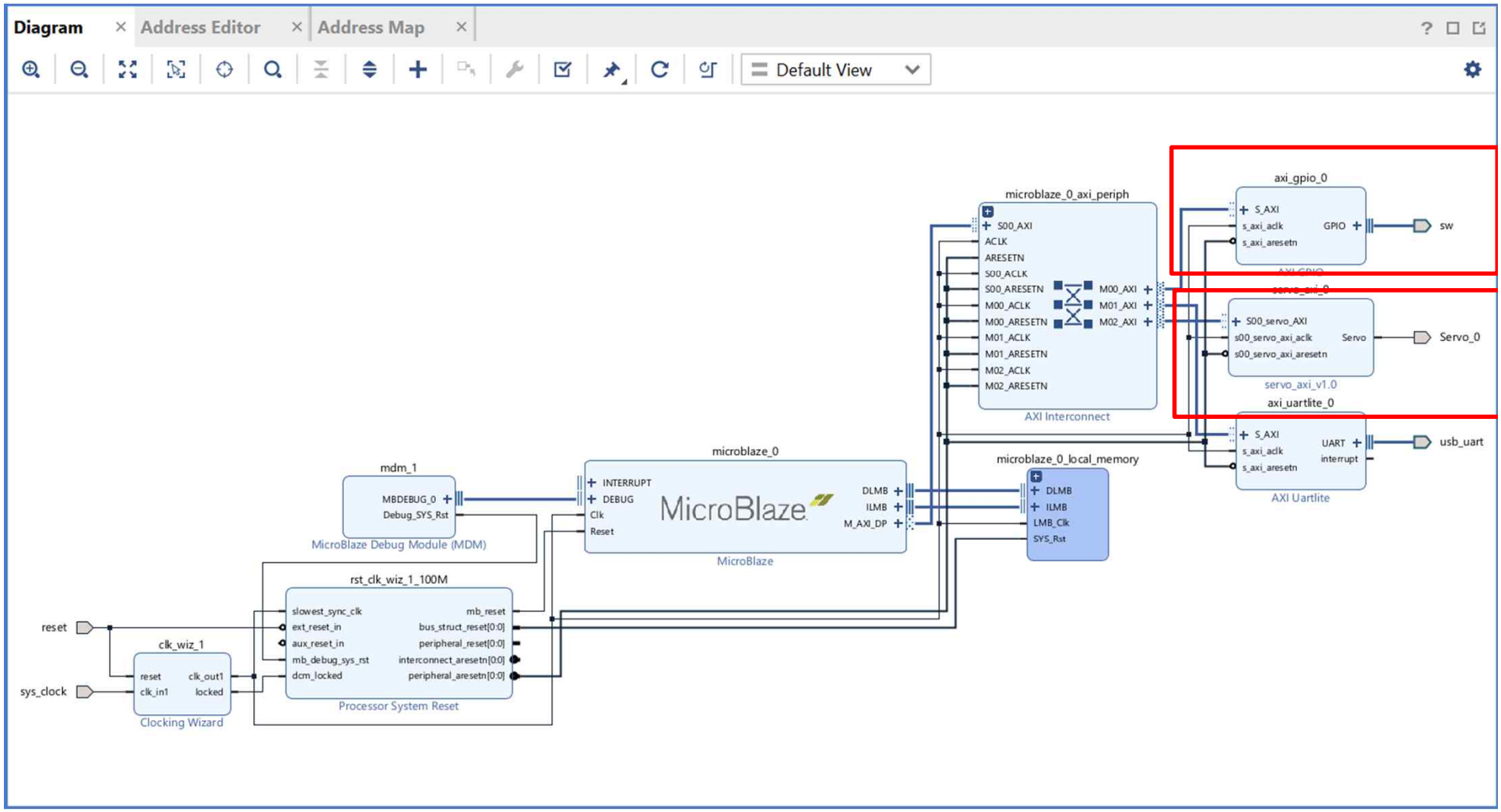
- AXI Connect ➔ AXI Template ➔ servo_axi_0 ➔ Make External



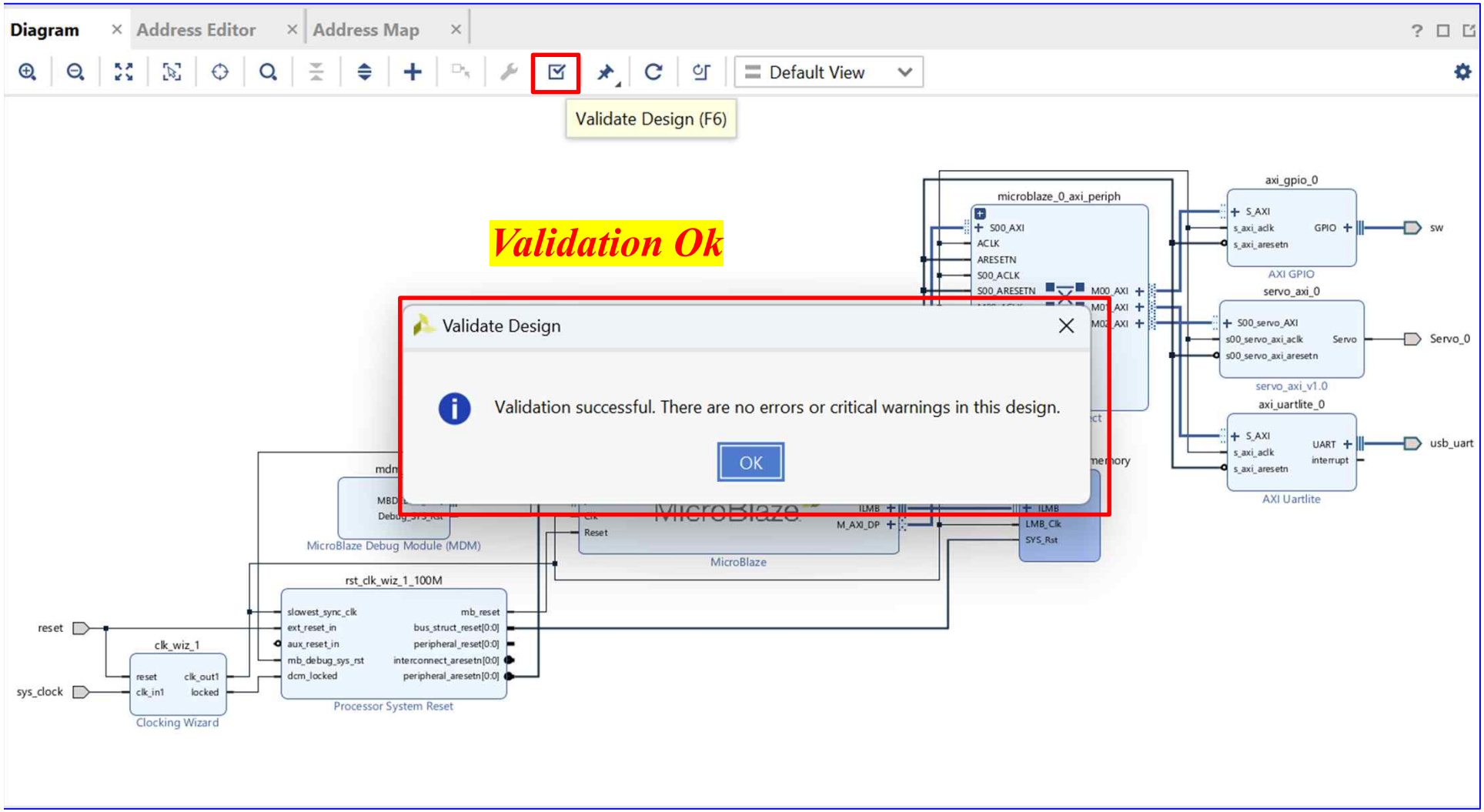
Servo IP make External

AMBA and AXI Connect [Servo Motor]

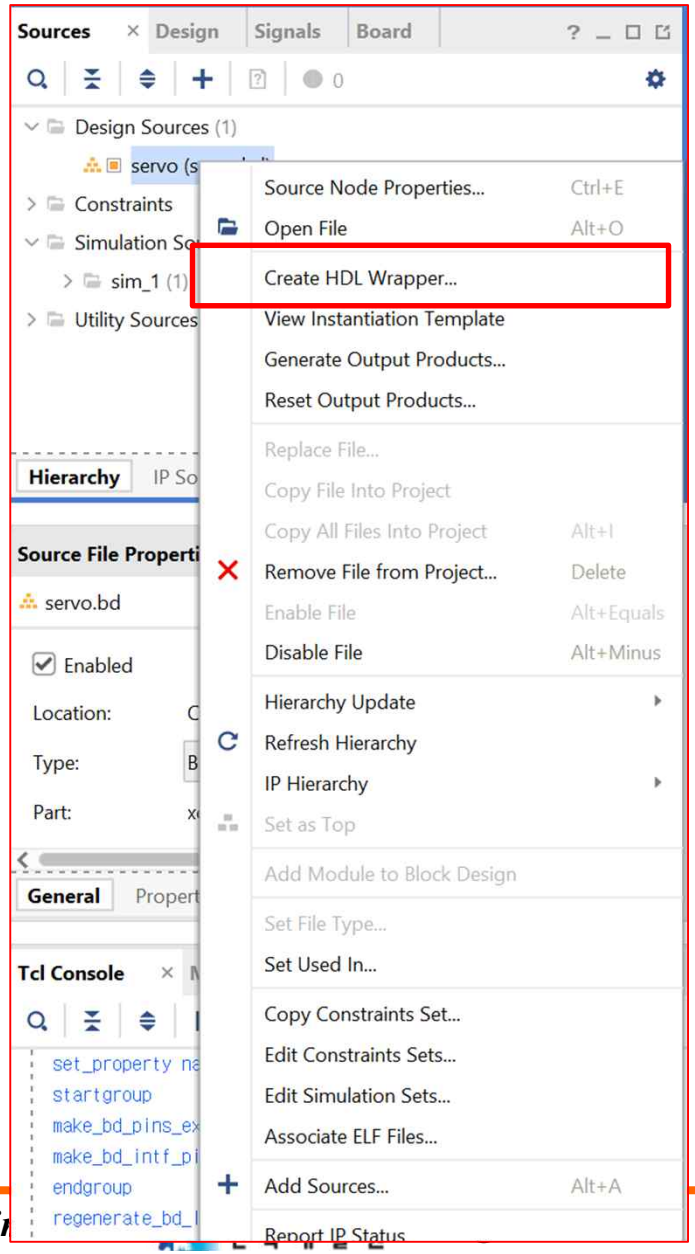
➢ AXI Connect ➔ AXI Template

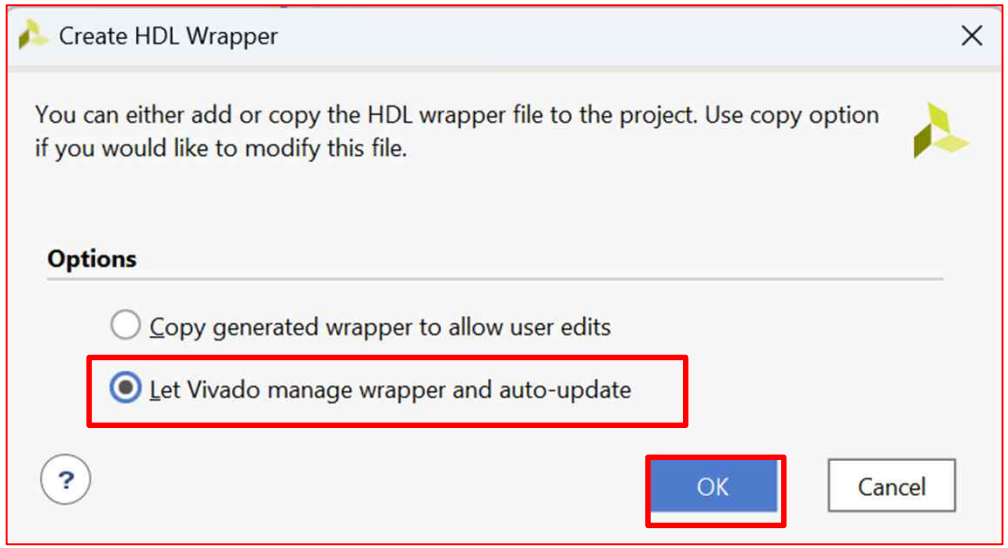


- *AXI Connect ➔ AXI Template ➔ Validate*

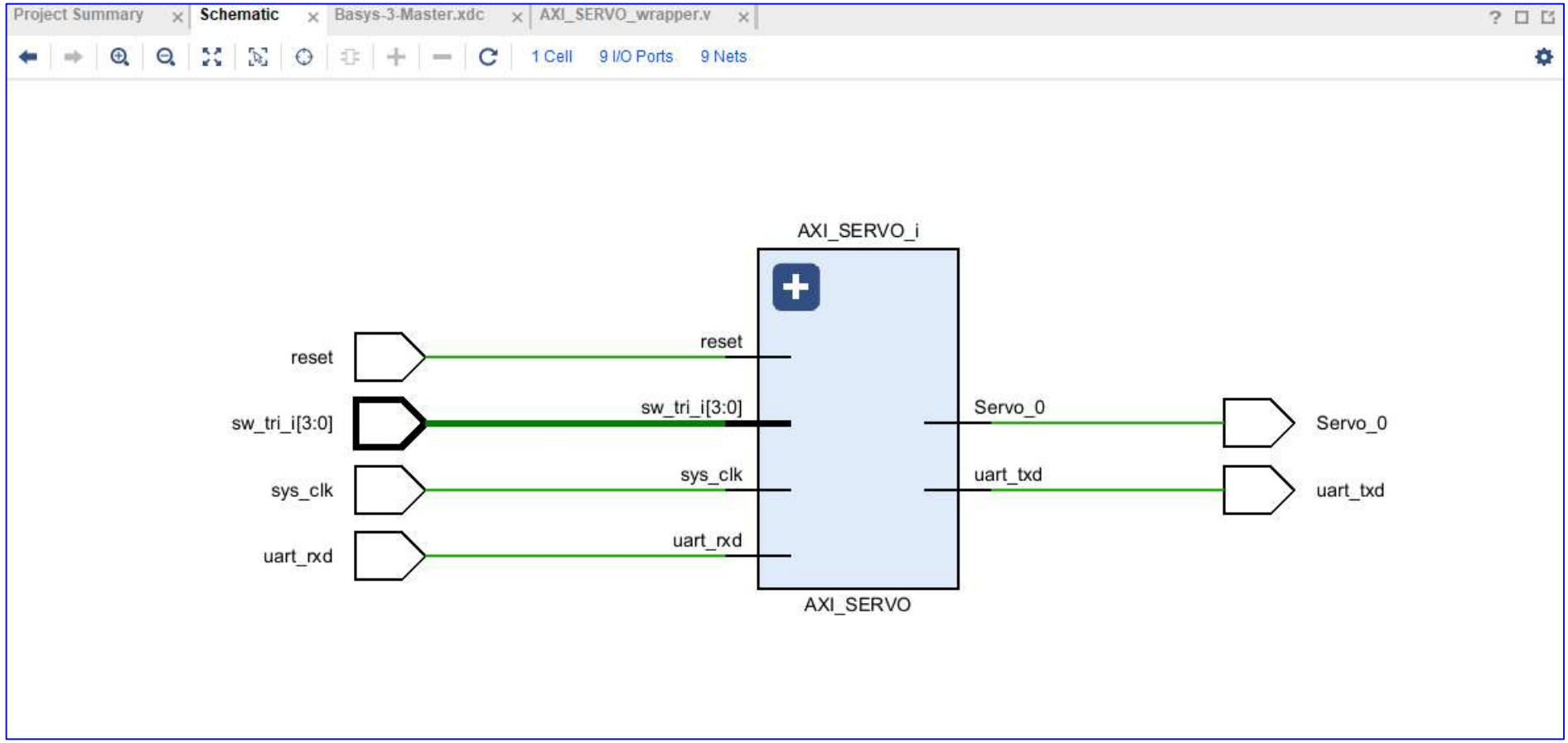


- *AXI Connect ➔ AXI Template ➔ Create HDL Wrapper*

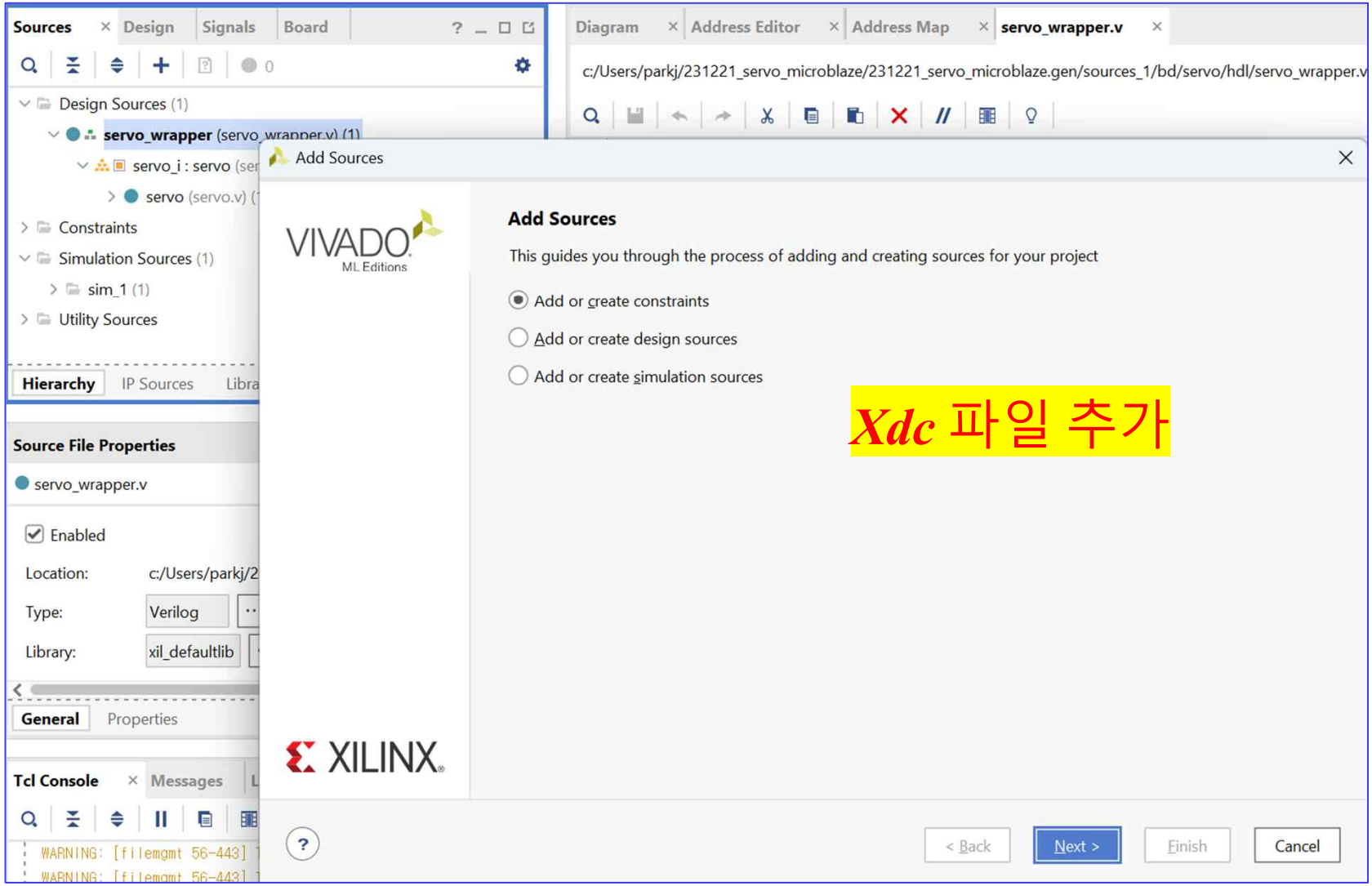




- *AXI Connect ➔ AXI Template ➔ RTL ➔ Schematic*



➢ AXI Connect ➔ AXI Template



- AXI Connect ➔ AXI Template
  - XDC Constraints

Basys-3-Master.xdc

```
1. set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports sys_clock]
2. create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports sys_clock]

3. ## Switches
4. set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {reset}]
5. set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {sw_tri_i[0]}]
6. set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {sw_tri_i[1]}]
7. set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {sw_tri_i[2]}]
8. set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {sw_tri_i[3]}]

9. ##Pmod Header JC
10. set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports {Servo_0}];#Sch
    name = JC1

11. ##USB-RS232 Interface
12. set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports usb_uart_rxd]
13. set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports usb_uart_txd]

14. ## Configuration options, can be used for all designs
15. set_property CONFIG_VOLTAGE 3.3 [current_design]
16. set_property CFGBVS VCCO [current_design]

17. ## SPI configuration mode options for QSPI boot, can be used for all designs
18. set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
19. set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
20. set_property CONFIG_MODE SPIx4 [current_design]


21. ## Configuration options, can be used for all designs
22. set_property CONFIG_VOLTAGE 3.3 [current_design]
23. set_property CFGBVS VCCO [current_design]

24. ## SPI configuration mode options for QSPI boot, can be used for all designs
25. set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
26. set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
27. set_property CONFIG_MODE SPIx4 [current_design]
```

➢ *AXI Connect ➔ AXI Template*

- *Create Block Design*
- *Blaze Uart & FND 추가 및 연결*
- *sys_clock & reset 설정*
- *GPIO, Servo Make External*
- *XDC Constraints 설정*
- *Bitstream*
- *Export Hardwar*

AMBA and AXI Connect [Servo Motor]

✓ Application SW(Vitis)

➢ *AXI Connect ➔ AXI Template*

File Edit Flow Tools Reports Window Layout View Help Quick Access

Flow Navigator

PROJECT MANAGER

IP INTEGRATOR

Create Block Desig

Open Block Desig

Generate Block De

SIMULATION

RTL ANALYSIS

Open Elaborated

Report Metho

Report DRC

Schematic

Open Dataflow Design

SYNTHESIS

Run Synthesis

Open Synthesized Design

Validate Design F6

Create and Package New IP...

Create Interface Definition...

Enable Dynamic Function eXchange...

Run Tcl Script...

Property Editor Ctrl+J

Associate ELF Files...

Generate Memory Configuration File...

Compile Simulation Libraries...

Vivado Store...

Custom Commands

Launch Vitis IDE

Language Templates

Settings...

Board

_1_wrapper.v) (1)

Compile Order

Diagram

Source File Properties

design_1_wrapper.v

Enabled

Location: c:/Users/parkj/project_30/project_30.gen/sources_1/bc

Type: Verilog

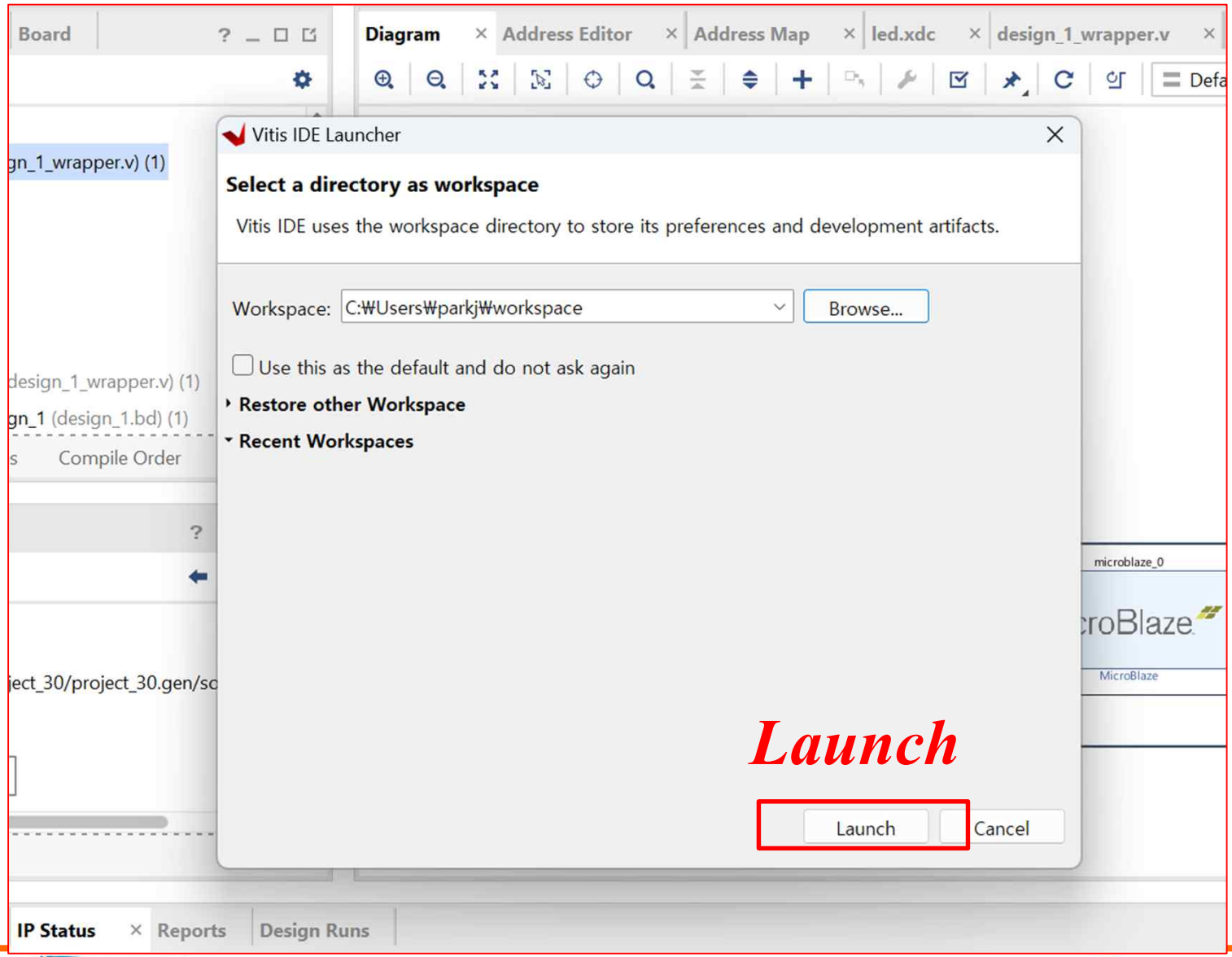
Library: xil_defaultlib

reset

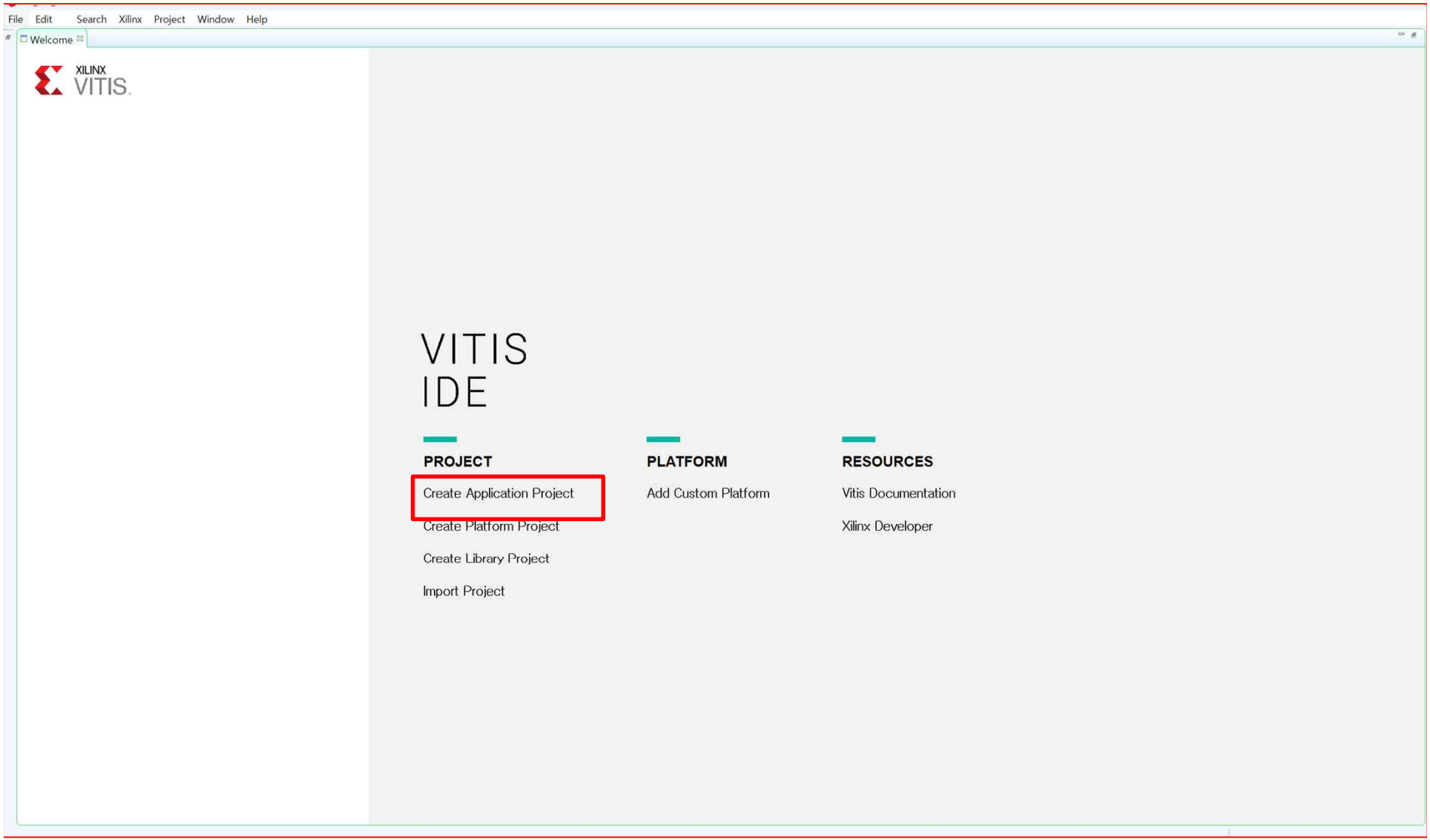
sys_clock

Launch Vitis 클릭

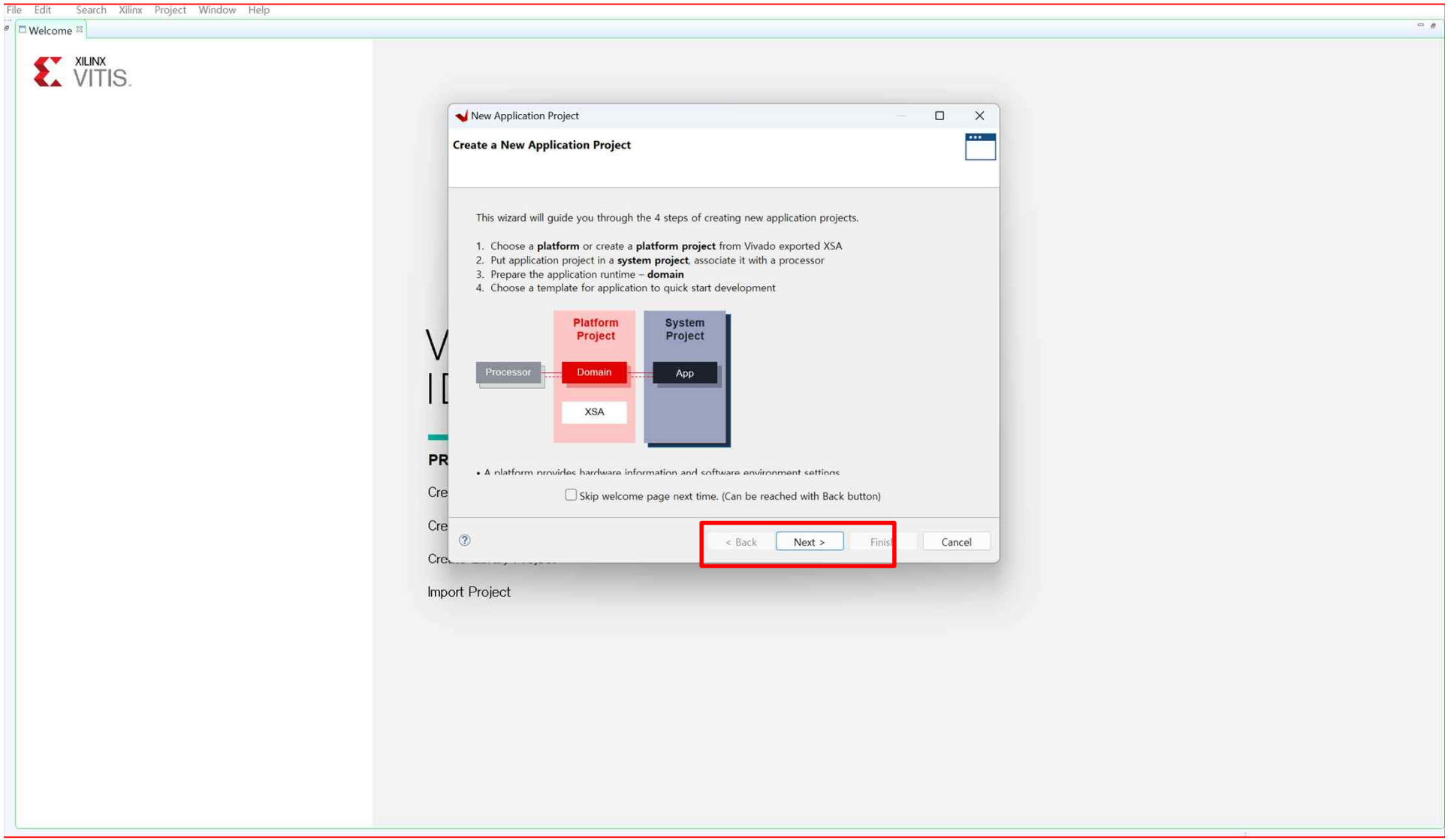
➢ AXI Connect ➔ AXI Template



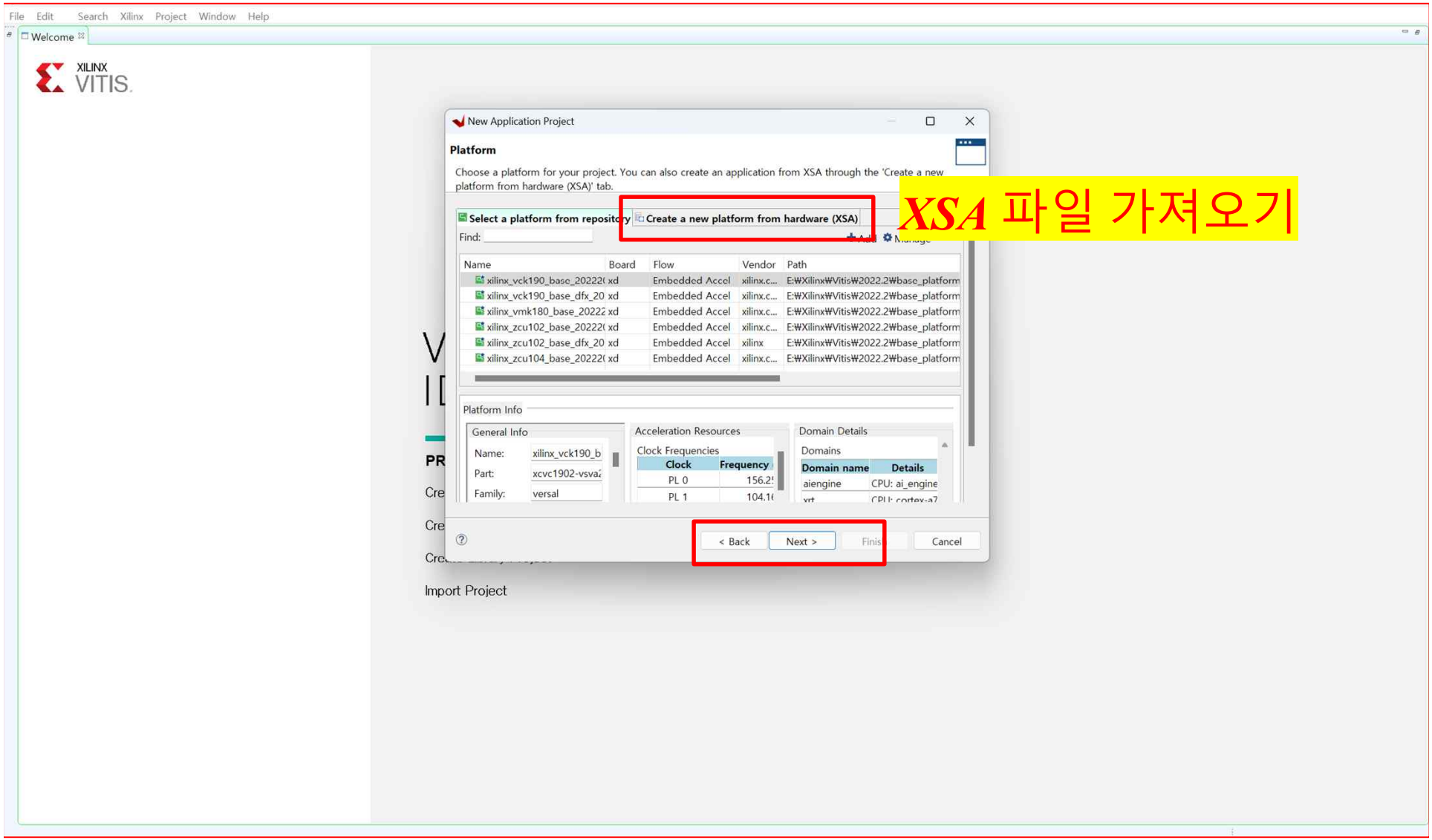
- *AXI Connect ➔ AXI Template*



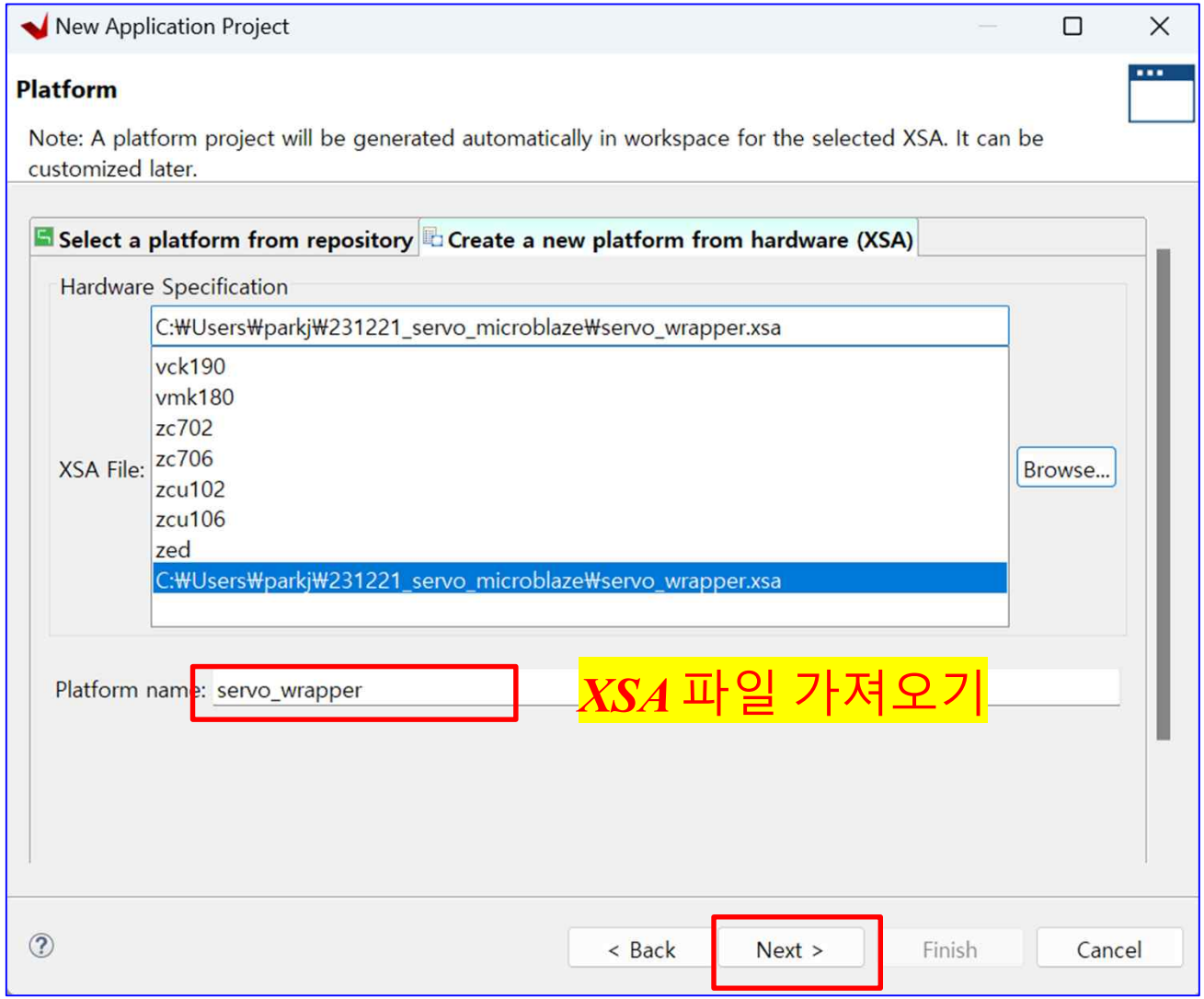
➢ AXI Connect ➔ AXI Template



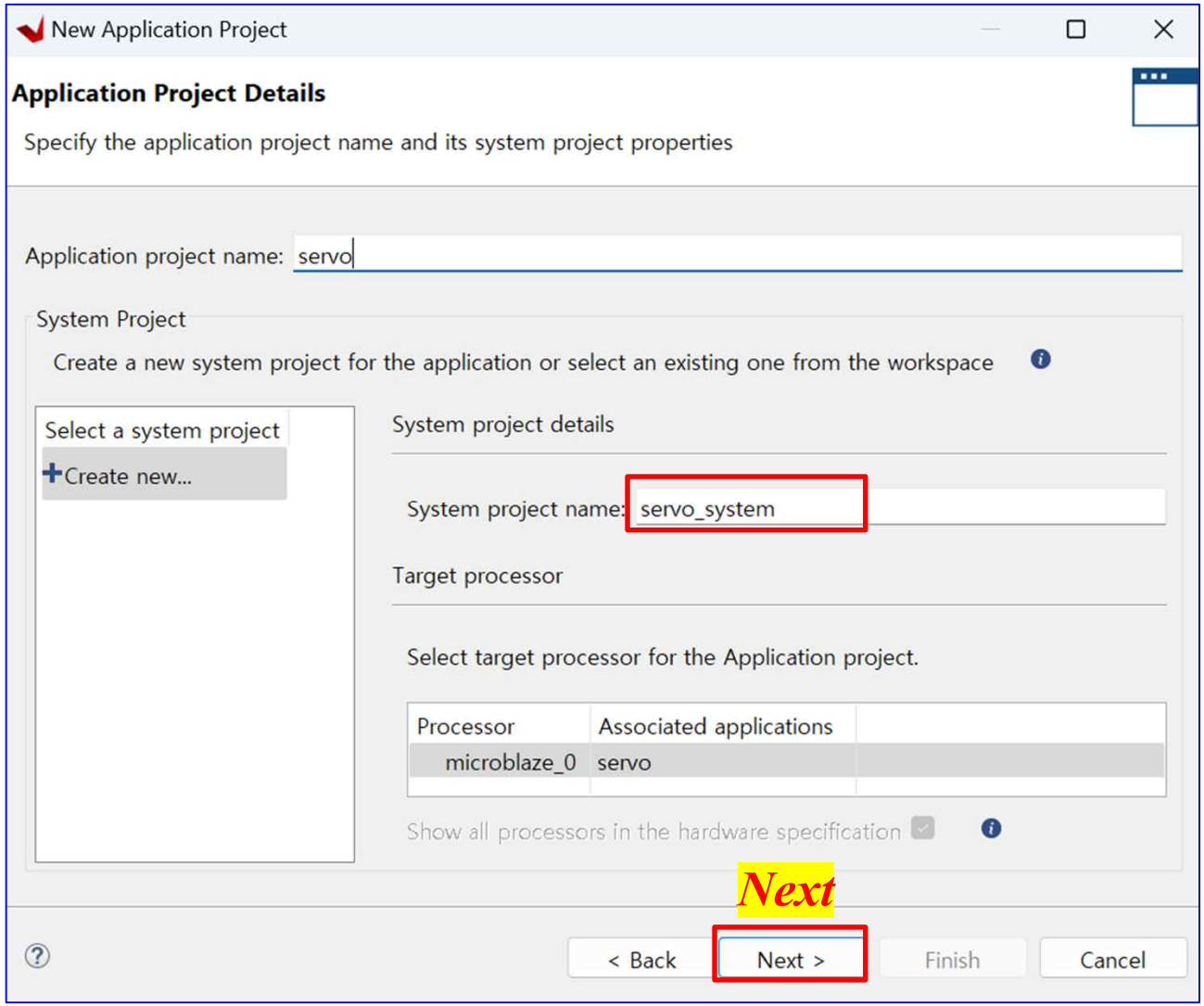
➢ *AXI Connect ➔ AXI Template*



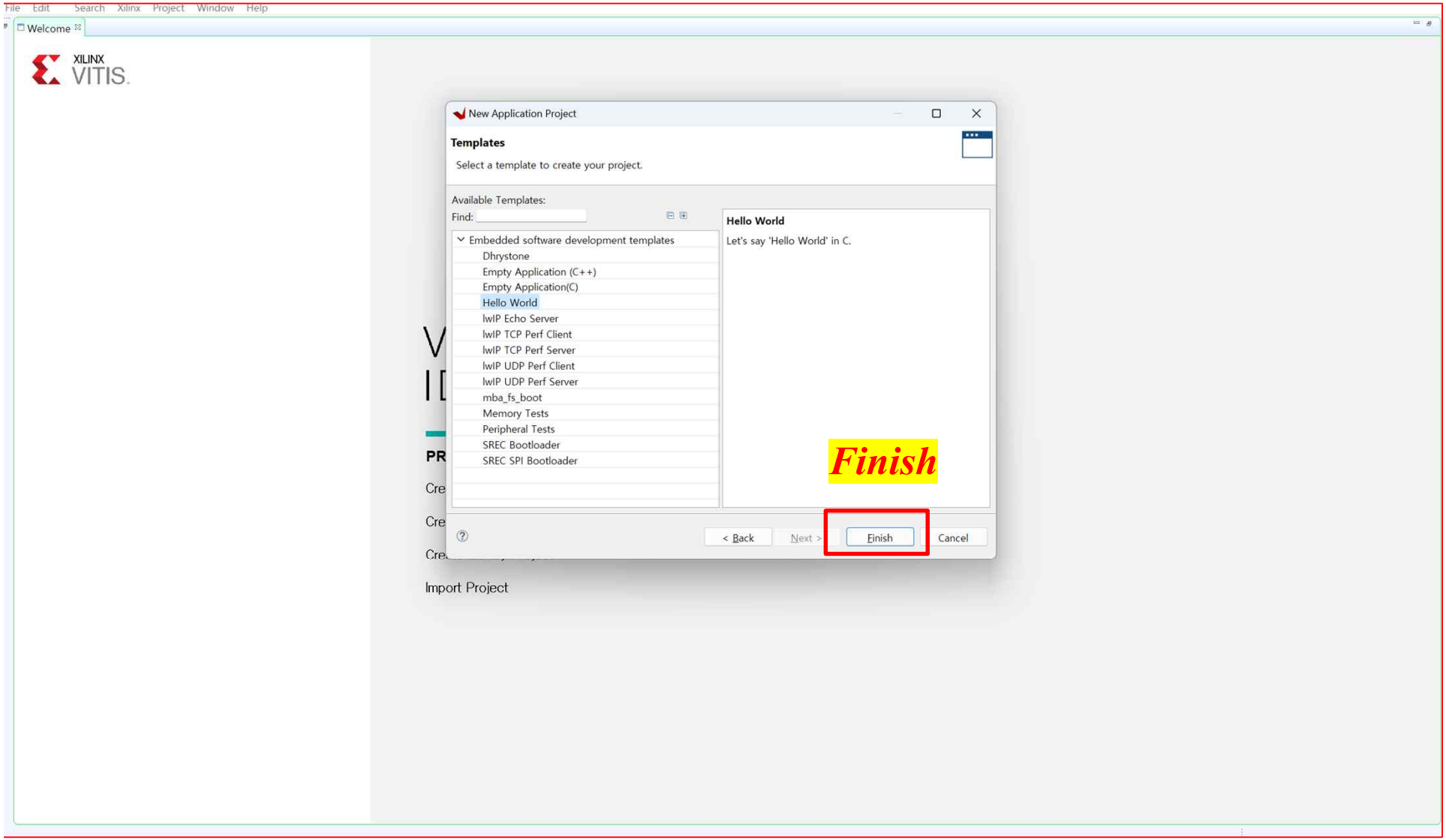
➢ AXI Connect ➔ AXI Template



➢ AXI Connect ➔ AXI Template



- ***AXI Connect ➔ AXI Template***



➢ AXI Connect ➔ AXI Template

Explorer
- servo_system [servo_wrapper]
  - servo [standalone_microblaze_0]
    - Includes
    - src
    - _ide
    - servo.prj
  - servo_system.sprj
- servo_wrapper

Assistant
- servo_system [System]
  - servo [Application]
    - Debug
    - Release
- servo_wrapper [Platform]

servo_system | servo

Application Project Settings

Active build configuration: Debug

General

Project name: servo

Platform: servo_wrapper

Runtime: cpp

Domain: standalone_microblaze_0

CPU: microblaze_0

OS: standalone

Hardware Specification: View processors, memory ranges and peripherals.

Options

View current BSP settings, or configure settings like STDIO peripheral selection, compiler flags, SW intrusive profiling, add/remove libraries, assign drivers to peripherals, change versions of OS/libraries/drivers etc.

Navigate to BSP Settings

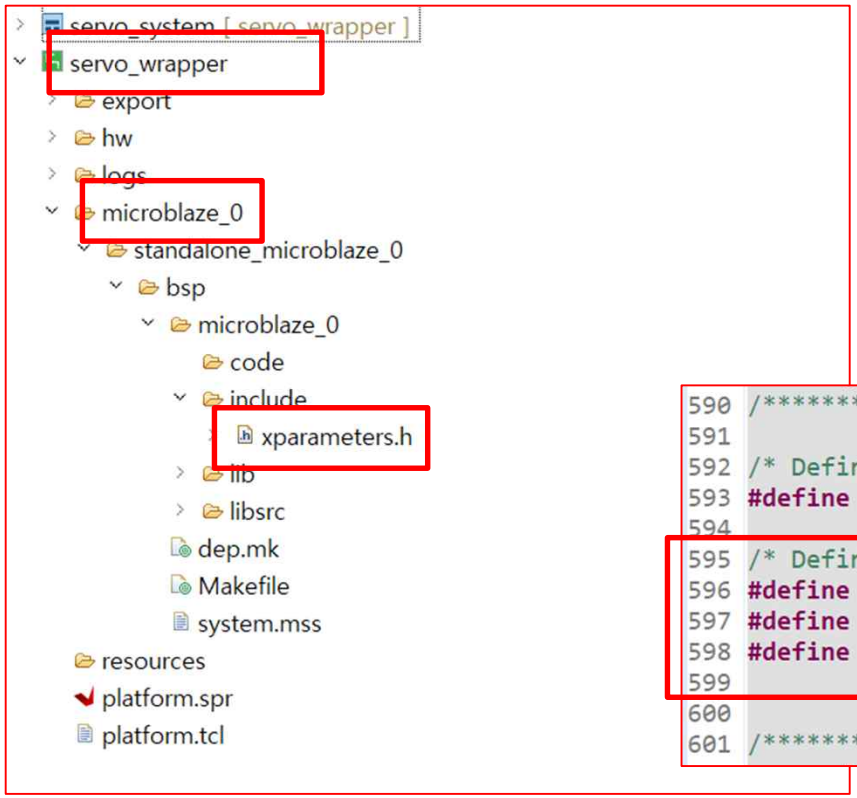
Outline

There is no active editor that provides an outline.

Console | Problems | Vitis Log | Guidance

Build Console [servo, Debug]

➢ AXI Connect ➔ AXI Template



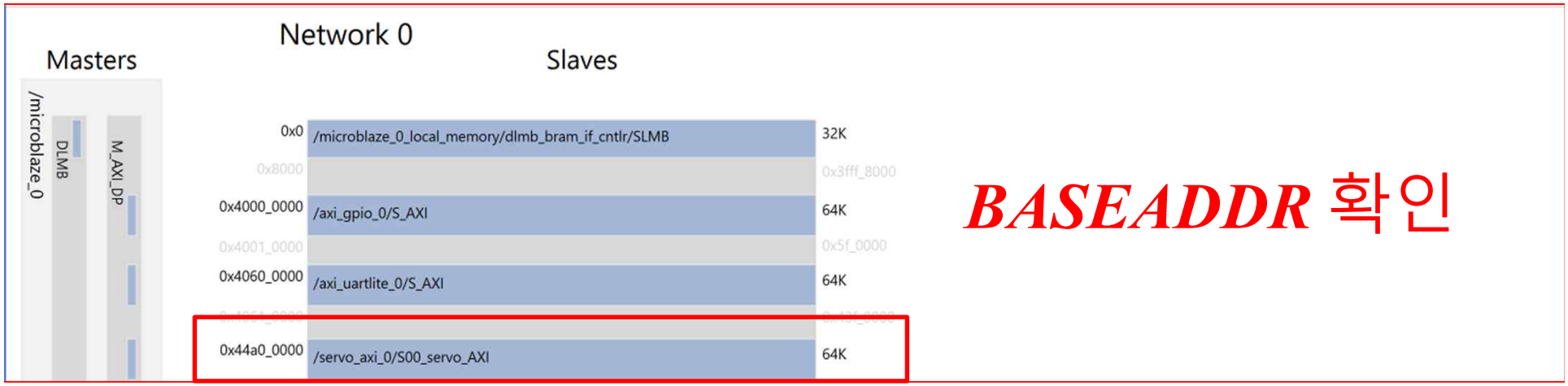

```
590 /*****************************************************************/
591
592 /* Definitions for driver SERVO_AXI */
593 #define XPAR_SERVO_AXI_NUM_INSTANCES 1
594
595 /* Definitions for peripheral SERVO_AXI_0 */
596 #define XPAR_SERVO_AXI_0_DEVICE_ID 0
597 #define XPAR_SERVO_AXI_0_S00_SERVO_AXI_BASEADDR 0x44A00000
598 #define XPAR_SERVO_AXI_0_S00_SERVO_AXI_HIGHADDR 0x44A0FFFF
599
600
601 /*****************************************************************/
```

BASEADDR 확인

AMBA and AXI Connect [Servo Motor]

➢ ***AXI Connect ➔ AXI Template***

Network 0					
/microblaze_0					
/microblaze_0/Data (32 address bits : 4G)					
/axi_gpio_0/S_AXI	S_AXI	Reg	0x4000_0000	64K	0x4000_FFFF
/axi_uartlite_0/S_AXI	S_AXI	Reg	0x4060_0000	64K	0x4060_FFFF
~~/microblaze_0_local_memory/dlmb_bram_if_cntlr/SLMB~~	~~SLMB~~	~~Mem~~	~~0x0000_0000~~	~~32K~~	~~0x0000_7FFF~~
/servo_axi_0/S00_servo_AXI	S00_servo_AXI	S00_servo_AXI_reg	0x44A0_0000	64K	0x44A0_FFFF
Network 1					
/microblaze_0					
/microblaze_0/Instruction (32 address bits : 4G)					
/microblaze_0_local_memory/ilmb_bram_if_cntlr/SLMB	SLMB	Mem	0x0000_0000	32K	0x0000_7FFF



BASEADDR 확인

➢ AXI Connect ➔ AXI Template

File Edit Search Xilinx Project Window Help

Explorer

- axi_fnd_system [axi_fnd_wrapper]
  - axi_fnd [standalone_microblaze_0]
    - Includes
    - src
      - helloworld.c
      - platform.config.h
      - platform.c
      - platform.h
      - lscript.ld
    - _ide
    - axi_fnd.prj
  - axi_fnd_system.sprj
- axi_fnd_wrapper

Assistant

- axi_fnd_system [System]
  - axi_fnd [Application]
    - Debug
    - Release
- axi_fnd_wrapper [Platform]

axi_fnd_system | axi_fnd | xparameters.h | helloworld.c

```
21 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
22 * XILINX  BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
23 * WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF
24 * OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
25 * SOFTWARE.
26 *
27 * Except as contained in this notice, the name of the Xilinx shall not be used
28 * in advertising or otherwise to promote the sale, use or other dealings in
29 * this Software without prior written authorization from Xilinx.
30 *
31 ******************************************************************************/
32
33 /*
34  * helloworld.c: simple test application
35  *
36  * This application configures UART 16550 to baud rate 9600.
37  * PS7 UART (Zynq) is not initialized by this application, since
38  * bootrom/bsp configures it to baud rate 115200
39  *
40  * ----------------------------------------------
41  * | UART TYPE   BAUD RATE                      |
42  * ----------------------------------------------
43  *   uartns550   9600
44  *   uartlite    Configurable only in HW design
45  *   ps7_uart    115200 (configured by bootrom/bsp)
46  */
47
48 #include <stdio.h>
49 #include "platform.h"
50 #include "xil_printf.h"
51
52
53 int main()
54 {
55     init_platform();
56
57     print("Hello World\n\r");
58     print("Successfully ran Hello World application");
59     cleanup_platform();
60     return 0;
61 }
62
```

Vitis Coding

Outline

- stdio.h
- platform.h
- xil_printf.h
- main() : int

Console | Problems | Vitis Log | Guidance

Build Console [axi_fnd, Debug]

Writable | Smart Insert | 1 : 1 : 0

➢ AXI Connect ➔ AXI Template

Helloworld.c

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xparameters.h"
#include "xgpio.h"
#include "sleep.h"

#define AXI_SERVO_BASEADDR   0x44A00000

/* definitions for btn0, AXI_GPIO_1 */
#define GPIO_BTN0_DEVICE_ID    XPAR_GPIO_0_DEVICE_ID
#define BTN0_CHANNEL1          1
#define BTN0_INT_MSK           XGPIO_IR_CH1_MASK

XGpio    GPIO_BTN;        /* The instance of the btn */
int main()
{

   init_platform();
   XGpio_Initialize(&GPIO_BTN,  GPIO_BTN0_DEVICE_ID);
   print("Hello World\n\r");
   cleanup_platform();
```

주소 확인

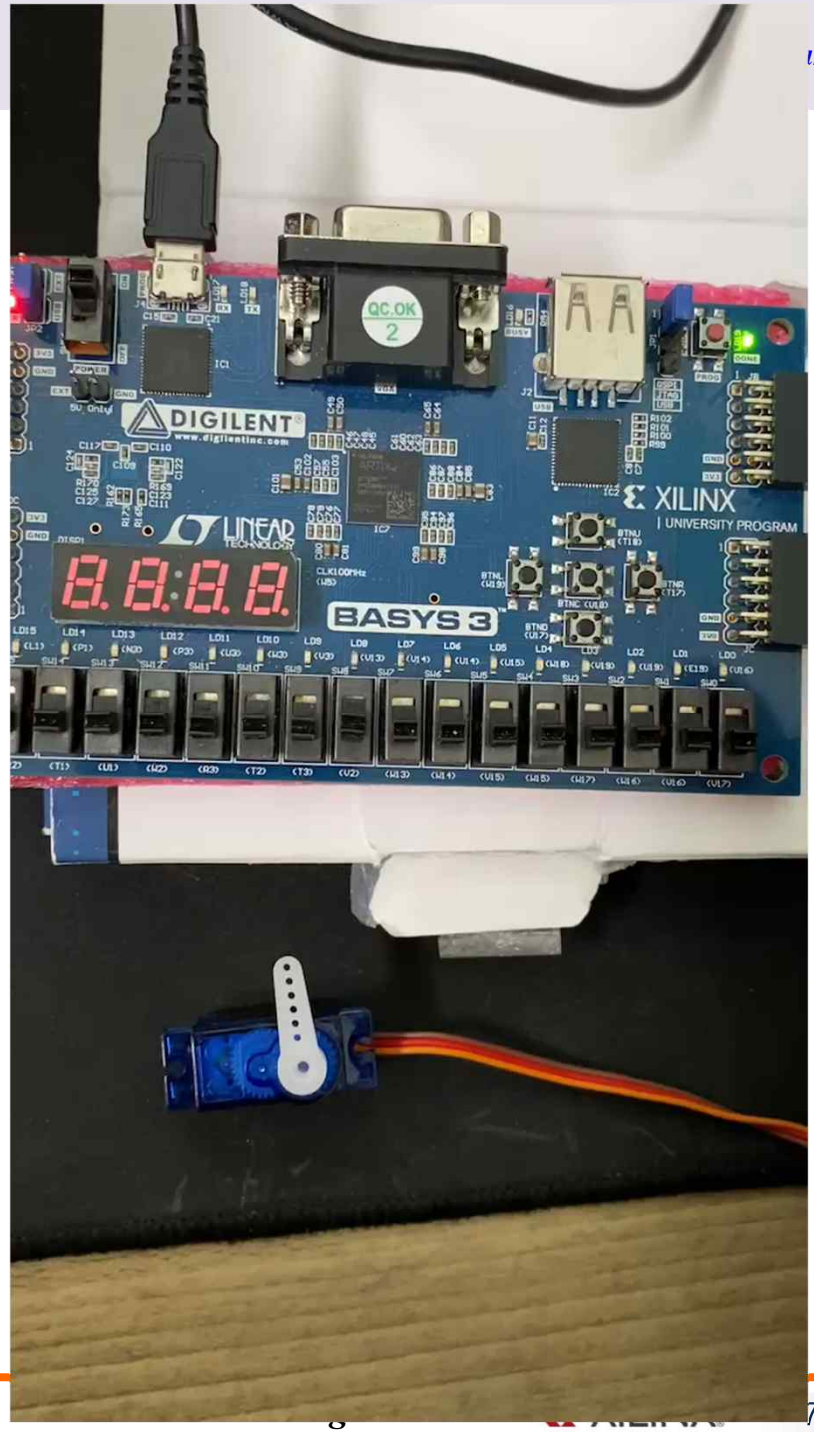
Helloworld.c

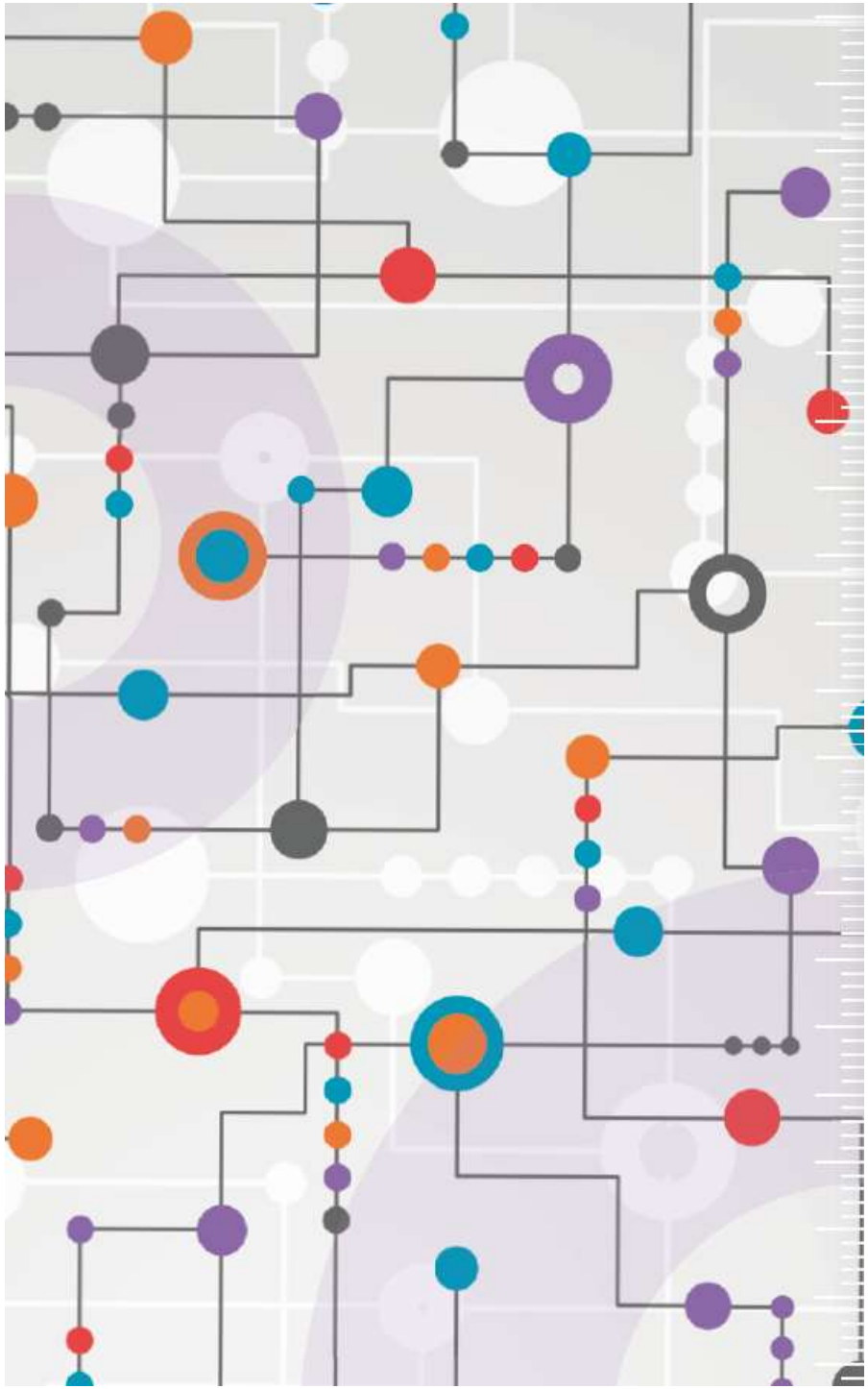
```
   while(1){

      if ((XGpio_DiscreteRead(&GPIO_BTN,
BTN0_CHANNEL1) & (1<<0)))
            Xil_Out32(AXI_SERVO_BASEADDR,0);
      else if ((XGpio_DiscreteRead(&GPIO_BTN,
BTN0_CHANNEL1) & (1<<1)))
         Xil_Out32(AXI_SERVO_BASEADDR,7);
      else if ((XGpio_DiscreteRead(&GPIO_BTN,
BTN0_CHANNEL1) & (1<<2)))
            Xil_Out32(AXI_SERVO_BASEADDR,16);
      else if ((XGpio_DiscreteRead(&GPIO_BTN,
BTN0_CHANNEL1) & (1<<3)))
            Xil_Out32(AXI_SERVO_BASEADDR,26);


   }
   return 0;
}
```

➢Check The Result





수고하셨습니다.