SoC 를 위한 *Peripheral* 설계

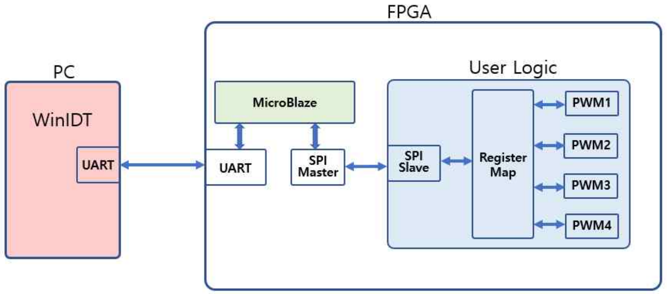
Reference : MicroBlaze.v15 [IHIL]

2024-06-19

Table of Contents

- ➢ SoC 를 위한 Peripheral 설계

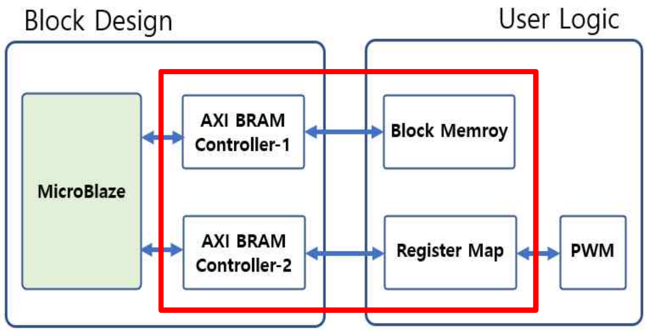
1. Xilinx IP
2. Create and Package New IP
3. SPI
   1) SPI Master
   2) SPI Slave
   3) SPI Controller
4. UART
5. AMBA
6. MicroBlaze_Hello World
7. MicroBlaze_LED_Counter
8. MicroBlaze_Peripheral Implementation
9. MicroBlaze_User Logic Interface



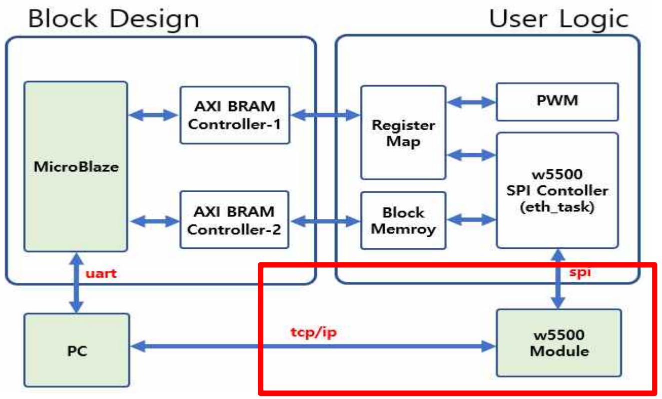
10. SPI_Master_IP(MicroBlaze_User Logic Interface)

spi_master_ip_test.xpr

11. TCP_IP Implementation Using W5500
12. MicroBlaze_Block Memory Interface-1
13. MicroBlaze_Block Memory Interface-2



14. w5500 Interface Implementation



- ➢ SoC Peripheral RTC Design Project

User Logic Interface Implementation

[SPI Master IP TEST]

[Reference] ▪

User Logic Interface Implementation

[Create Block Design]

User Logic Interface Implementation

- ***SPI Master IP Implementation and Test***

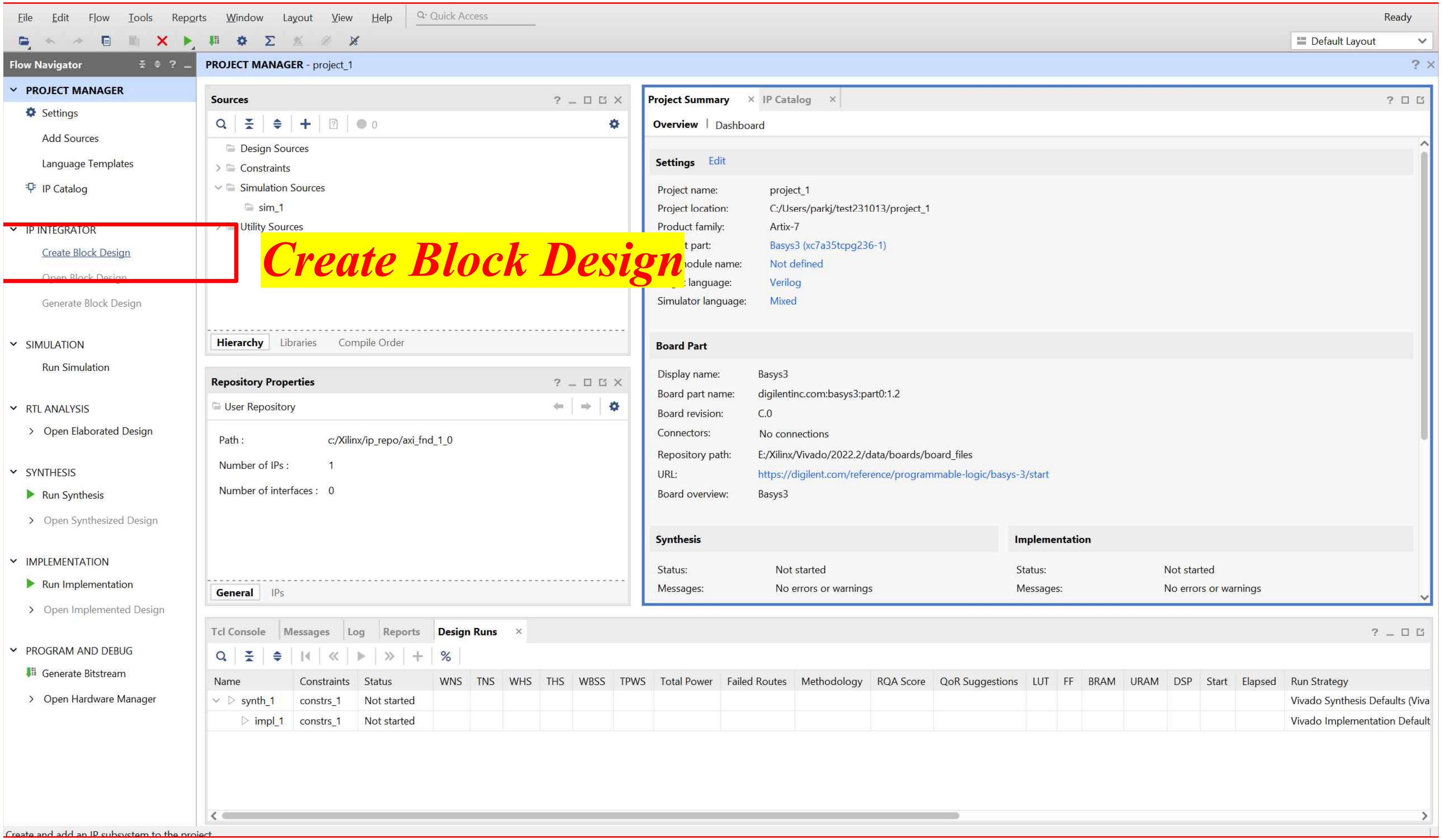
User Logic Interface Implementation

➢ SPI Master IP TEST → Sequency

- Create Block Design
- Blaze Uart & quad_SPI 추가 및 연결
- sys_clock & reset 설정
- SPI ss, sck Make External
- XDC Constraints 설정
- Bitstream
- Export Hardwar

User Logic Interface Implementation

spi_master_ip_test.xpr



Create Block Design

Path : c:/Xilinx/ip_repo/axi_fnd_1_0

Number of IPs : 1

Number of interfaces : 0

Project name: project_1

Project location: C:/Users/parkj/test231013/project_1

Product family: Artix-7

t part: Basys3 (xc7a35tcpg236-1)

odule name: Not defined

language: Verilog

Simulator language: Mixed

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

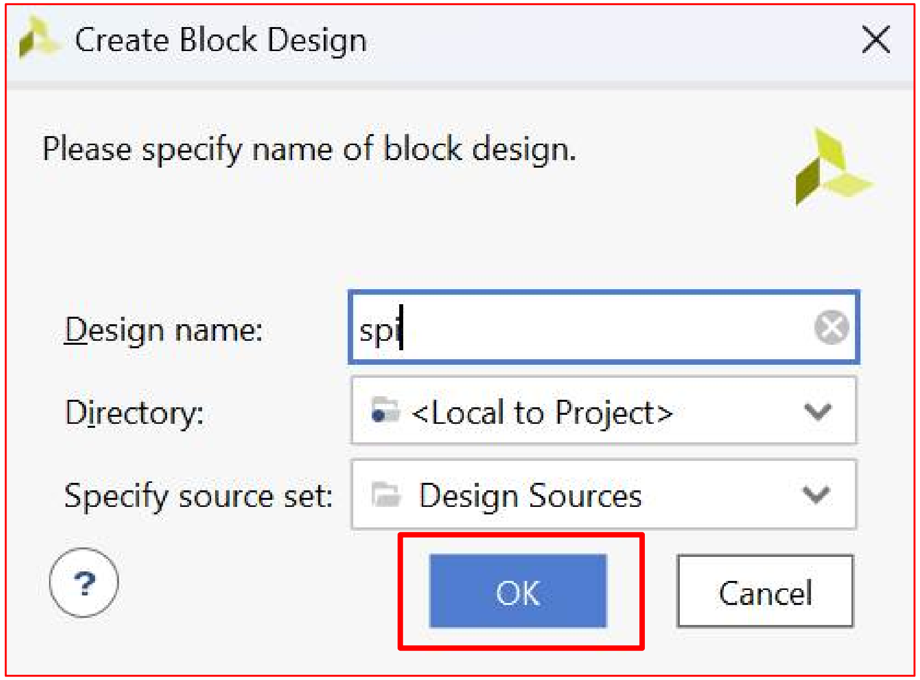
Board overview: Basys3

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

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

User Logic Interface Implementation



Create Block Design

Please specify name of block design.

Design name: spi

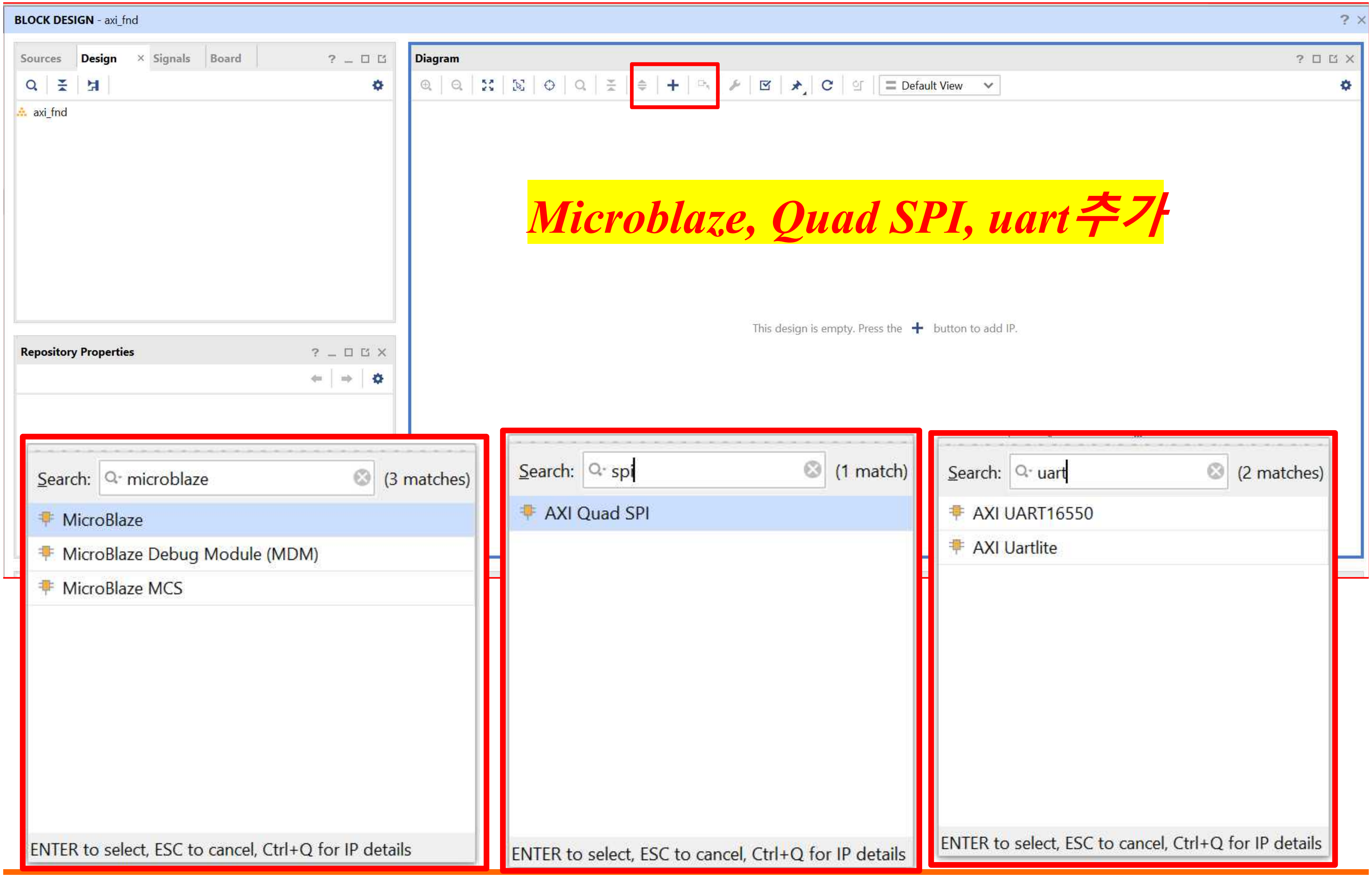
Directory: <Local to Project>

Specify source set: Design Sources

이름 설정 후 OK

User Logic Interface Implementation

spi_master_ip_test.xpr



BLOCK DESIGN - axi_fnd

Microblaze, Quad SPI, uart추가

This design is empty. Press the + button to add IP.

Search: microblaze (3 matches)

- MicroBlaze
- MicroBlaze Debug Module (MDM)
- MicroBlaze MCS

ENTER to select, ESC to cancel, Ctrl+Q for IP details

Search: spi (1 match)

- AXI Quad SPI

ENTER to select, ESC to cancel, Ctrl+Q for IP details

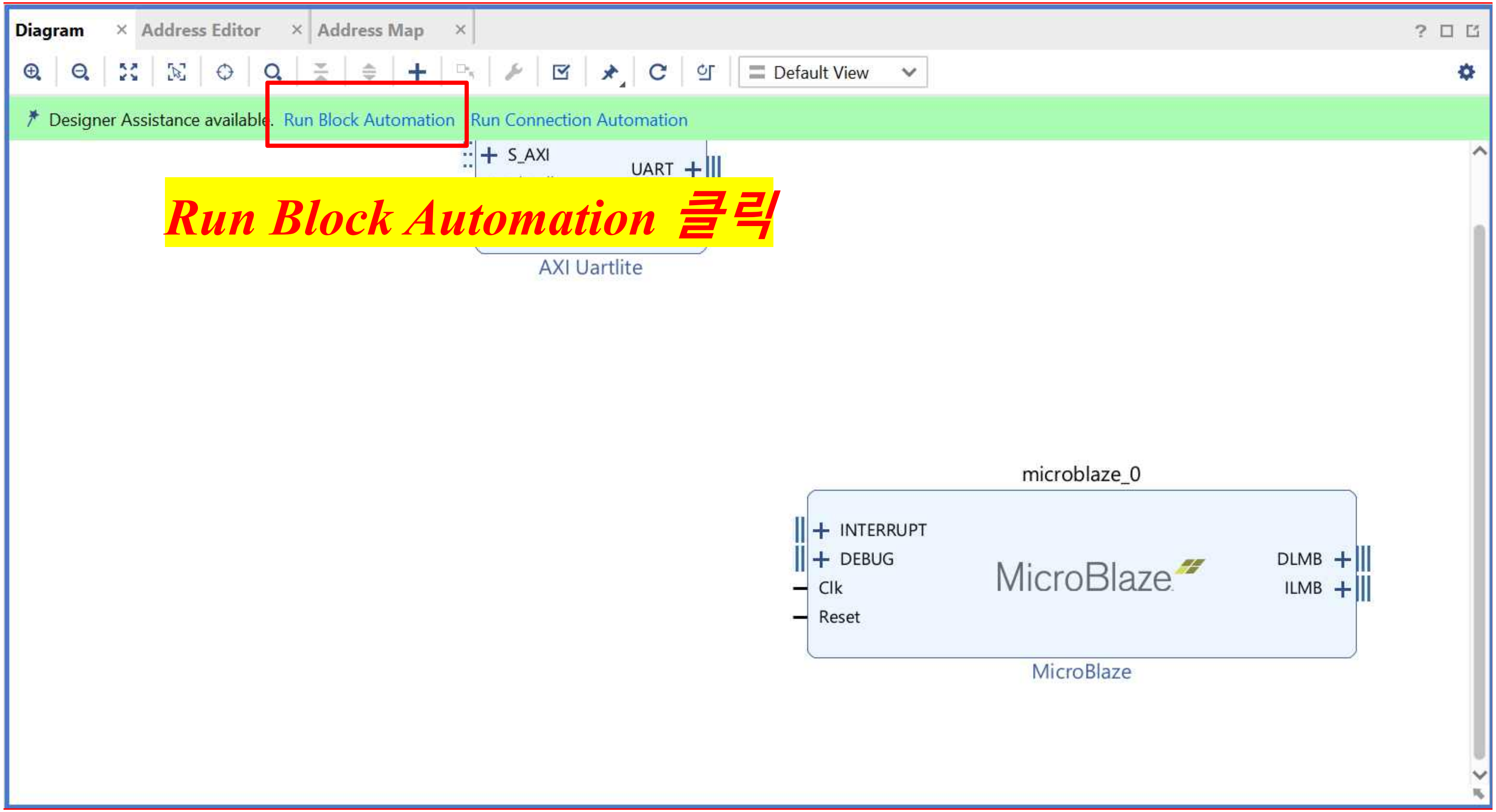
Search: uart (2 matches)

- AXI UART16550
- AXI Uartlite

ENTER to select, ESC to cancel, Ctrl+Q for IP details

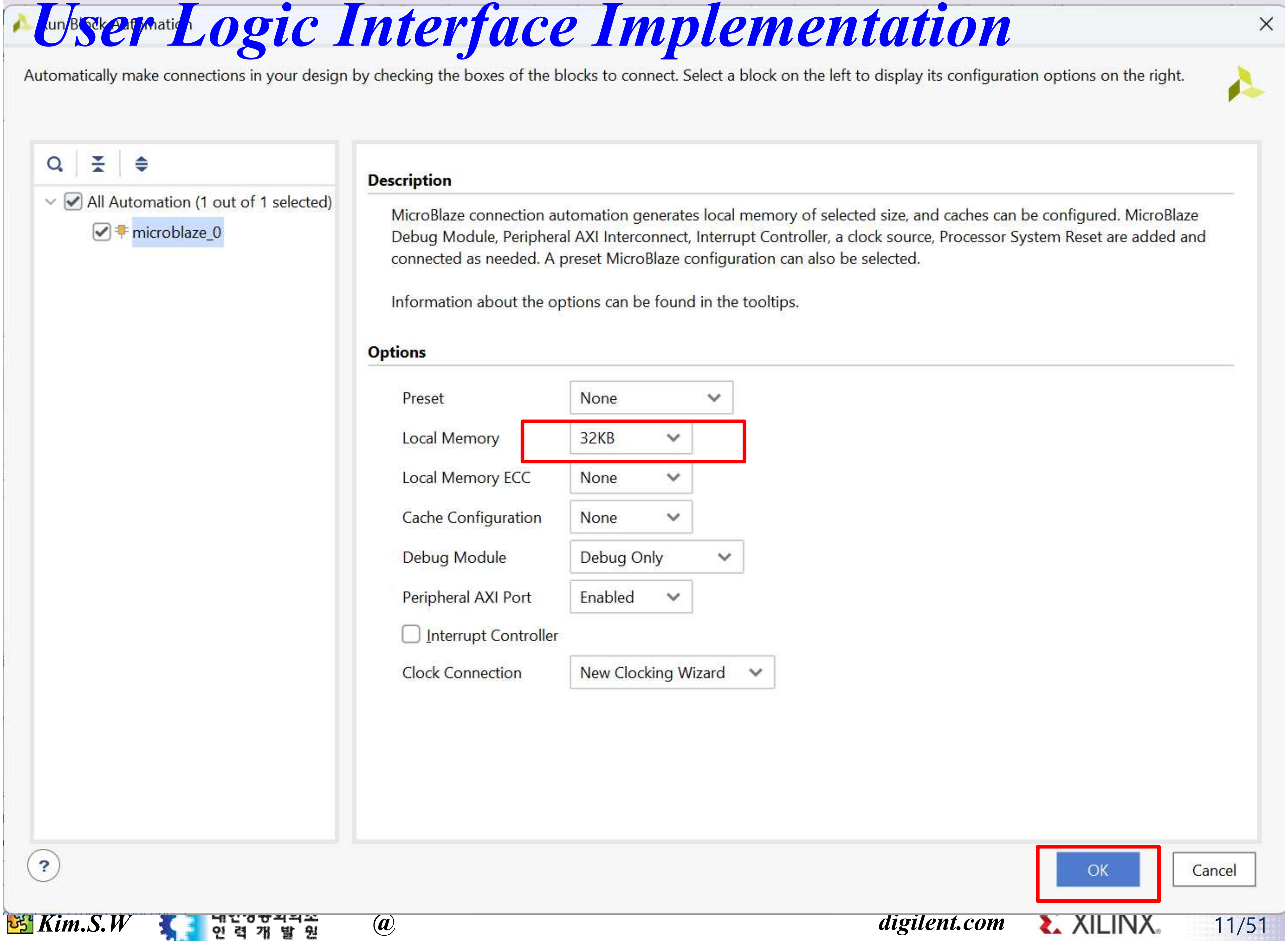
User Logic Interface Implementation

spi_master_ip_test.xpr



Run Block Automation 클릭

User Logic Interface Implementation



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

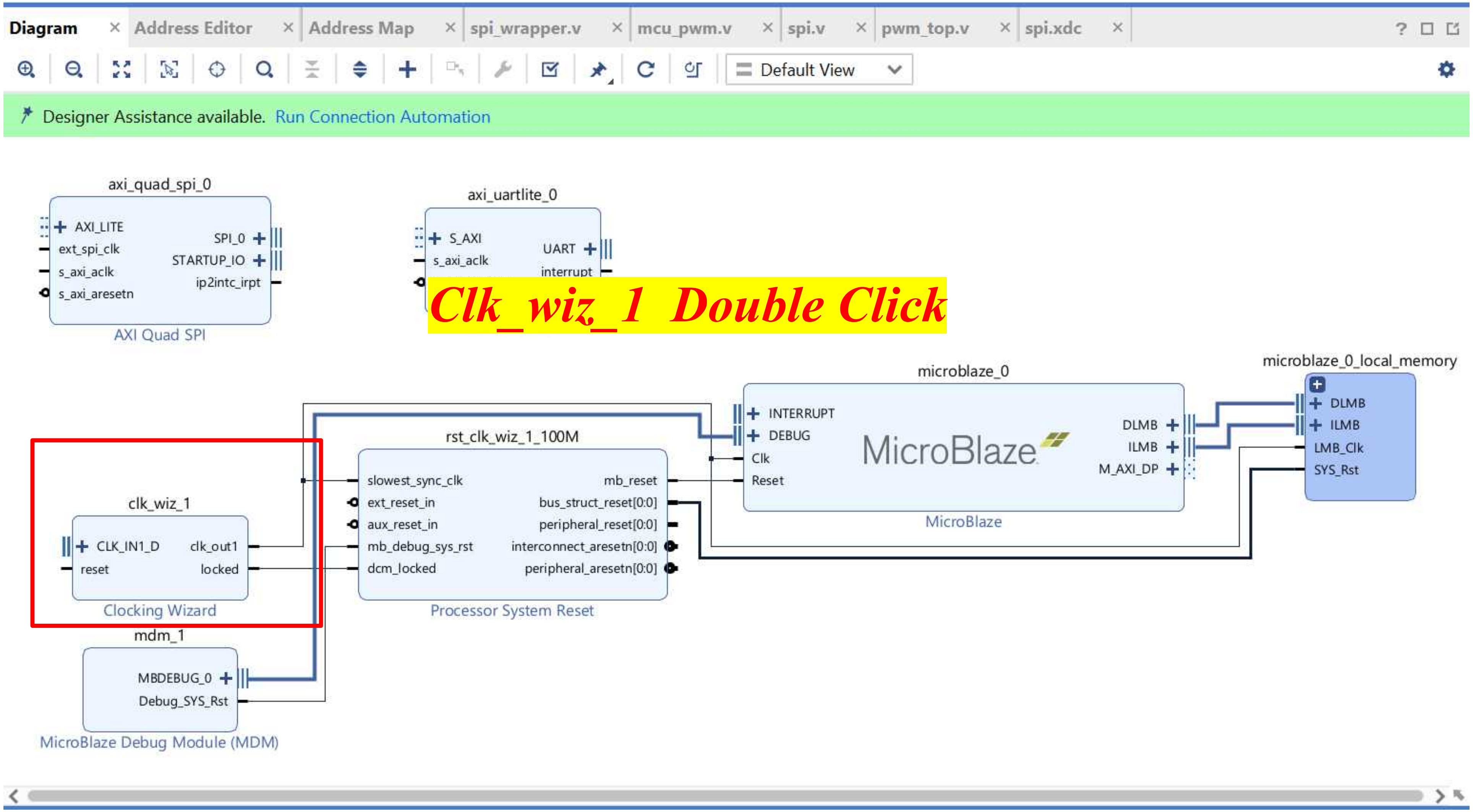
Preset	None
Local Memory	32KB
Local Memory ECC	None
Cache Configuration	None
Debug Module	Debug Only
Peripheral AXI Port	Enabled

□ Interrupt Controller

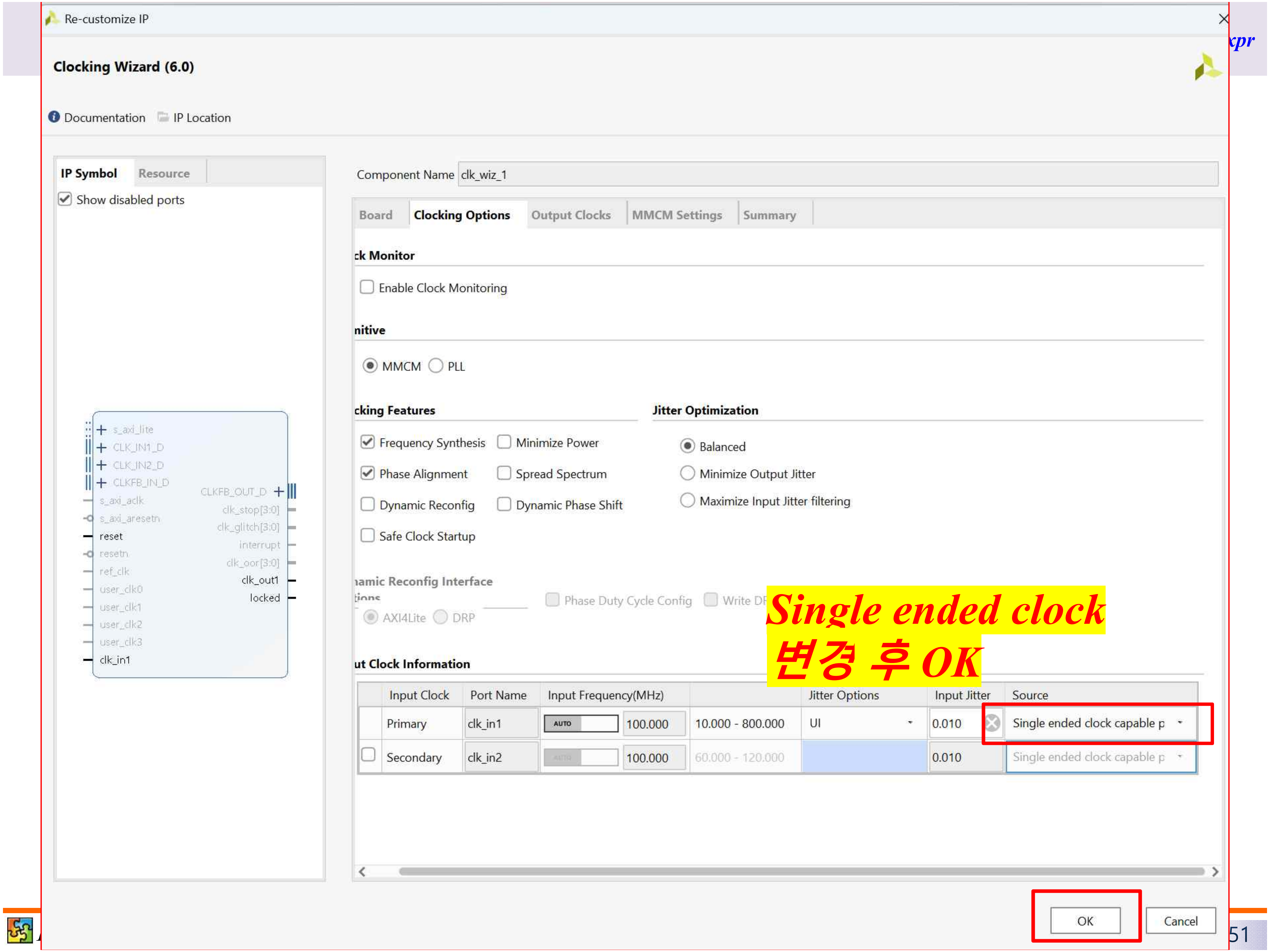
Clock Connection	New Clocking Wizard

User Logic Interface Implementation

spi_master_ip_test.xpr



Clk_wiz_1 Double Click



Re-customize IP

Clocking Wizard (6.0)

Documentation   IP Location

Component Name clk_wiz_1

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

ck Monitor

□ Enable Clock Monitoring

nitive

● MMCM ○ PLL

cking 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

amic Reconfig Interface

tions

● AXI4Lite ○ DRP

□ Phase Duty Cycle Config □ Write DF

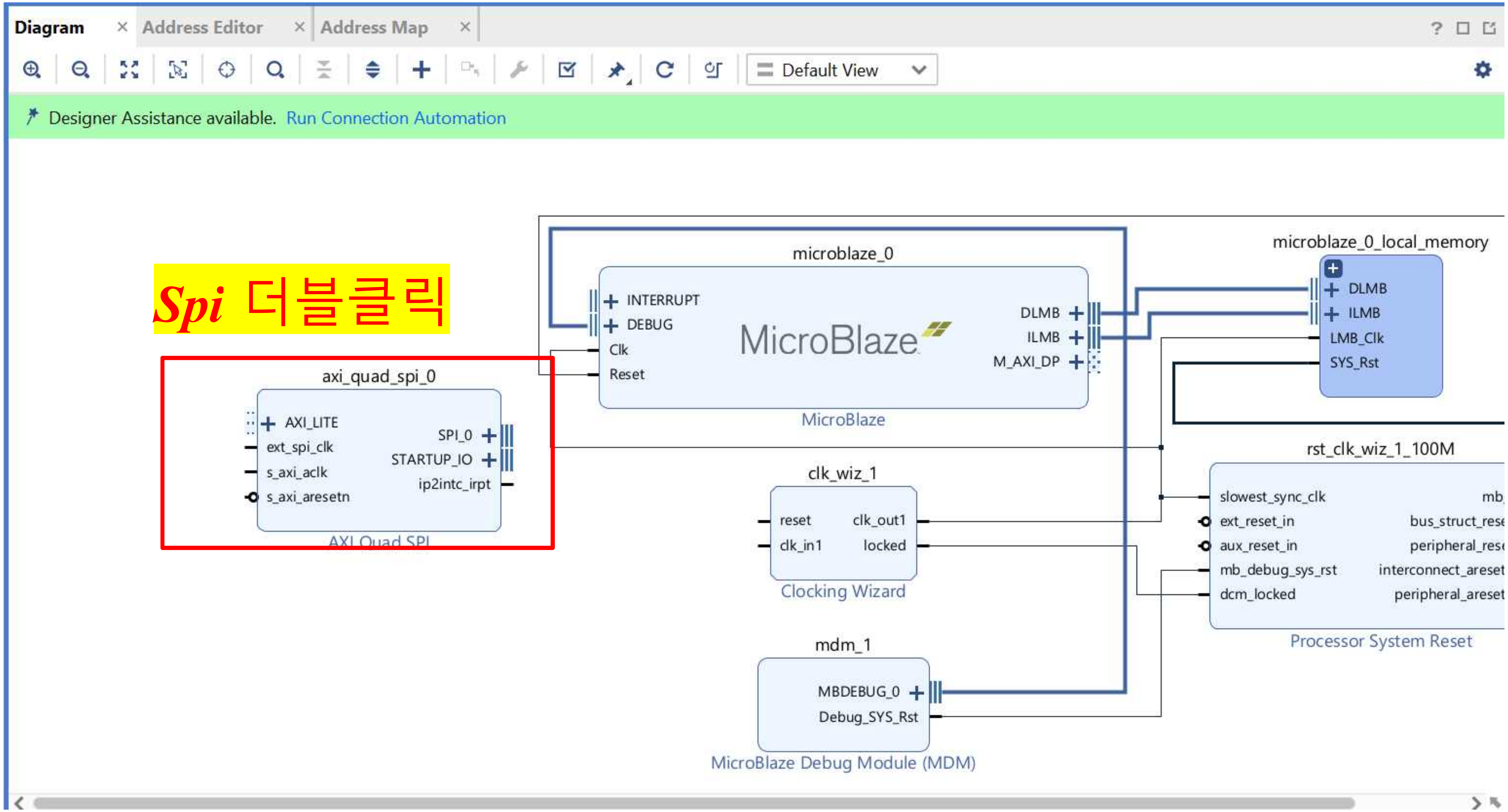
ut Clock Information

	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

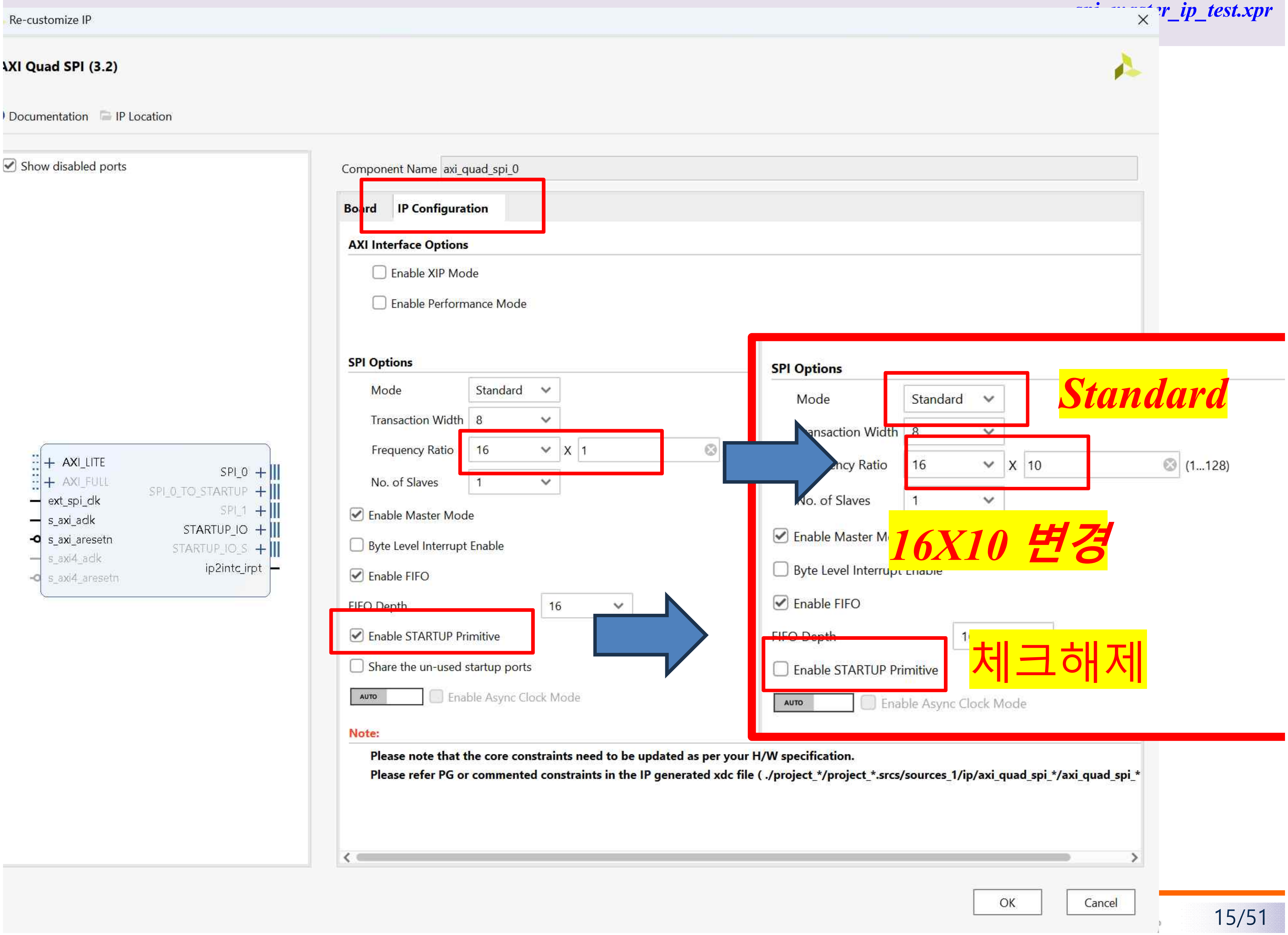
OK   Cancel

Single ended clock 변경 후 OK

User Logic Interface Implementation



Spi 더블클릭



Re-customize IP

AXI Quad SPI (3.2)

Documentation   IP Location

☑ Show disabled ports

Component Name axi_quad_spi_0

Board   IP Configuration

AXI Interface Options

☐ Enable XIP Mode

☐ Enable Performance Mode

SPI Options

Mode Standard

Transaction Width 8

Frequency Ratio 16 X 1

No. of Slaves 1

☑ Enable Master Mode

☐ Byte Level Interrupt Enable

☑ Enable FIFO

FIFO Depth 16

☑ Enable STARTUP Primitive

☐ Share the un-used startup ports

AUTO ☐ Enable Async Clock Mode

Note:

Please note that the core constraints need to be updated as per your H/W specification.

Please refer PG or commented constraints in the IP generated xdc file (./project_*/project_*.srcs/sources_1/ip/axi_quad_spi_*/axi_quad_spi_*

OK   Cancel

SPI Options

Mode Standard

ansaction Width 8

ncy Ratio 16 X 10 (1...128)

No. of Slaves 1

☑ Enable Master M

☐ Byte Level Interrupt Enable

☑ Enable FIFO

FIFO Depth 1

☐ Enable STARTUP Primitive

AUTO ☐ Enable Async Clock Mode

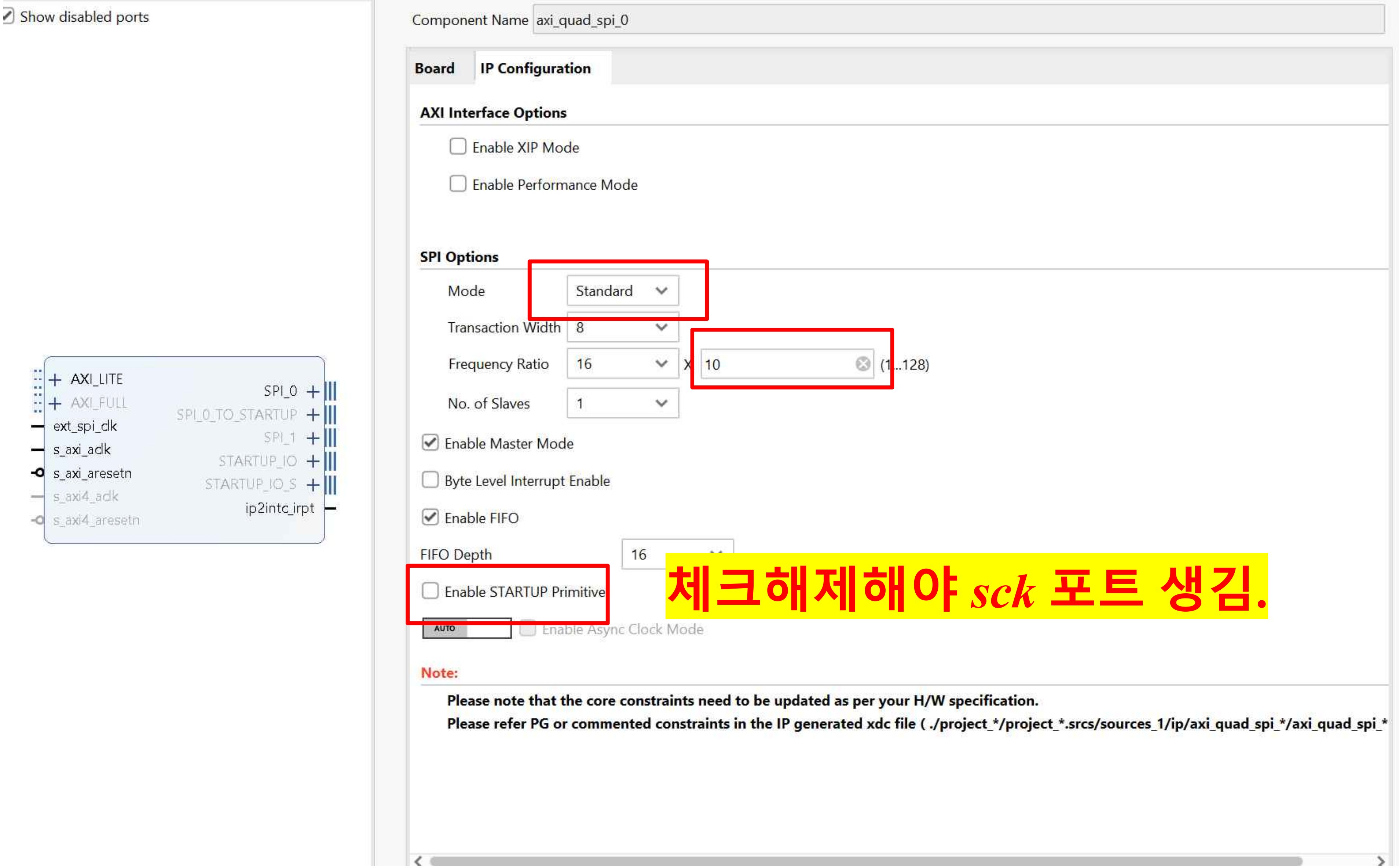
Standard

16X10 변경

체크해제

User Logic Interface Implementation

spi_master_ip_test.xpr



☑ Show disabled ports

Component Name axi_quad_spi_0

Board | IP Configuration

AXI Interface Options

☐ Enable XIP Mode

☐ Enable Performance Mode

SPI Options

Mode: Standard

Transaction Width: 8

Frequency Ratio: 16 X 10 (1..128)

No. of Slaves: 1

☑ Enable Master Mode

☐ Byte Level Interrupt Enable

☑ Enable FIFO

FIFO Depth: 16

☐ Enable STARTUP Primitive

AUTO ☐ Enable Async Clock Mode

Note:

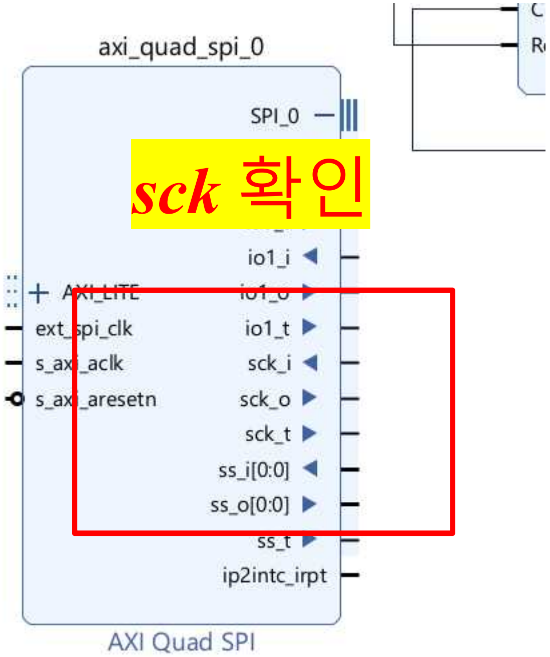
Please note that the core constraints need to be updated as per your H/W specification.

Please refer PG or commented constraints in the IP generated xdc file (./project_*/project_*.srcs/sources_1/ip/axi_quad_spi_*/axi_quad_spi_*

체크해제해야 *sck* 포트 생김.

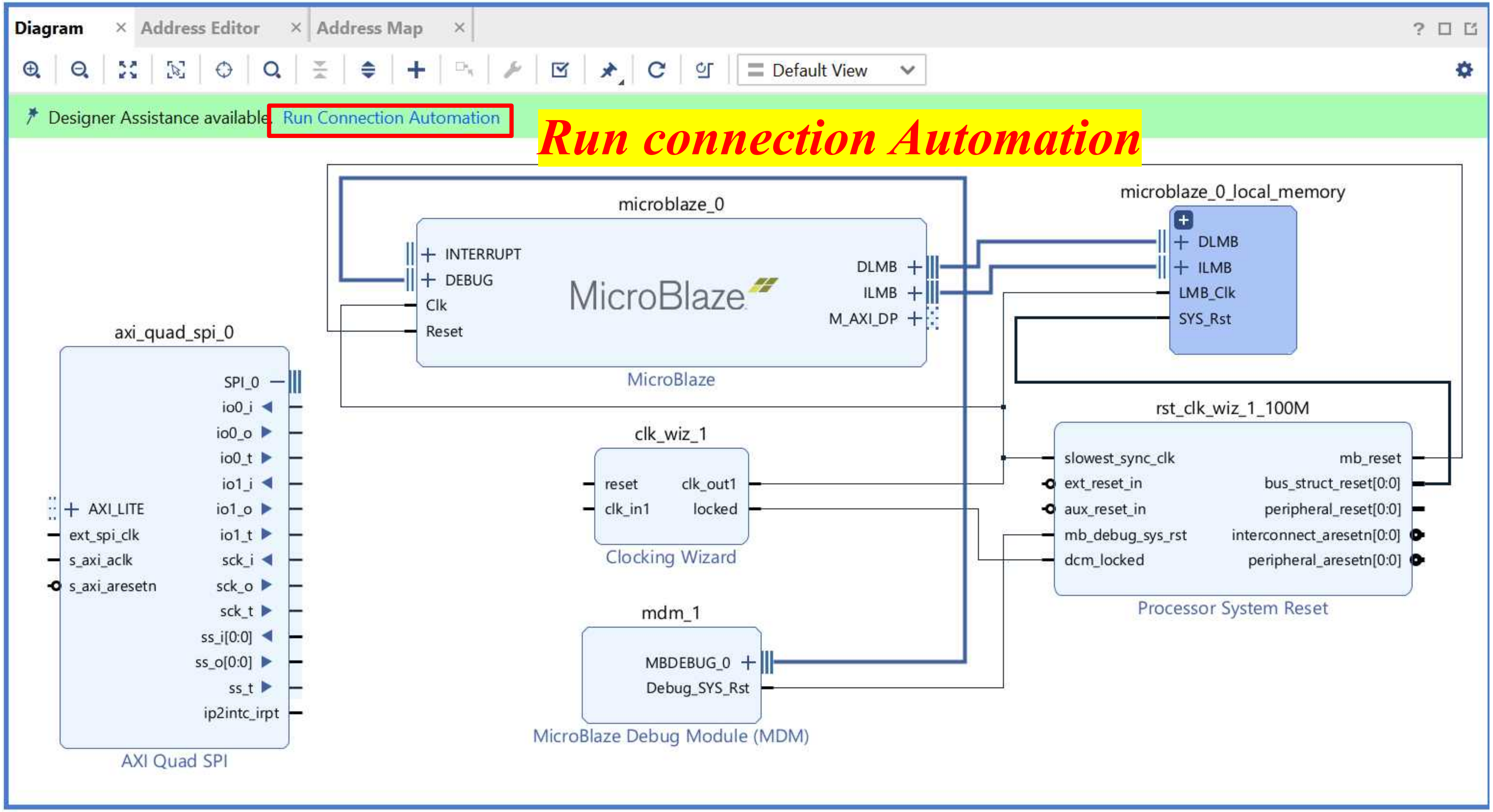
User Logic Interface Implementation

spi_master_ip_test.xpr

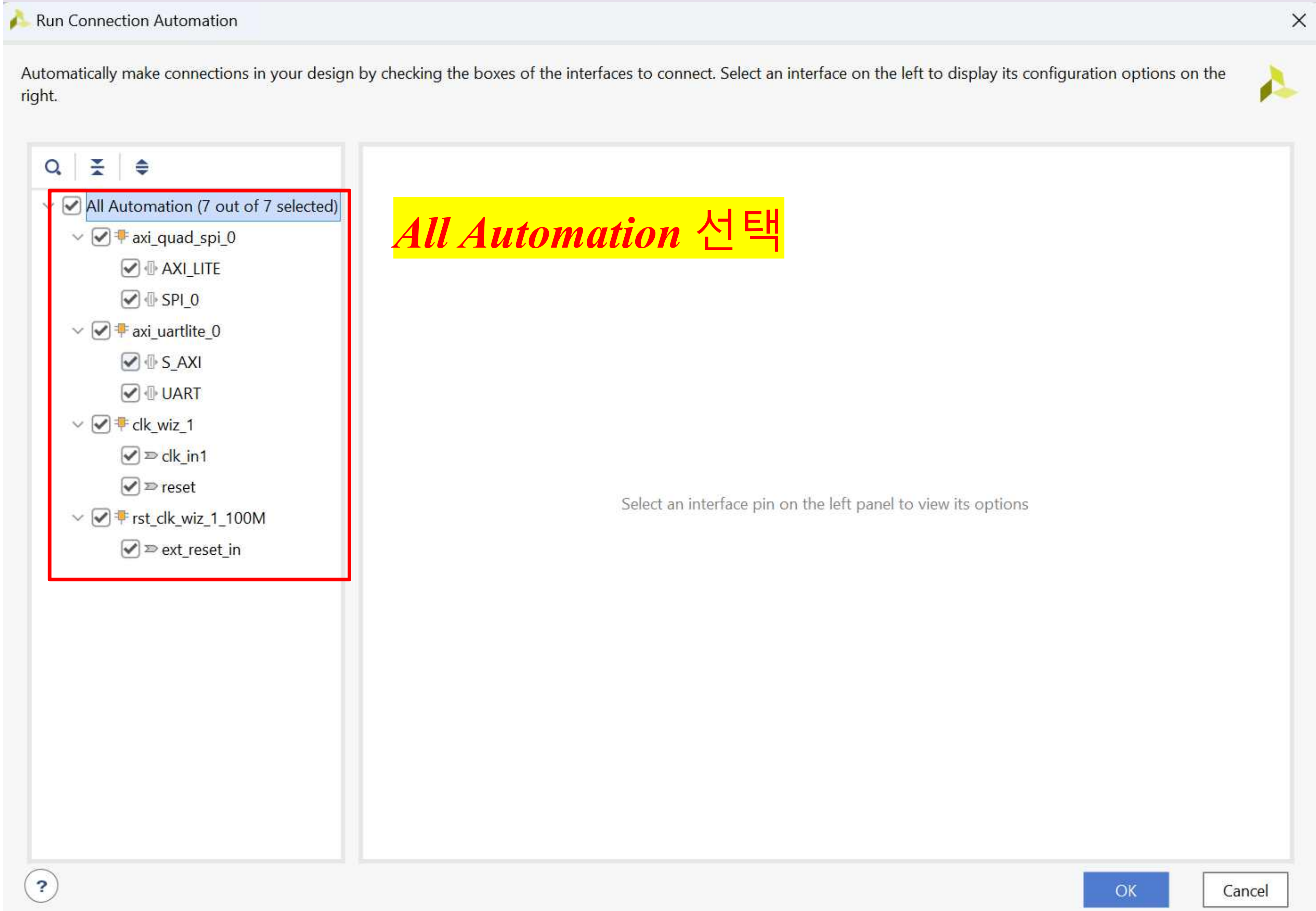


User Logic Interface Implementation

spi_master_ip_test.xpr



Run connection Automation



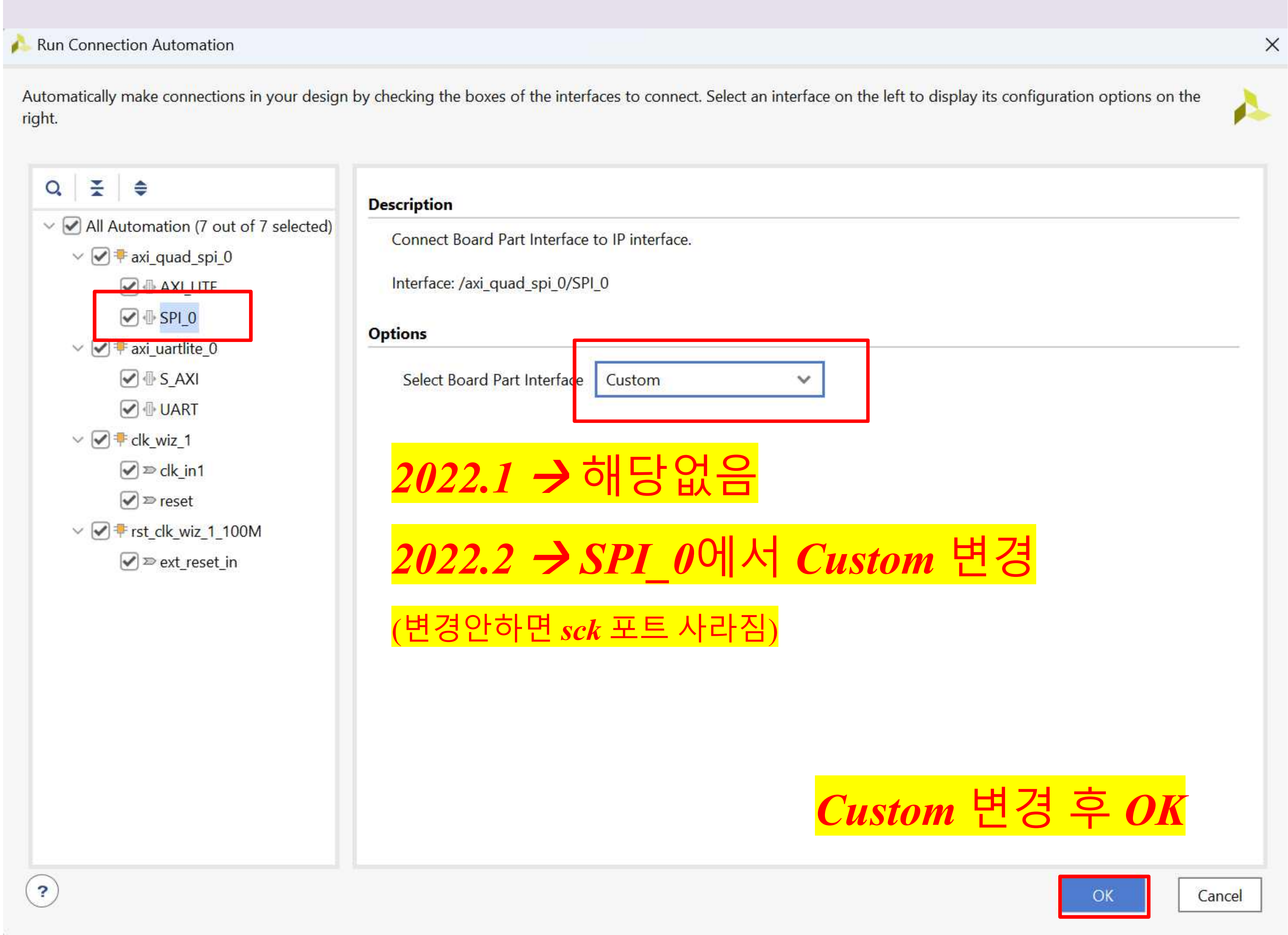
Run Connection Automation

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 (7 out of 7 selected)
  - ☑ axi_quad_spi_0
    - ☑ AXI_LITE
    - ☑ SPI_0
  - ☑ axi_uartlite_0
    - ☑ S_AXI
    - ☑ UART
  - ☑ clk_wiz_1
    - ☑ clk_in1
    - ☑ reset
  - ☑ rst_clk_wiz_1_100M
    - ☑ ext_reset_in

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

All Automation 선택



Run Connection Automation

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 (7 out of 7 selected)
  - ☑ axi_quad_spi_0
    - ☑ AXI_LITE
    - ☑ SPI_0
  - ☑ axi_uartlite_0
    - ☑ S_AXI
    - ☑ UART
  - ☑ clk_wiz_1
    - ☑ clk_in1
    - ☑ reset
  - ☑ rst_clk_wiz_1_100M
    - ☑ ext_reset_in

Description

Connect Board Part Interface to IP interface.

Interface: /axi_quad_spi_0/SPI_0

Options

Select Board Part Interface: Custom

2022.1 → 해당없음

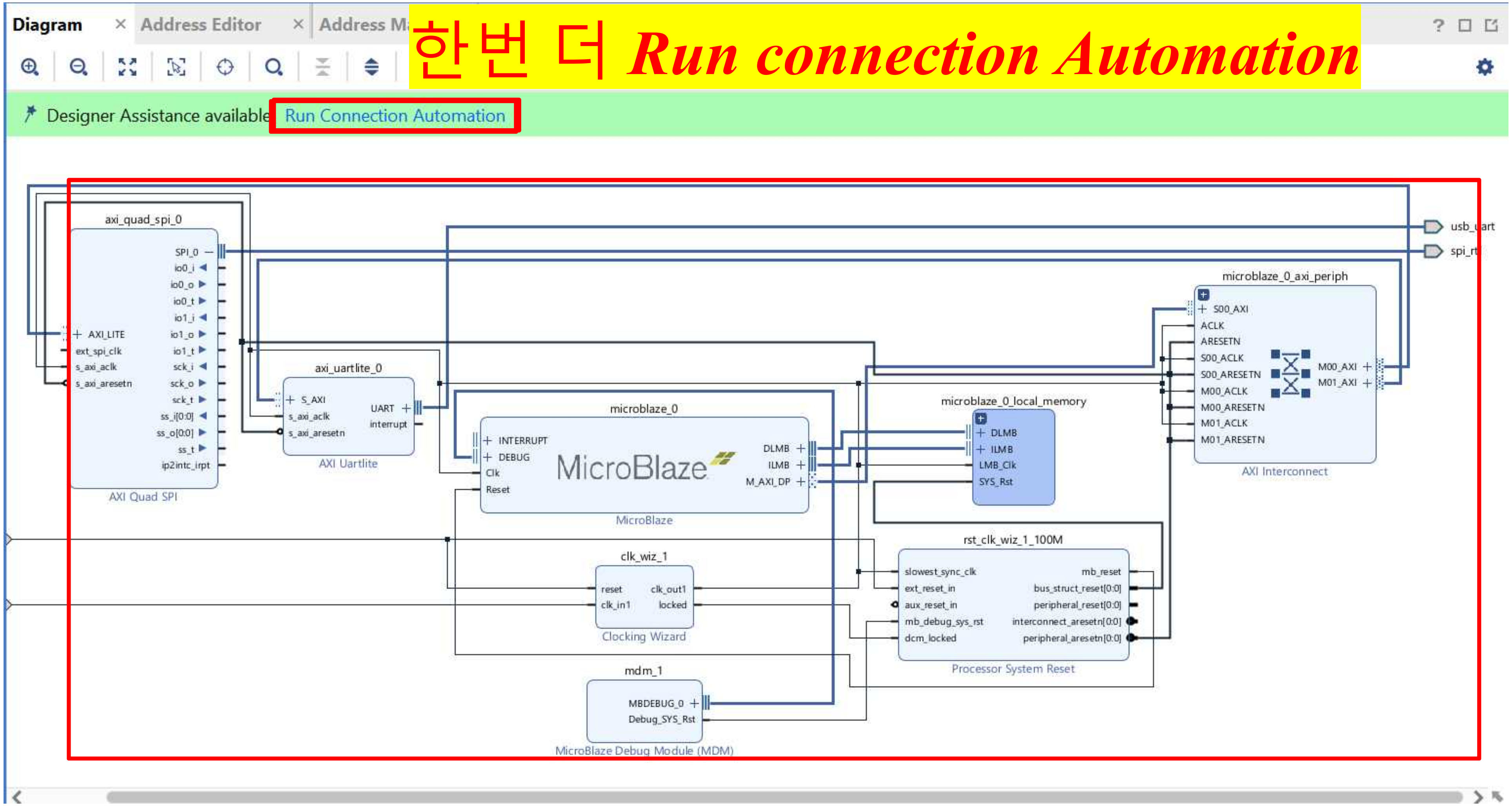
2022.2 → SPI_0에서 Custom 변경

(변경안하면 *sck* 포트 사라짐)

Custom 변경 후 OK

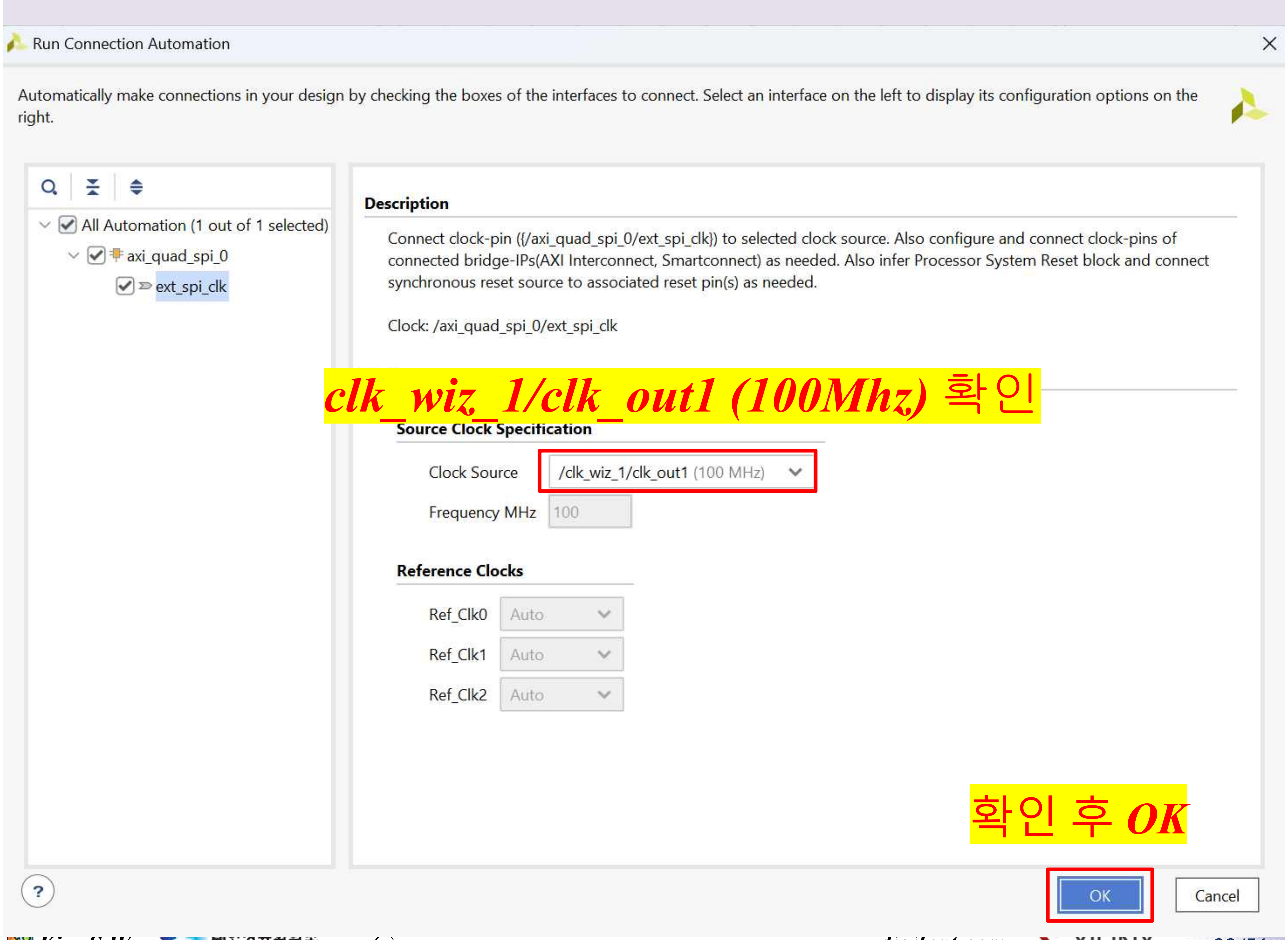
User Logic Interface Implementation

spi_master_ip_test.xpr



한번 더 *Run connection Automation*

Designer Assistance available. Run Connection Automation



Run Connection Automation

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 (1 out of 1 selected)
  - ☑ axi_quad_spi_0
    - ☑ ext_spi_clk

Description

Connect clock-pin ({/axi_quad_spi_0/ext_spi_clk}) to selected clock source. Also configure and connect clock-pins of connected bridge-IPs(AXI Interconnect, Smartconnect) as needed. Also infer Processor System Reset block and connect synchronous reset source to associated reset pin(s) as needed.

Clock: /axi_quad_spi_0/ext_spi_clk

***clk_wiz_1/clk_out1 (100Mhz)* 확인**

Source Clock Specification

Clock Source: /clk_wiz_1/clk_out1 (100 MHz)

Frequency MHz: 100

Reference Clocks

Ref_Clk0: Auto

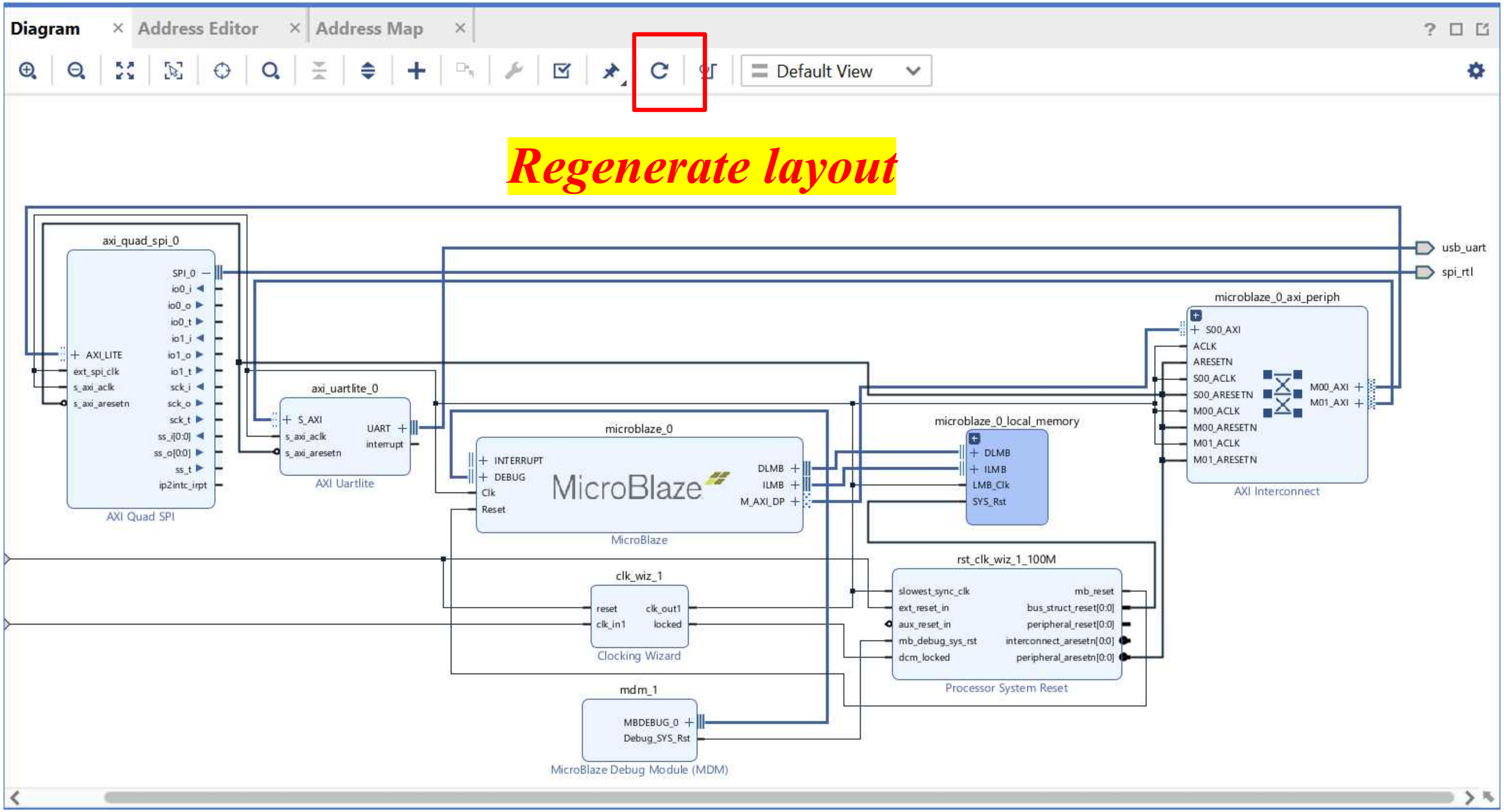
Ref_Clk1: Auto

Ref_Clk2: Auto

확인 후 *OK*

User Logic Interface Implementation

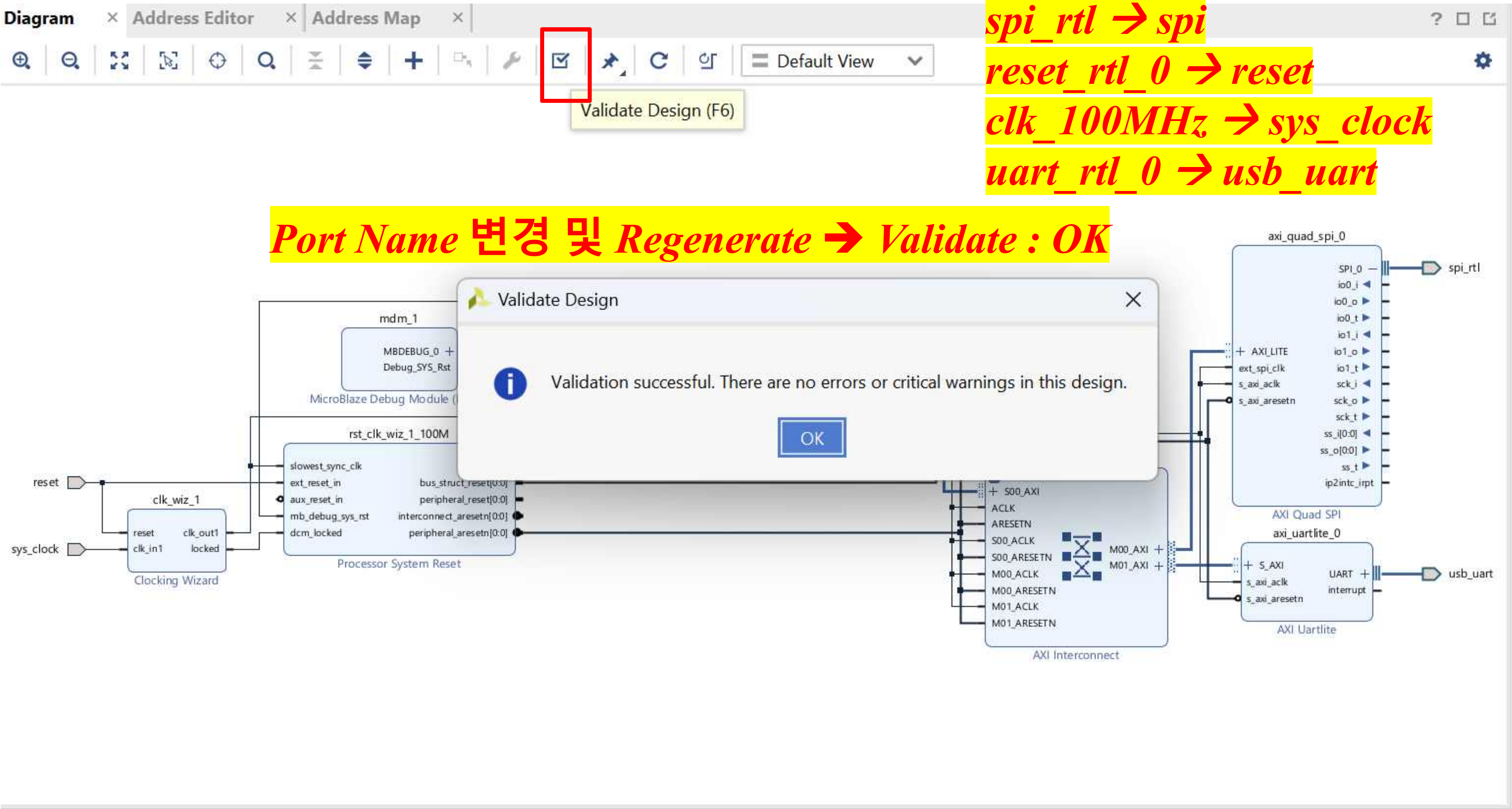
spi_master_ip_test.xpr



Regenerate layout

User Logic Interface Implementation

spi_master_ip_test.xpr



Port Name 변경

spi_rtl → spi

reset_rtl_0 → reset

clk_100MHz → sys_clock

uart_rtl_0 → usb_uart

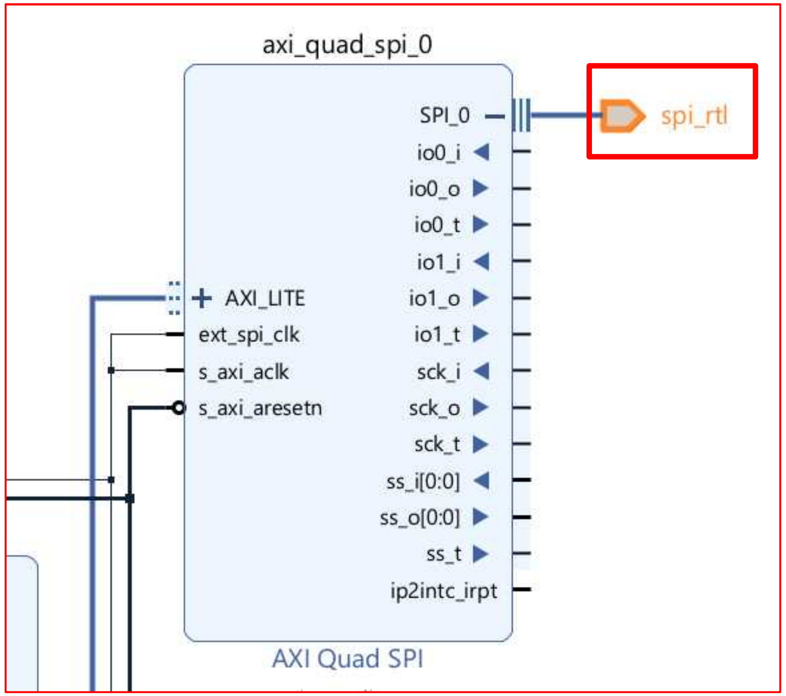
Port Name 변경 및 Regenerate → Validate : OK

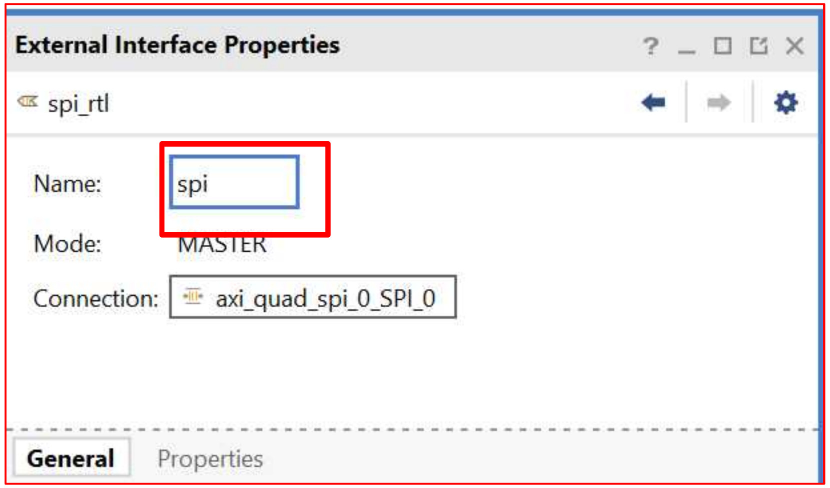
Validate Design

Validation successful. There are no errors or critical warnings in this design.

OK

User Logic Interface Implementation




External Interface Properties

spi_rtl

Name: spi

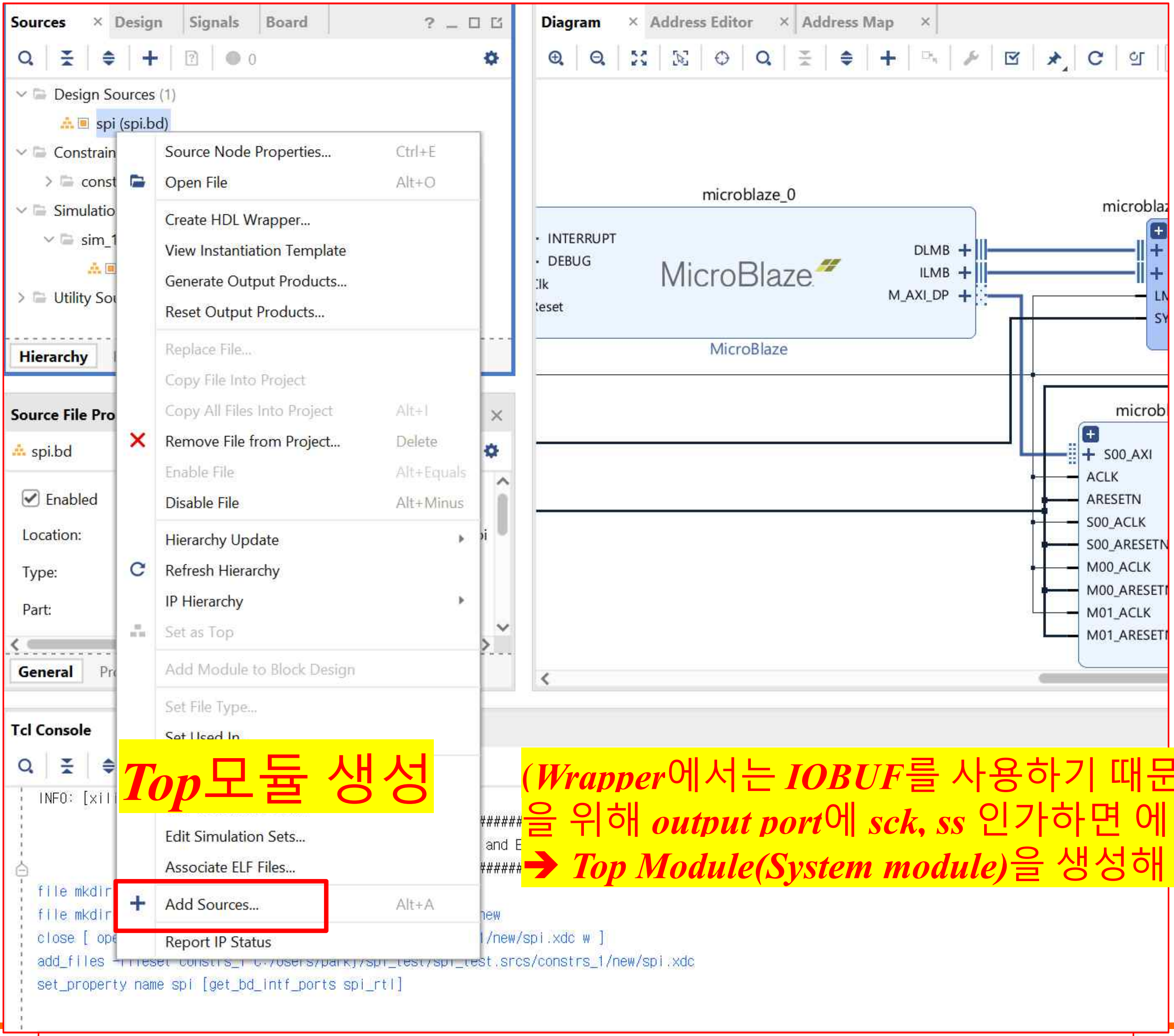
Mode: MASTER

Connection: axi_quad_spi_0_SPI_0

spi_rtl → spi 변경

User Logic Interface Implementation

spi_master_ip_test.xpr

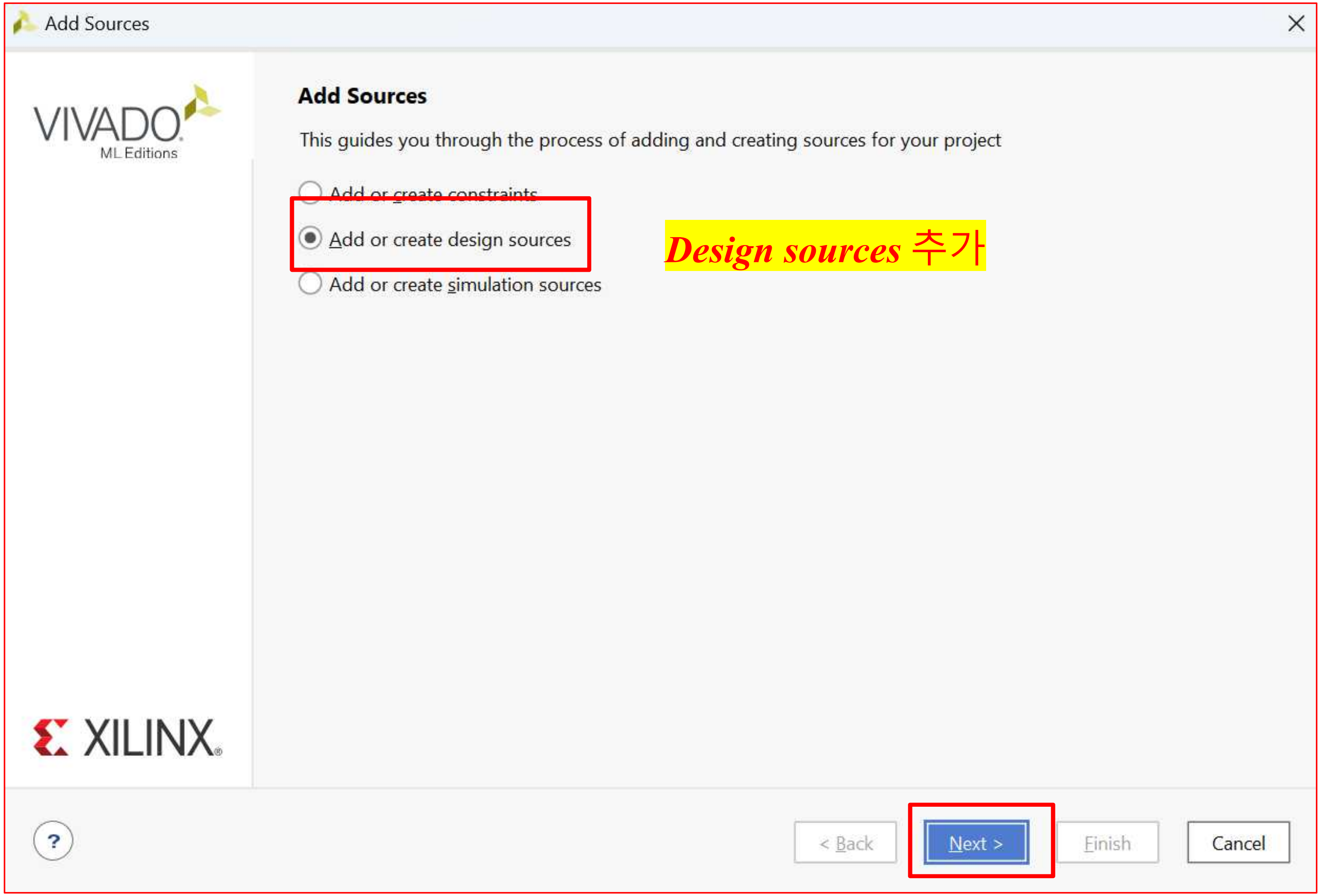


```
INFO: [xil
######
and E
######
file mkdir
file mkdir                                       new
close [ ope                                      1/new/spi.xdc w ]
add_files -fileset constrs_1 C:/Users/parkj/spi_test/spi_test.srcs/constrs_1/new/spi.xdc
set_property name spi [get_bd_intf_ports spi_rtl]
```

***Top*모듈 생성**

(*Wrapper*에서는 *IOBUF*를 사용하기 때문에 파형 출력을 위해 ***output port***에 ***sck, ss*** 인가하면 에러 발생
➔ ***Top Module(System module)***을 생성해 이용)

User Logic Interface Implementation



Add Sources

This guides you through the process of adding and creating sources for your project

○ Add or create constraints

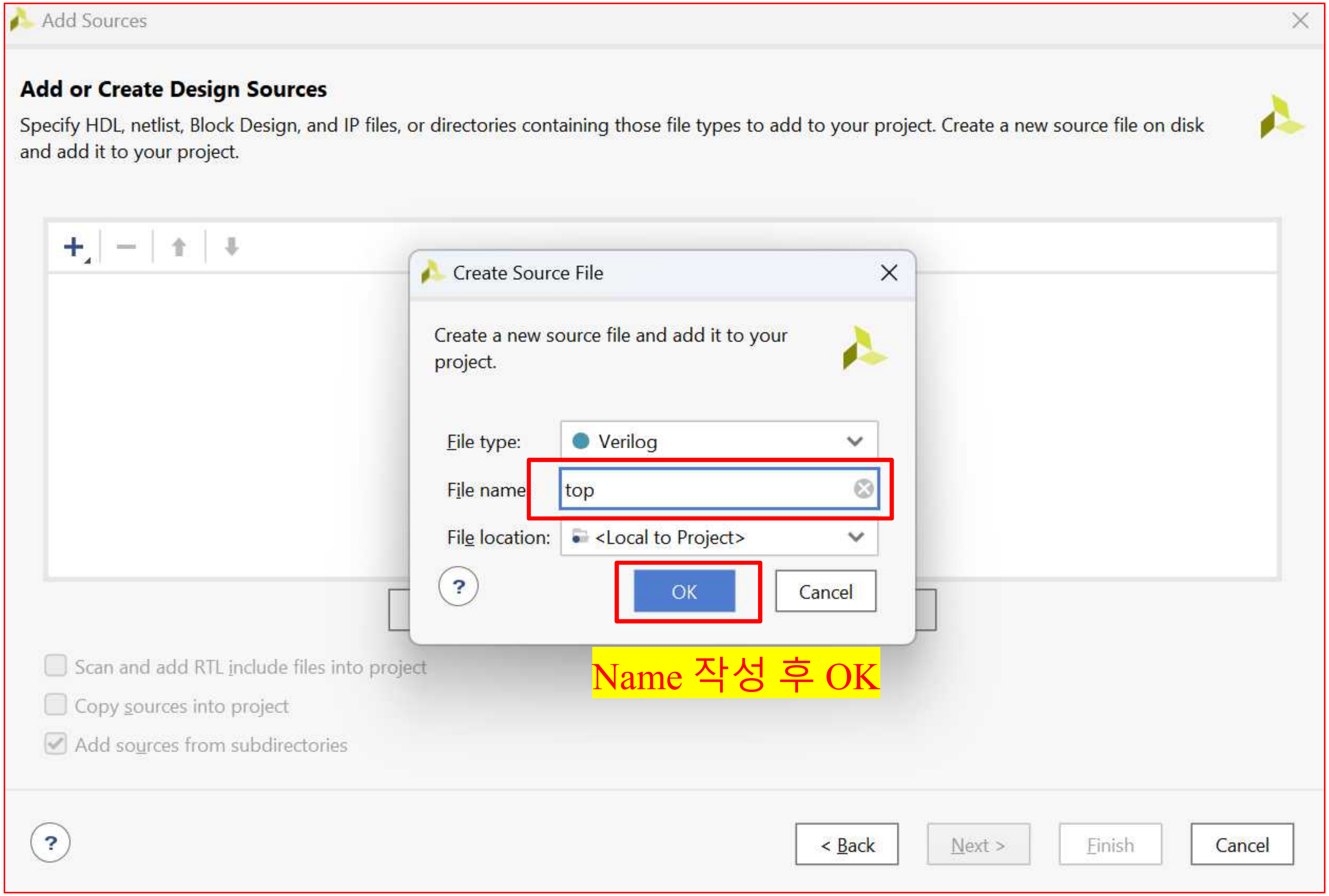
● Add or create design sources

○ Add or create simulation sources

Design sources 추가

User Logic Interface Implementation

spi_master_ip_test.xpr



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.

Create Source File

Create a new source file and add it to your project.

File type: Verilog

File name: top

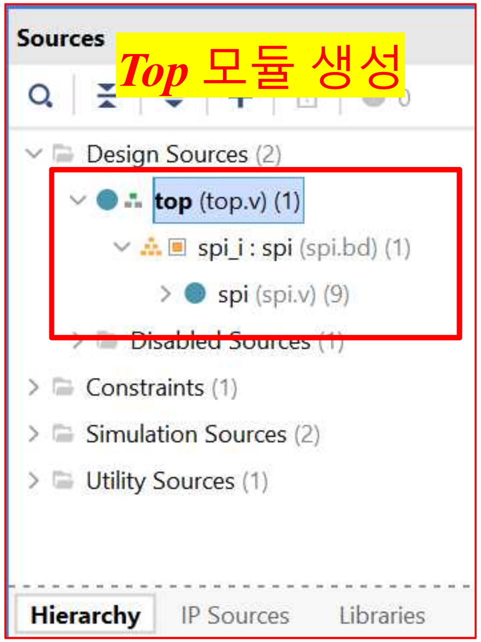
File location: <Local to Project>

Name 작성 후 OK

- □ Scan and add RTL include files into project
- □ Copy sources into project
- ☑ Add sources from subdirectories

User Logic Interface Implementation

spi_master_ip_test.xpr



Top 모듈 생성

Sources

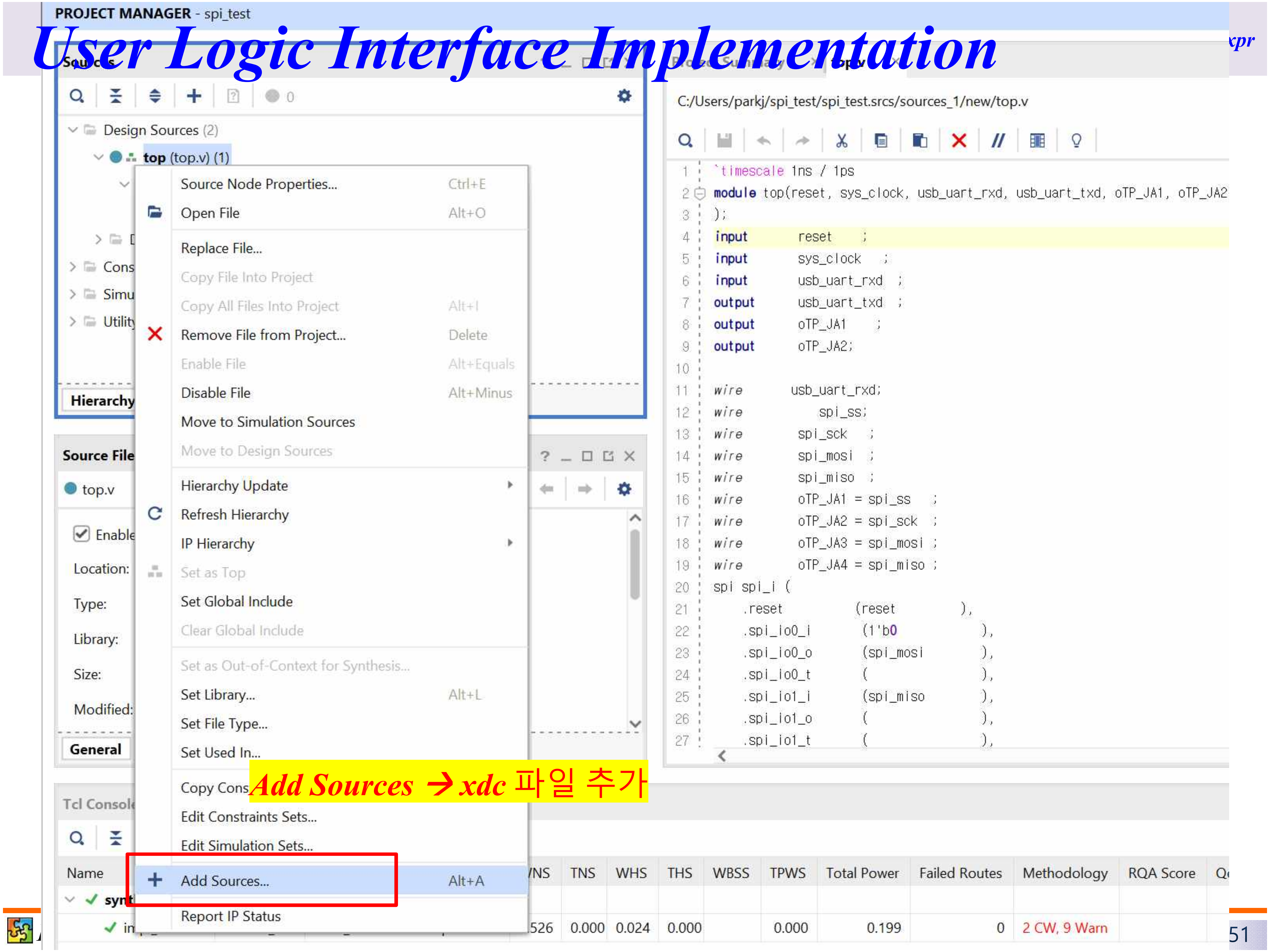
- Design Sources (2)
  - top (top.v) (1)
    - spi_i : spi (spi.bd) (1)
      - spi (spi.v) (9)
  - Disabled Sources (1)
- Constraints (1)
- Simulation Sources (2)
- Utility Sources (1)

Hierarchy | IP Sources | Libraries

```
`timescale 1ns / 1psmodule top(
reset,
sys_clock,
usb_uart_rxd,
usb_uart_txd,
oTP_JA1,
oTP_JA2
);

input       reset    ;
input       sys_clock   ;
input       usb_uart_rxd  ;
output      usb_uart_txd  ;
output      oTP_JA1    ;
output      oTP_JA2;
wire        usb_uart_rxd;
wire        spi_ss;
wire        spi_sck   ;
wire        spi_mosi  ;
wire        spi_miso  ;
wire        oTP_JA1 = spi_ss   ;
wire        oTP_JA2 = spi_sck  ;
wire        oTP_JA3 = spi_mosi ;
wire        oTP_JA4 = spi_miso ;
spi spi_i (
            .reset        (reset        ),
            .spi_io0_i     (1'b0          ),
            .spi_io0_o     (spi_mosi      ),
            .spi_io0_t     (              ),
            .spi_io1_i     (spi_miso      ),
            .spi_io1_o     (              ),
            .spi_io1_t     (              ),
            .spi_sck_i     (1'b0          ),
            .spi_sck_o     (spi_sck       ),
            .spi_sck_t     (              ),
            .spi_ss_i      (1'b0          ),
            .spi_ss_o      (spi_ss        ),
            .spi_ss_t      (              ),
            .sys_clock     (sys_clock     ),
            .usb_uart_rxd  (usb_uart_rxd      ),
            .usb_uart_txd  (usb_uart_txd      )
);
endmodule
```

➔ Bitstream 후 xsa파일 Export 하여 Vitis 실행

User Logic Interface Implementation



C:/Users/parkj/spi_test/spi_test.srcs/sources_1/new/top.v

```
`timescale 1ns / 1ps
module top(reset, sys_clock, usb_uart_rxd, usb_uart_txd, oTP_JA1, oTP_JA2
);
input       reset     ;
input       sys_clock    ;
input       usb_uart_rxd  ;
output      usb_uart_txd  ;
output      oTP_JA1     ;
output      oTP_JA2;

wire       usb_uart_rxd;
wire          spi_ss;
wire        spi_sck    ;
wire        spi_mosi   ;
wire        spi_miso   ;
wire        oTP_JA1 = spi_ss    ;
wire        oTP_JA2 = spi_sck   ;
wire        oTP_JA3 = spi_mosi ;
wire        oTP_JA4 = spi_miso ;
spi spi_i (
    .reset          (reset          ),
    .spi_io0_i       (1'b0            ),
    .spi_io0_o       (spi_mosi        ),
    .spi_io0_t       (                ),
    .spi_io1_i       (spi_miso        ),
    .spi_io1_o       (                ),
    .spi_io1_t       (                ),
```

Name	NS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	Q
✓ synt											
✓ in	526	0.000	0.024	0.000		0.000	0.199	0	2 CW, 9 Warn		

Add Sources → xdc 파일 추가

User Logic Interface Implementation

SPI Master IP TEST

- XDC Constraints

Basys-3-Master.xdc

```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports sys_clock]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports sys_clock]

set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {reset}]


##Pmod Header JA
set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports oTP_JA1];#Sch name = JA1
set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports oTP_JA2];#Sch name = JA2
#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name = JA3
#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name = JA4
#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name = JA7
#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name = JA8
#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name = JA9
#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name = JA10

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

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

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

User Logic Interface Implementation

➢ ***SPI Master IP TEST → Sequency***

- ***Create Block Design***
- ***Blaze Uart & quad_SPI 추가 및 연결***
- ***sys_clock & reset 설정***
- ***SPI ss, sck Make External***
- ***XDC Constraints 설정***
- ***Bitstream***
- ***Export Hardwar***

User Logic Interface Implementation

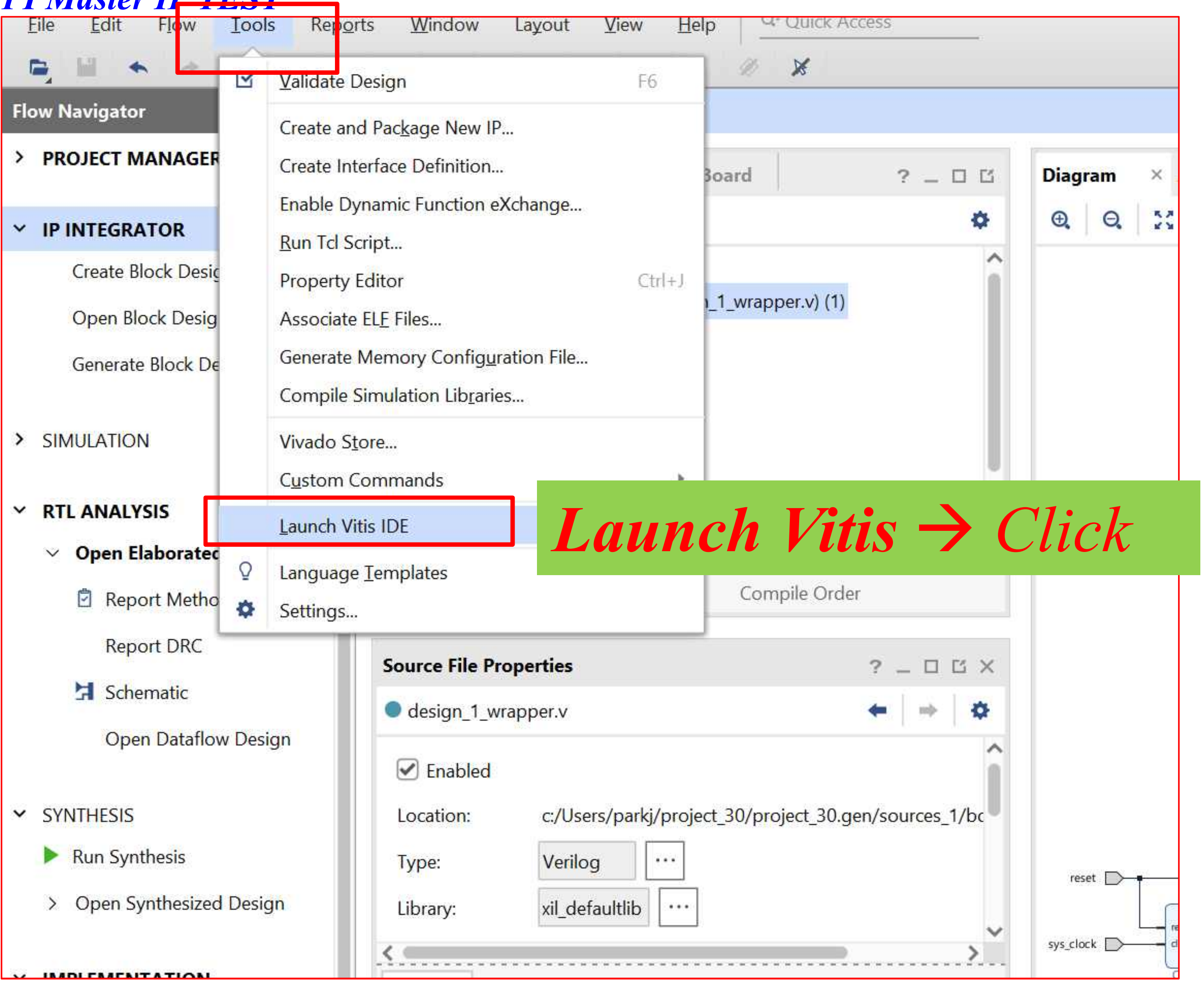
[*SPI Master IP TEST*]

✓ *Vitis*

User Logic Interface Implementation

spi_master_ip_test.xpr

- ***SPI Master IP TEST***

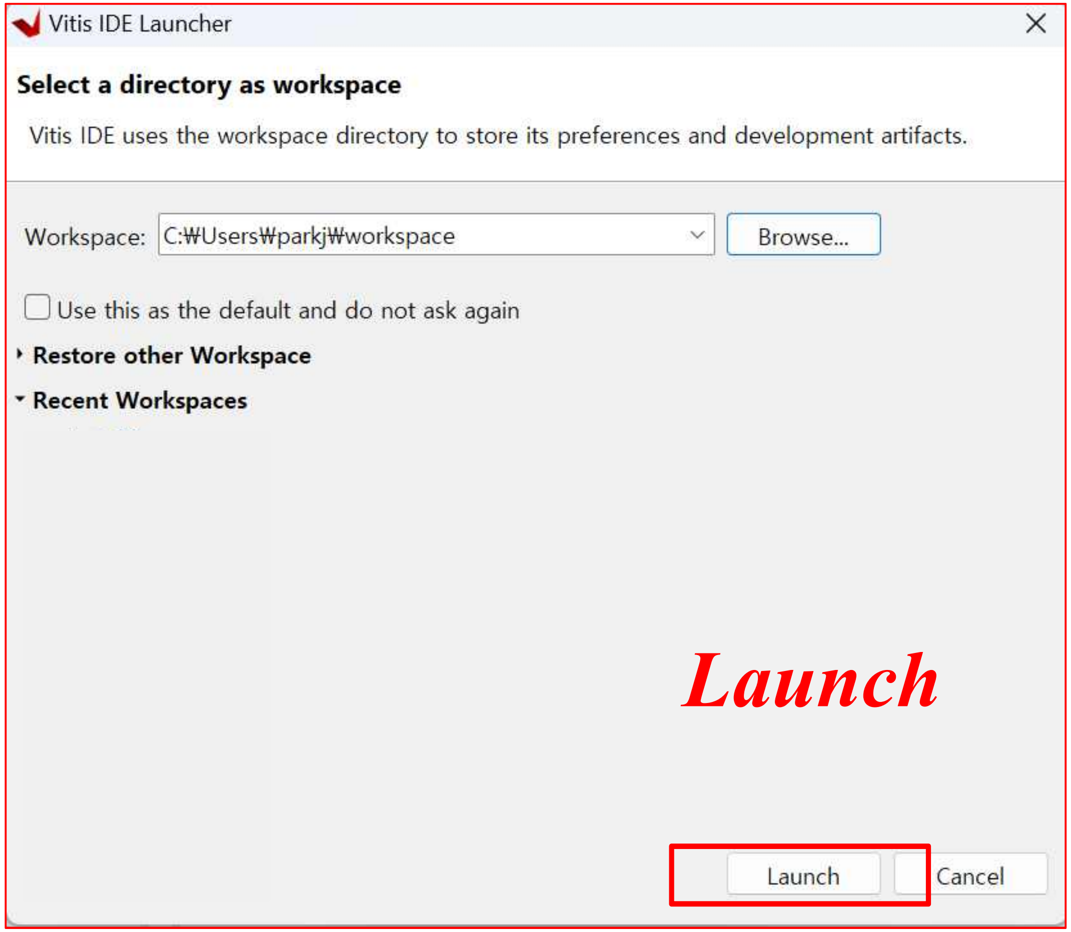


Launch Vitis → Click

User Logic Interface Implementation

spi_master_ip_test.xpr

- ➢ *SPI Master IP TEST*



Vitis IDE Launcher

Select a directory as workspace

Vitis IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:₩Users₩parkj₩workspace

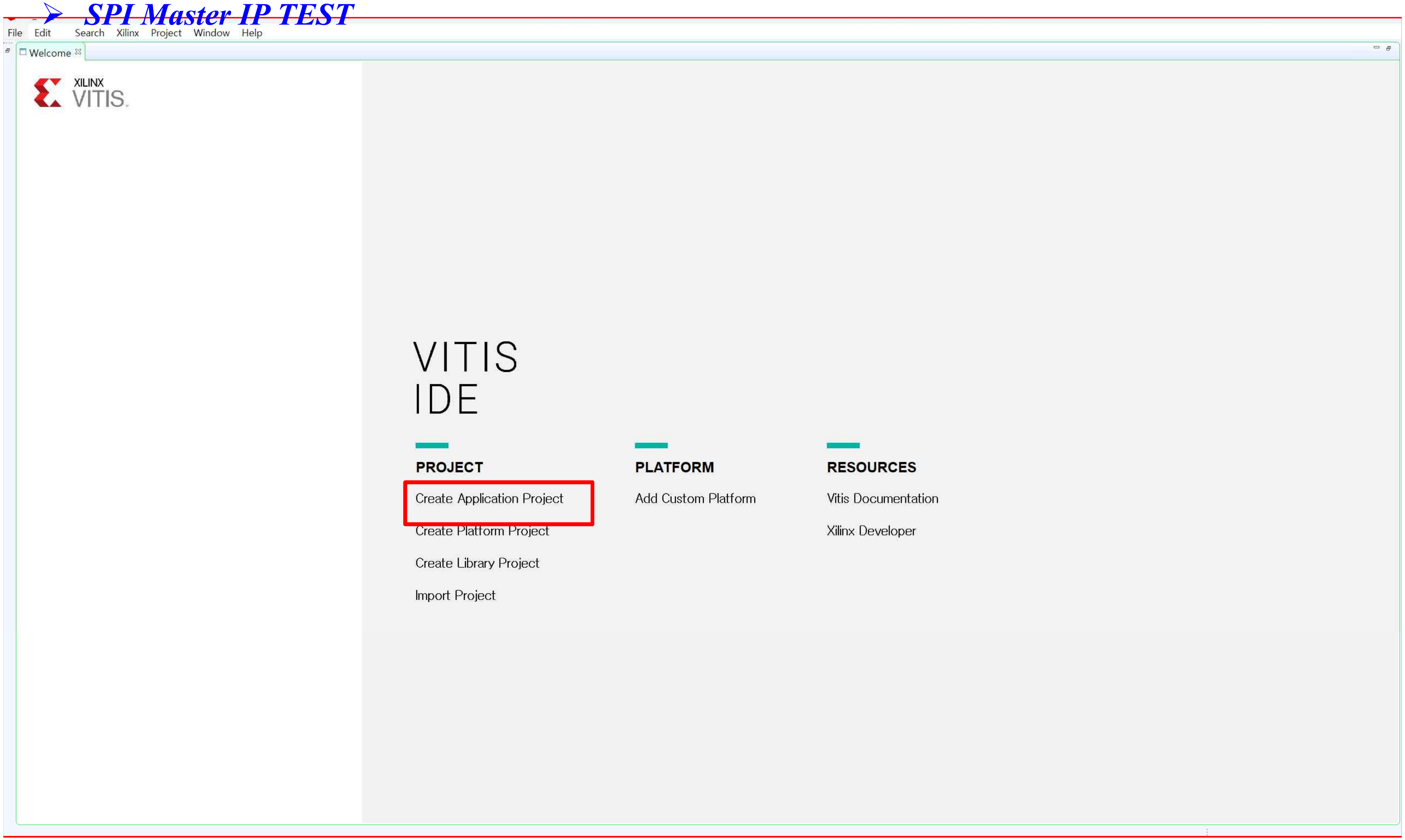
□ Use this as the default and do not ask again

Launch

User Logic Interface Implementation

spi_master_ip_test.xpr

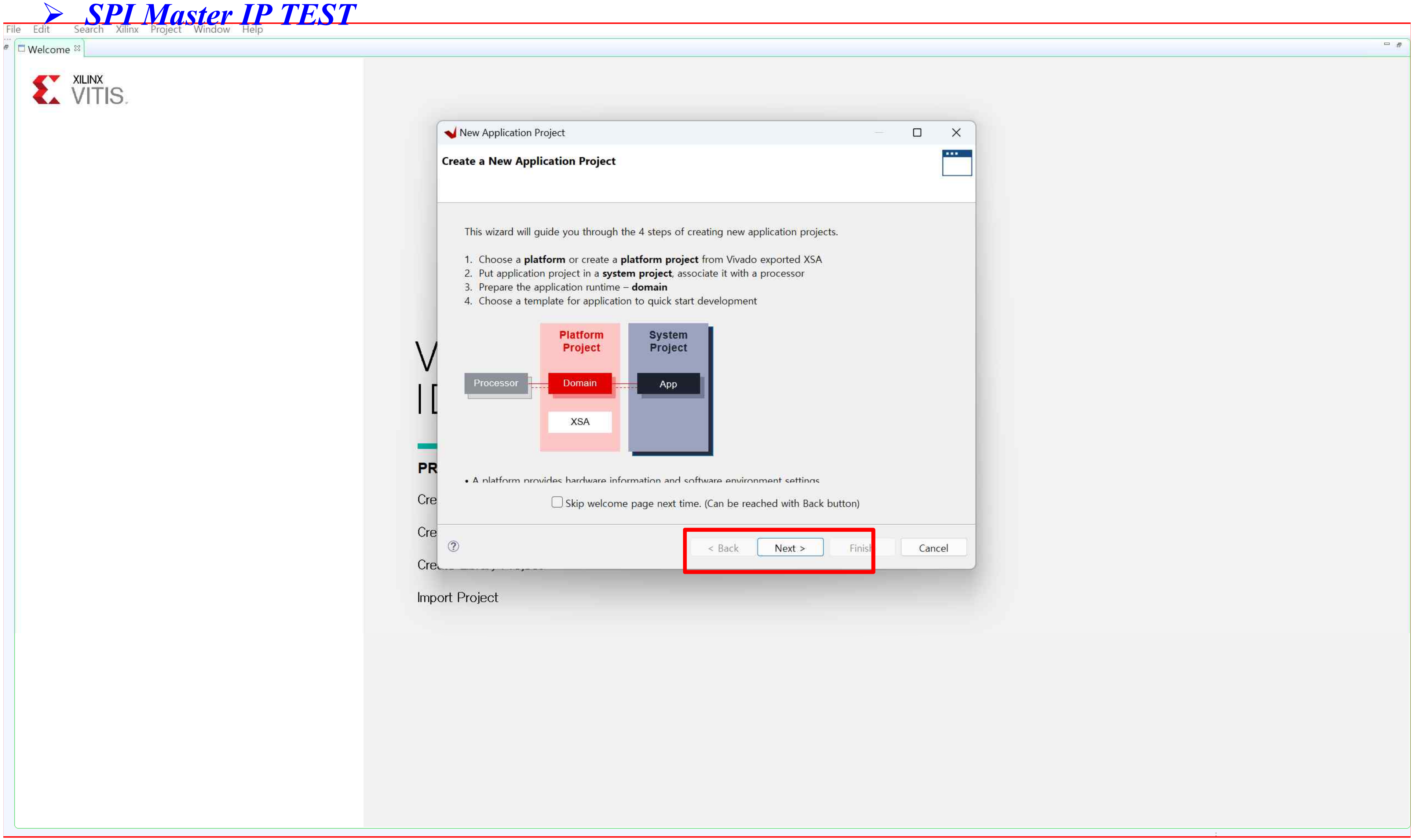
- ***SPI Master IP TEST***



User Logic Interface Implementation

spi_master_ip_test.xpr

➢ SPI Master IP TEST



New Application Project

Create a New Application Project

This wizard will guide you through the 4 steps of creating new application projects.

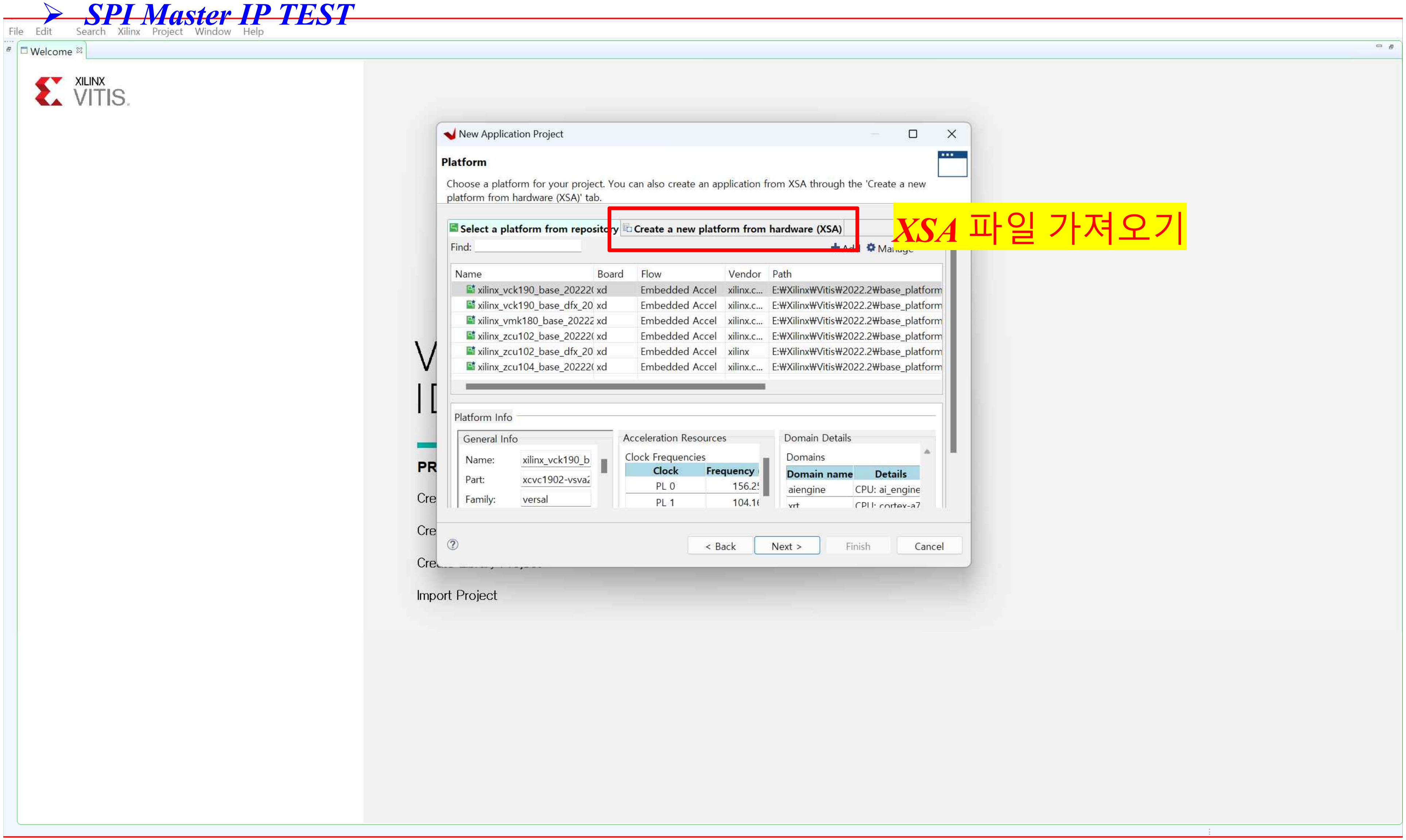
1. Choose a **platform** or create a **platform project** from Vivado exported XSA
2. Put application project in a **system project**, associate it with a processor
3. Prepare the application runtime – **domain**
4. Choose a template for application to quick start development

- A platform provides hardware information and software environment settings

□ Skip welcome page next time. (Can be reached with Back button)

User Logic Interface Implementation

➤ SPI Master IP TEST



New Application Project

Platform

Choose a platform for your project. You can also create an application from XSA through the 'Create a new platform from hardware (XSA)' tab.

Select a platform from repository | Create a new platform from hardware (XSA)

XSA 파일 가져오기

Find:

Name	Board	Flow	Vendor	Path
xilinx_vck190_base_20222(	xd	Embedded Accel	xilinx.c...	E:₩Xilinx₩Vitis₩2022.2₩base_platform
xilinx_vck190_base_dfx_20	xd	Embedded Accel	xilinx.c...	E:₩Xilinx₩Vitis₩2022.2₩base_platform
xilinx_vmk180_base_20222	xd	Embedded Accel	xilinx.c...	E:₩Xilinx₩Vitis₩2022.2₩base_platform
xilinx_zcu102_base_20222(	xd	Embedded Accel	xilinx.c...	E:₩Xilinx₩Vitis₩2022.2₩base_platform
xilinx_zcu102_base_dfx_20	xd	Embedded Accel	xilinx	E:₩Xilinx₩Vitis₩2022.2₩base_platform
xilinx_zcu104_base_20222(	xd	Embedded Accel	xilinx.c...	E:₩Xilinx₩Vitis₩2022.2₩base_platform

Platform Info

General Info

Name: xilinx_vck190_b

Part: xcvc1902-vsva2

Family: versal

Acceleration Resources

Clock Frequencies

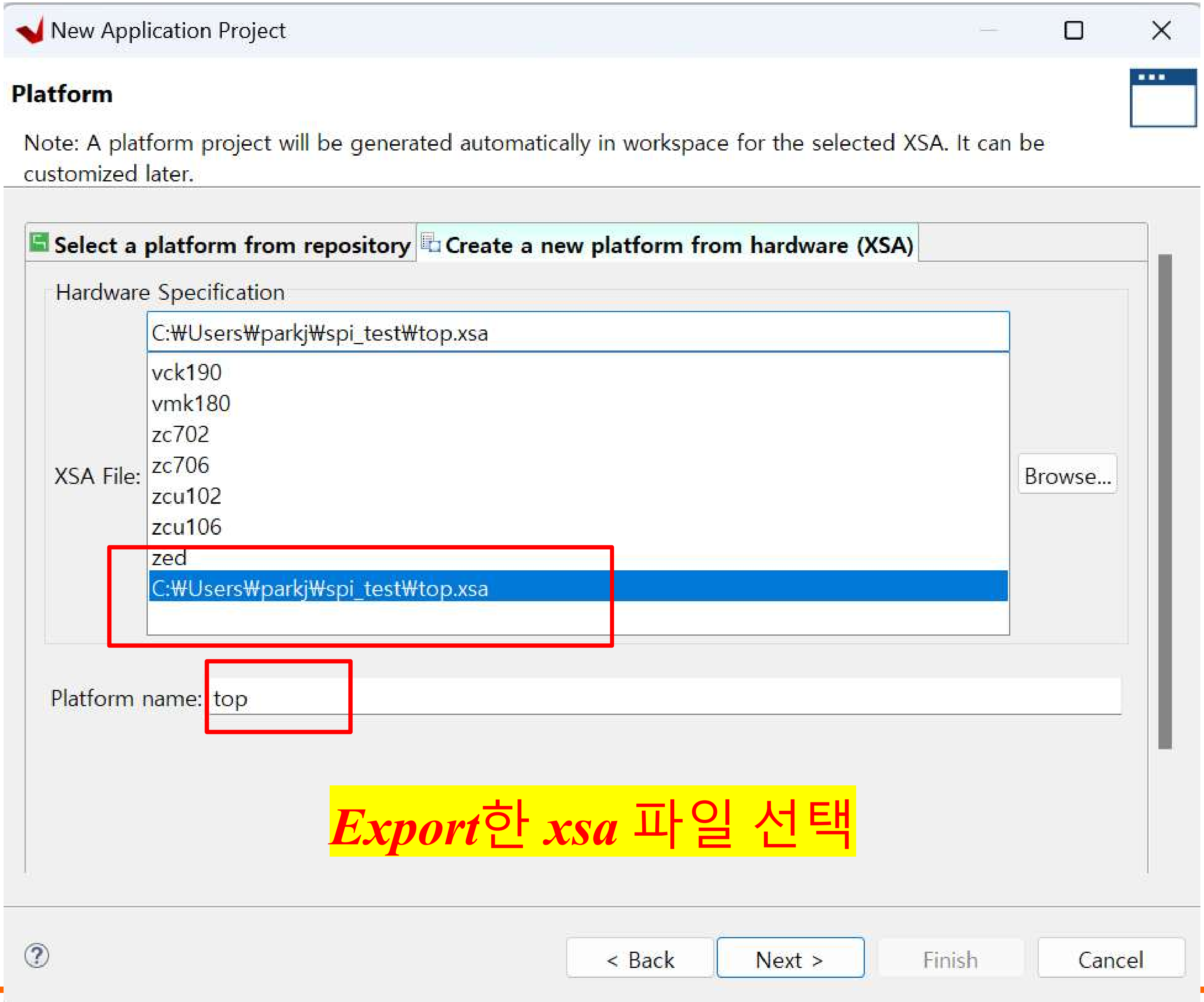
Clock	Frequency
PL 0	156.2!
PL 1	104.1(

Domain Details

Domains

Domain name	Details
aiengine	CPU: ai_engine
xrt	CPU: cortex-a7

User Logic Interface Implementation



New Application Project

Platform

Note: A platform project will be generated automatically in workspace for the selected XSA. It can be customized later.

Select a platform from repository | Create a new platform from hardware (XSA)

Hardware Specification

XSA File: C:₩Users₩parkj₩spi_test₩top.xsa

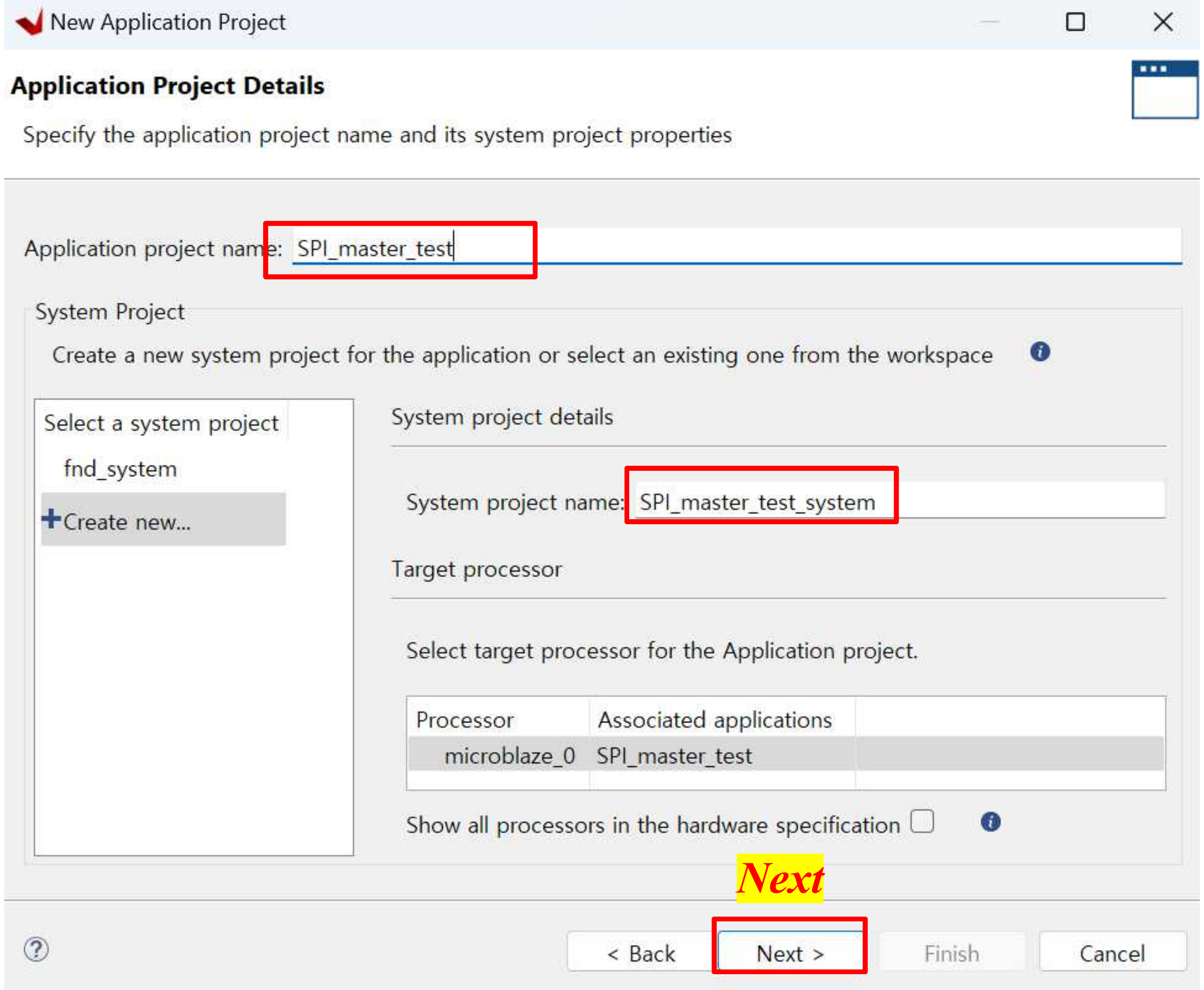
vck190
vmk180
zc702
zc706
zcu102
zcu106
zed
C:₩Users₩parkj₩spi_test₩top.xsa

Platform name: top

Export한 xsa 파일 선택

User Logic Interface Implementation

- ***SPI Master IP TEST***



New Application Project

Application Project Details

Specify the application project name and its system project properties

Application project name: SPI_master_test

System Project

Create a new system project for the application or select an existing one from the workspace

Select a system project

fnd_system

+Create new...

System project details

System project name: SPI_master_test_system

Target processor

Select target processor for the Application project.

Processor	Associated applications
microblaze_0	SPI_master_test

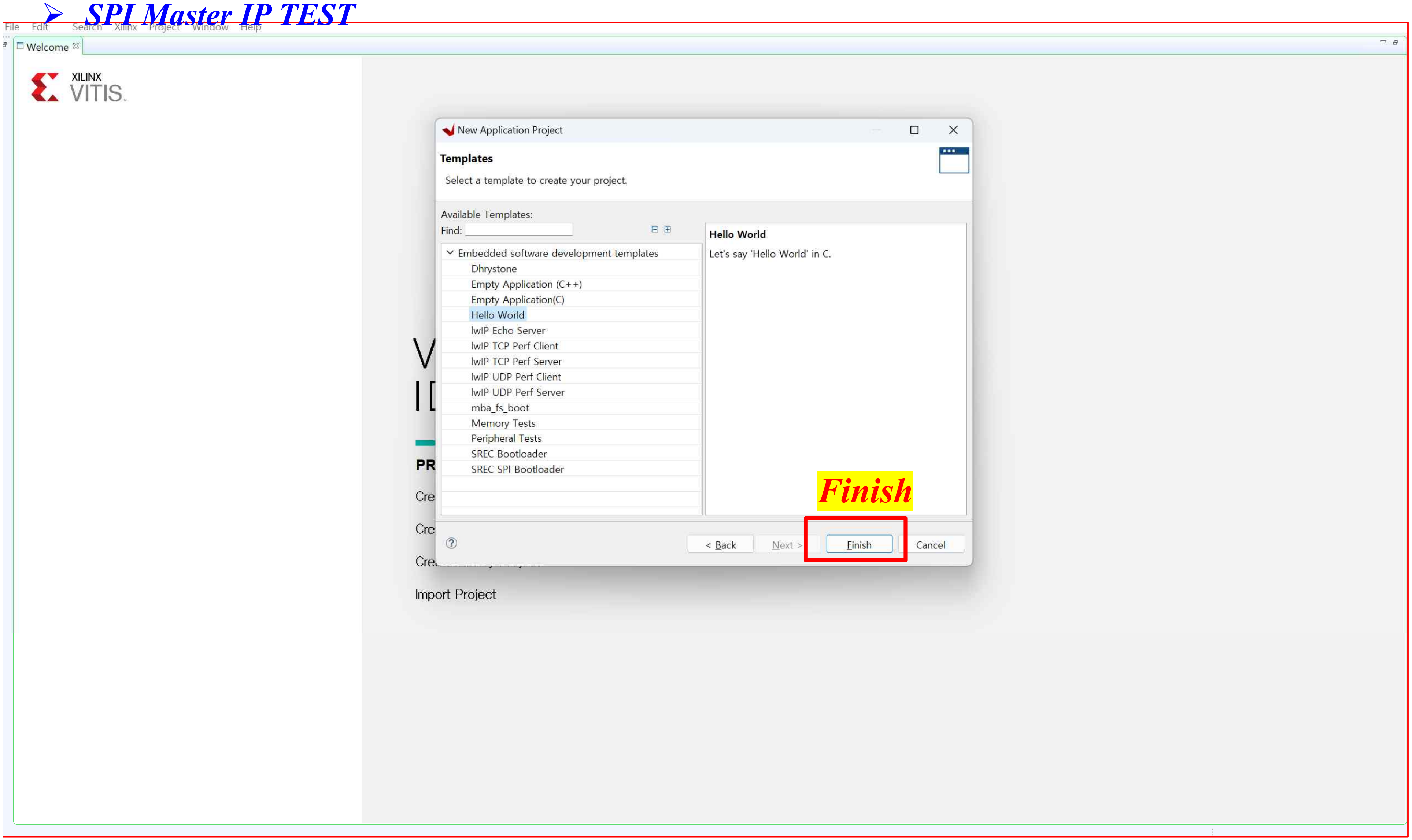
Show all processors in the hardware specification □

Next

User Logic Interface Implementation

spi_master_ip_test.xpr

➢ SPI Master IP TEST



New Application Project

Templates

Select a template to create your project.

Available Templates:

Find:

- Embedded software development templates
  - Dhrystone
  - Empty Application (C++)
  - Empty Application(C)
  - Hello World
  - lwIP Echo Server
  - lwIP TCP Perf Client
  - lwIP TCP Perf Server
  - lwIP UDP Perf Client
  - lwIP UDP Perf Server
  - mba_fs_boot
  - Memory Tests
  - Peripheral Tests
  - SREC Bootloader
  - SREC SPI Bootloader

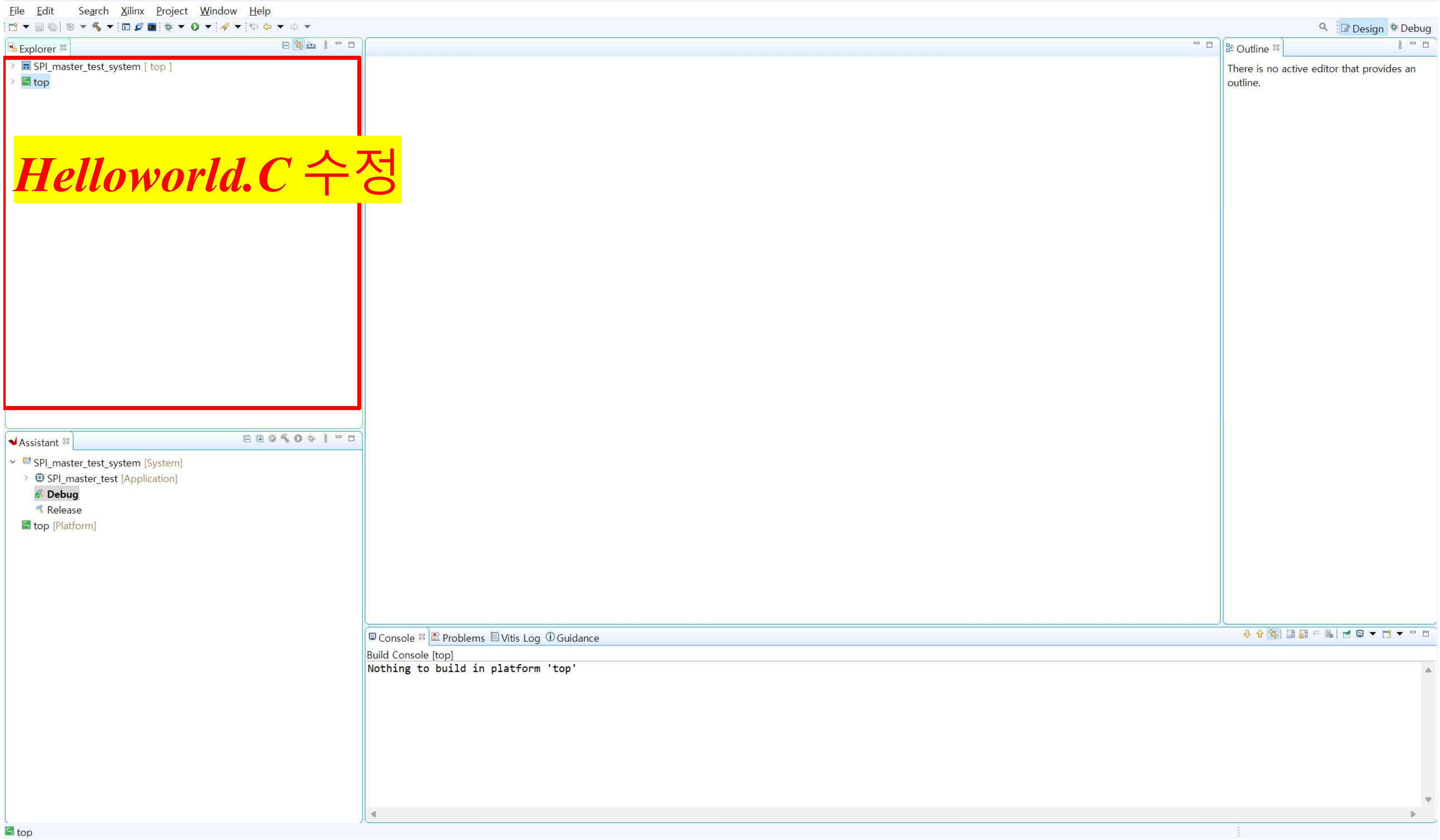
Hello World

Let's say 'Hello World' in C.

Finish

User Logic Interface Implementation

spi_master_ip_test.xpr


File Edit Search Xilinx Project Window Help
Design
Debug
Explorer
SPI_master_test_system [top]
top
Outline
Assistant
SPI_master_test_system [System]
SPI_master_test [Application]
Debug
Release
top [Platform]
Console
Problems
Vitis Log
Guidance
top

There is no active editor that provides an outline.

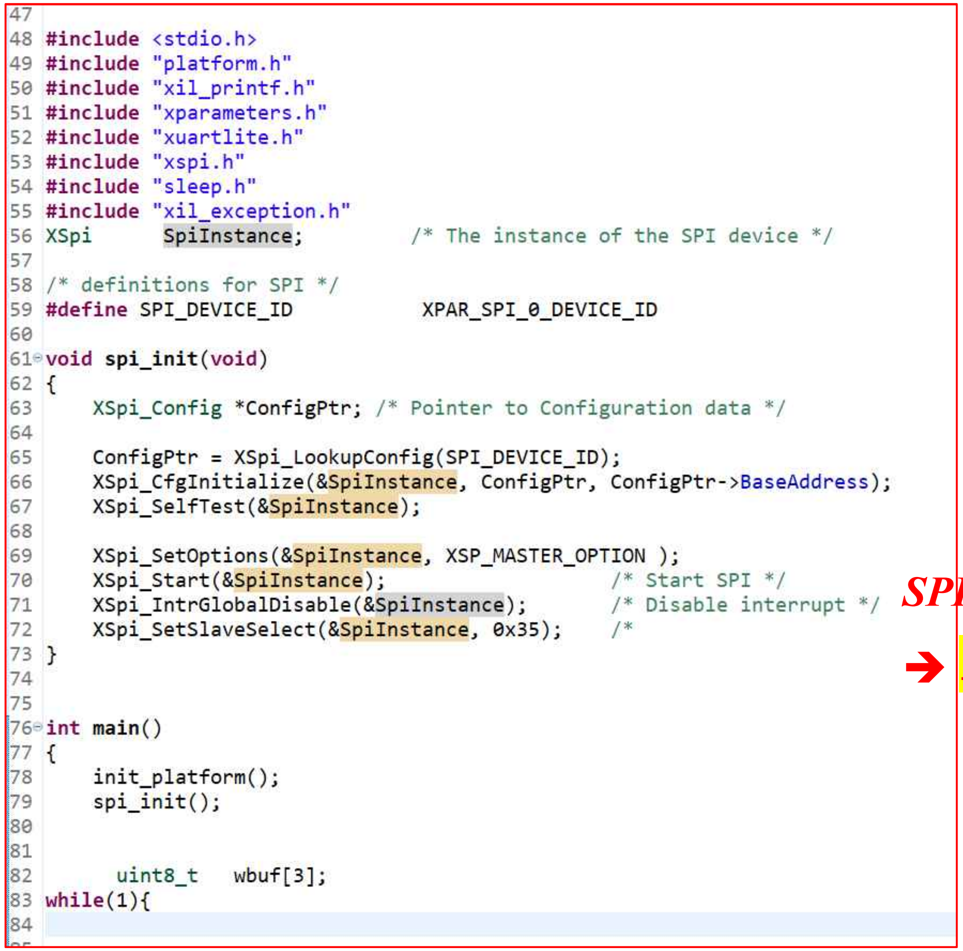
Build Console [top]

```
Nothing to build in platform 'top'
```

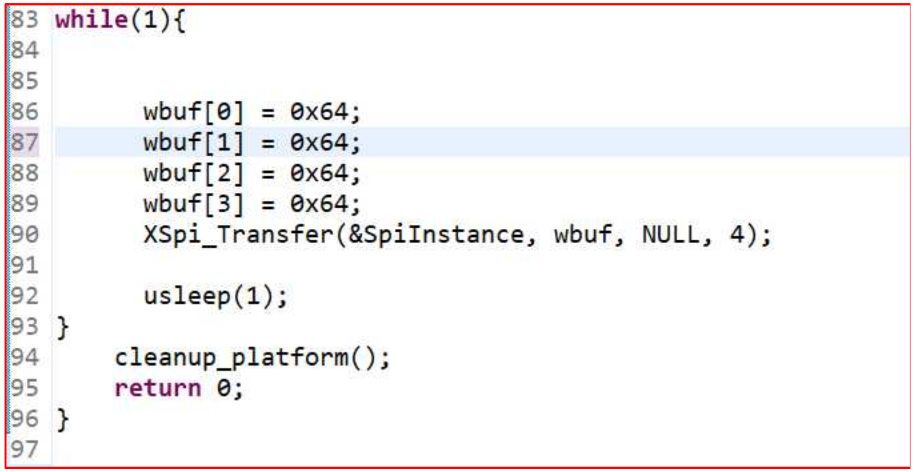
Helloworld.C 수정

User Logic Interface Implementation

spi_master_ip_test.xpr



```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xparameters.h"
#include "xuartlite.h"
#include "xspi.h"
#include "sleep.h"
#include "xil_exception.h"
XSpi      SpiInstance;         /* The instance of the SPI device */

/* definitions for SPI */
#define SPI_DEVICE_ID           XPAR_SPI_0_DEVICE_ID

void spi_init(void)
{
    XSpi_Config *ConfigPtr; /* Pointer to Configuration data */

    ConfigPtr = XSpi_LookupConfig(SPI_DEVICE_ID);
    XSpi_CfgInitialize(&SpiInstance, ConfigPtr, ConfigPtr->BaseAddress);
    XSpi_SelfTest(&SpiInstance);

    XSpi_SetOptions(&SpiInstance, XSP_MASTER_OPTION );
    XSpi_Start(&SpiInstance);                /* Start SPI */
    XSpi_IntrGlobalDisable(&SpiInstance);    /* Disable interrupt */
    XSpi_SetSlaveSelect(&SpiInstance, 0x35);     /*
}


int main()
{
    init_platform();
    spi_init();


      uint8_t   wbuf[3];
while(1){

```



```
while(1){


       wbuf[0] = 0x64;
       wbuf[1] = 0x64;
       wbuf[2] = 0x64;
       wbuf[3] = 0x64;
       XSpi_Transfer(&SpiInstance, wbuf, NULL, 4);

       usleep(1);
}
     cleanup_platform();
     return 0;
}

```

*SPI*로 임의의 4바이트 출력

➔ 오실로스코프 활용해 SPI master 동작 원리 확인.

***Helloworld.C* 수정**

User Logic Interface Implementation

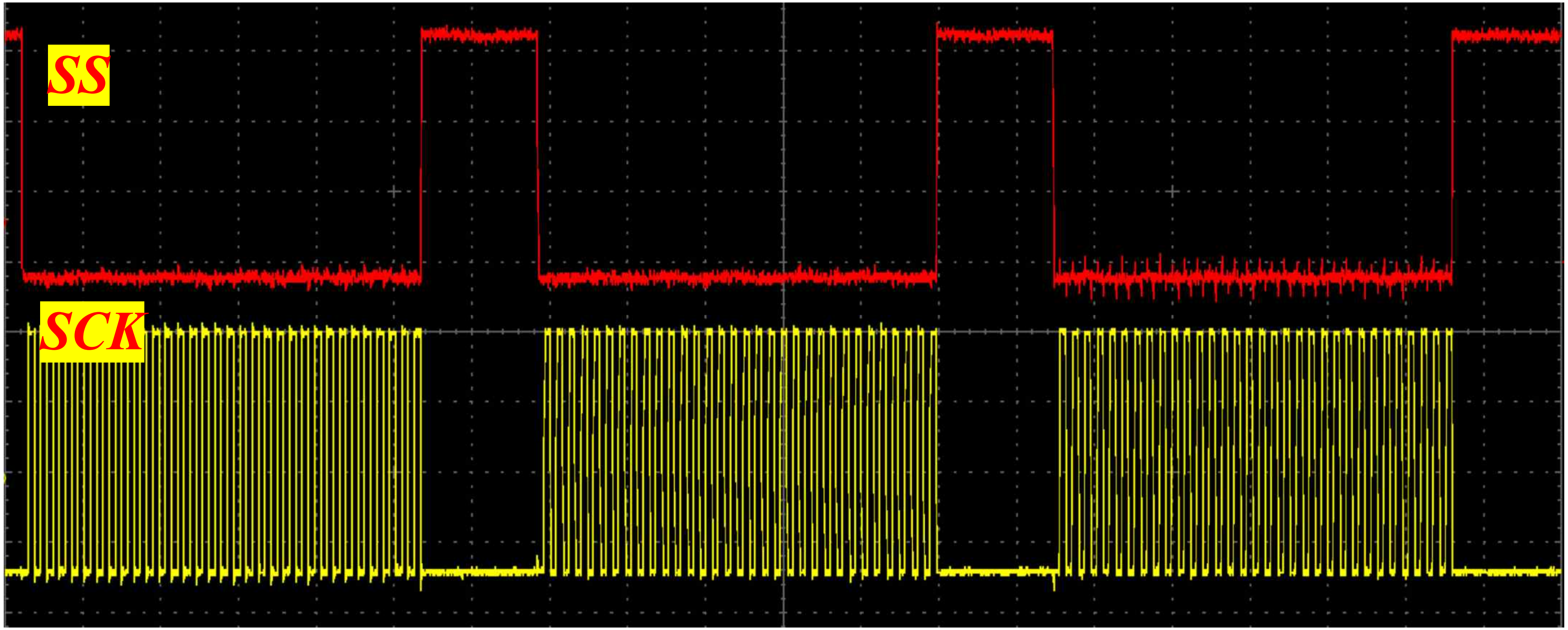
spi_master_ip_test.xpr

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "xparameters.h"
#include "xuartlite.h"
#include "xspi.h"
#include "sleep.h"
#include "xil_exception.h “

XSpi     SpiInstance;      /* The instance of the SPI device */

/* definitions for SPI */
#define SPI_DEVICE_ID    XPAR_SPI_0_DEVICE_ID

void spi_init(void)
{
XSpi_Config *ConfigPtr;/* Pointer to Configuration data */

ConfigPtr = XSpi_LookupConfig(SPI_DEVICE_ID);
XSpi_CfgInitialize(&SpiInstance, ConfigPtr, ConfigPtr->BaseAddress);
XSpi_SelfTest(&SpiInstance);

XSpi_SetOptions(&SpiInstance, XSP_MASTER_OPTION );
XSpi_Start(&SpiInstance);              /* Start SPI */
XSpi_IntrGlobalDisable(&SpiInstance);/* Disable interrupt */
XSpi_SetSlaveSelect(&SpiInstance, 0x35);/*                         */
}
```

```
int main()

{
    init_platform();

    spi_init();

   uint8_t   wbuf[3];

   while(1){
               wbuf[0] = 0x64;
               wbuf[1] = 0x64;
               wbuf[2] = 0x64;
               wbuf[3] = 0x64;
               XSpi_Transfer(&SpiInstance, wbuf, NULL, 4);
               usleep(1);
}
    cleanup_platform();
    return 0;
}
```

User Logic Interface Implementation

spi_master_ip_test.xpr

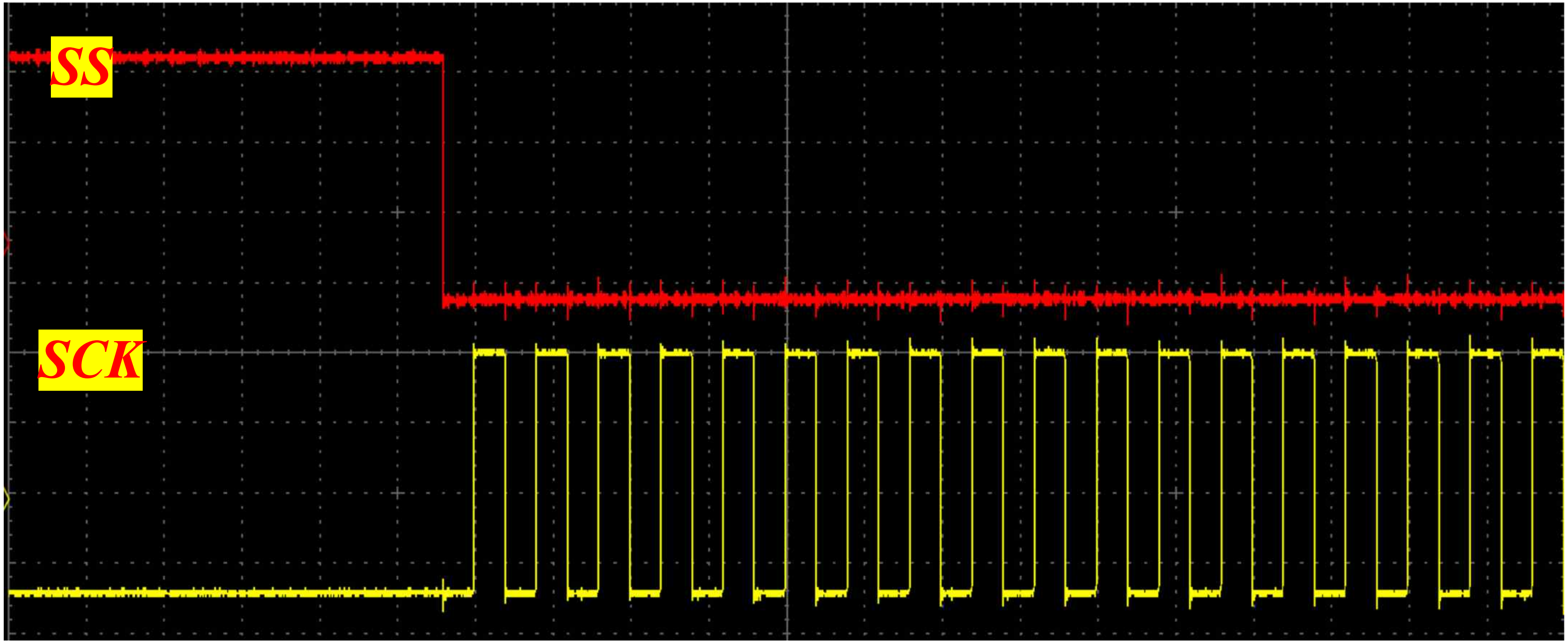


M 10us /div

SS가 LOW 일 때 SCK 파형이 발생함을 확인할 수 있음

User Logic Interface Implementation

spi_master_ip_test.xpr

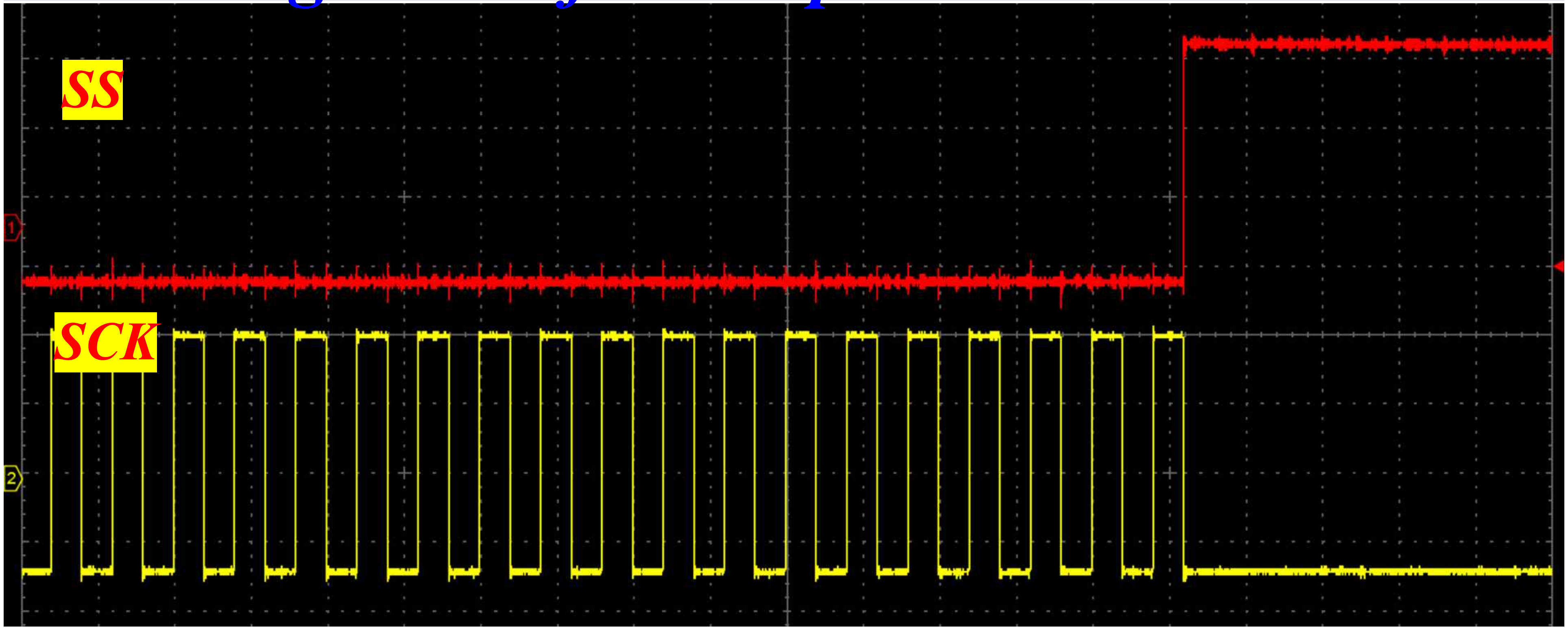


1. 전송 시작 부분

***SS*가 *LOW* → 일정 *delay* 후에 *SCK*가 발생(*HIGH*)**

User Logic Interface Implementation

spi_master_ip_test.xpr



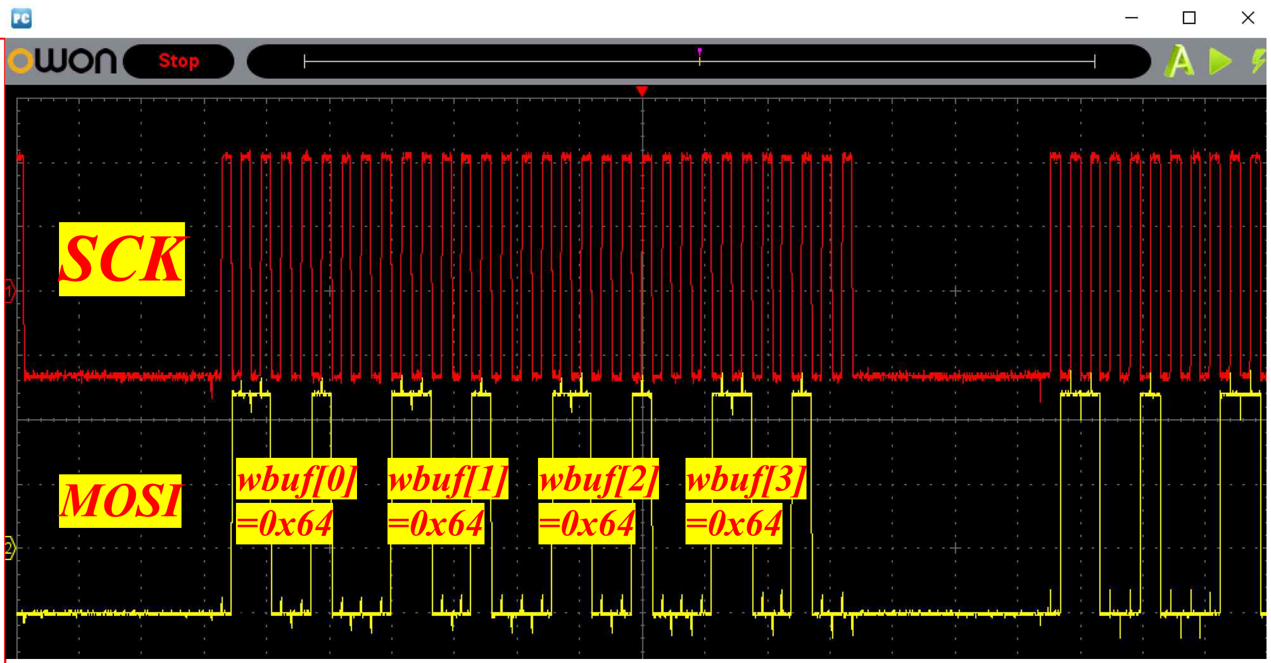
2. 전송 종료 부분

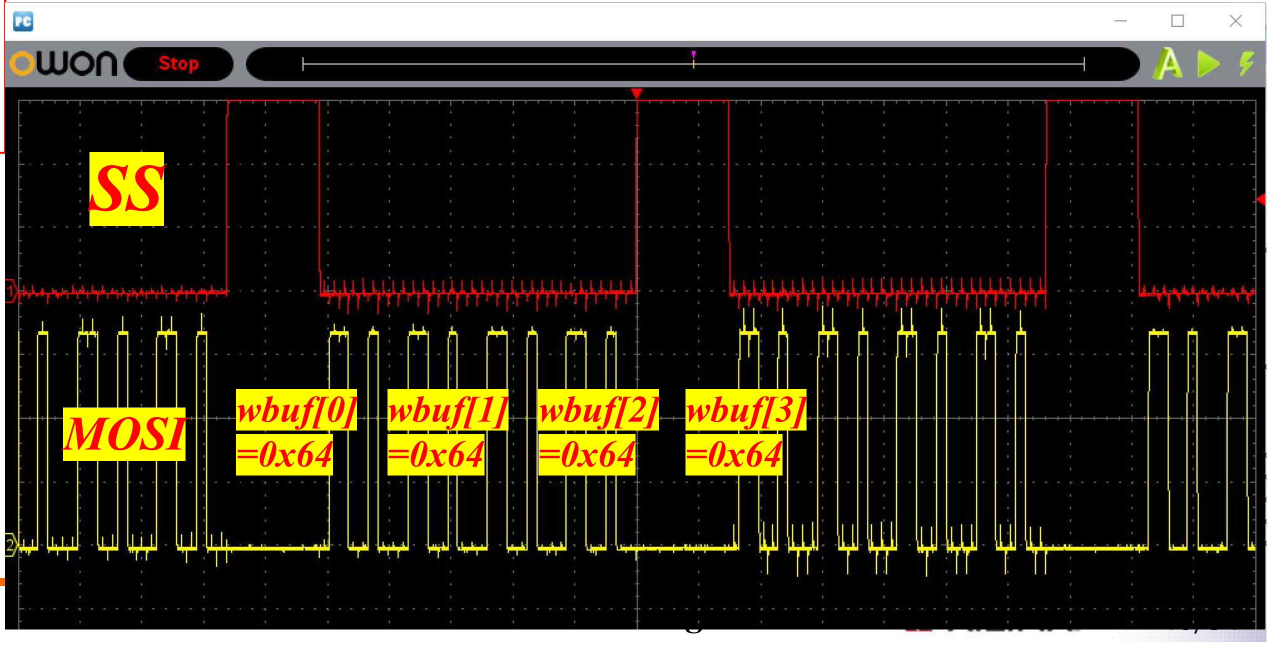
***SS*의 *HIGH*와 *SCK*의 *LOW*가 동시에 발생**

User Logic Interface Implementation

spi_master_ip_test.xpr

```
int main()

{
    init_platform();

    spi_init();

    uint8_t  wbuf[3];

    while(1){
                wbuf[0] = 0x64;
                wbuf[1] = 0x64;
                wbuf[2] = 0x64;
                wbuf[3] = 0x64;
                XSpi_Transfer(&SpiInstance, wbuf, NULL, 4);
                usleep(1);
}
    cleanup_platform();
    return 0;
}
```

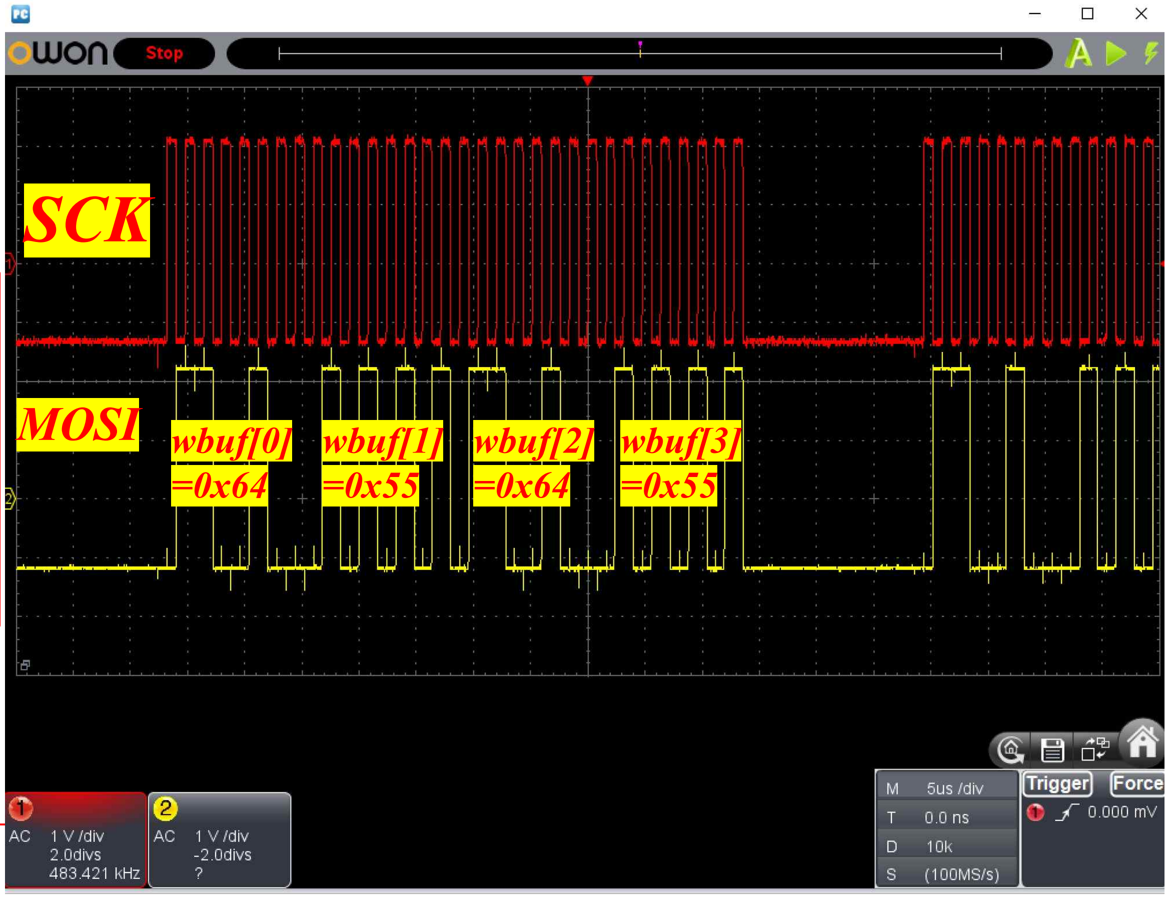




User Logic Interface Implementation

spi_master_ip_test.xpr

```
int main()

{
    init_platform();

    spi_init();

    uint8_t   wbuf[3];

    while(1){
            wbuf[0] = 0x64;
            wbuf[1] = 0x55;
            wbuf[2] = 0x64;
            wbuf[3] = 0x55;
            XSpi_Transfer(&SpiInstance, wbuf, NULL, 4);
            usleep(1);
}
    cleanup_platform();
    return 0;
}
```



1 AC 1 V /div 2.0divs 483.421 kHz

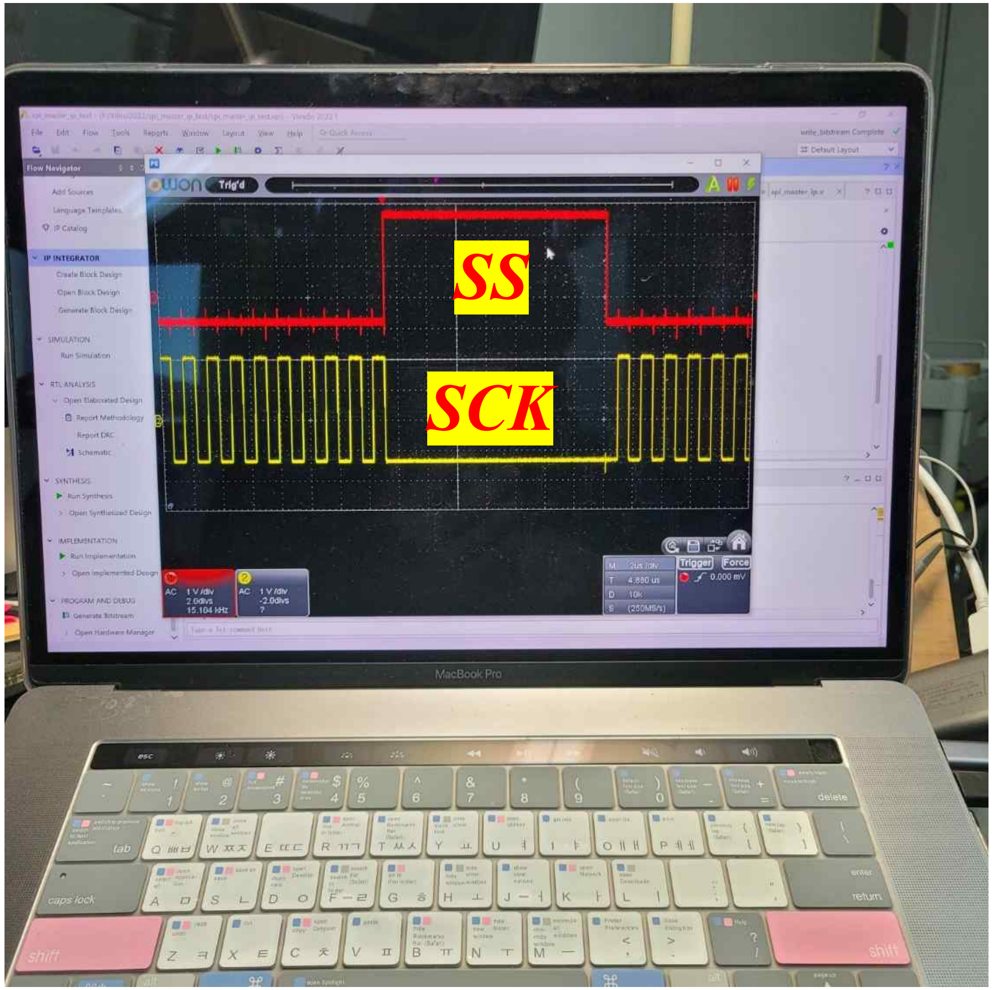
2 AC 1 V /div -2.0divs ?

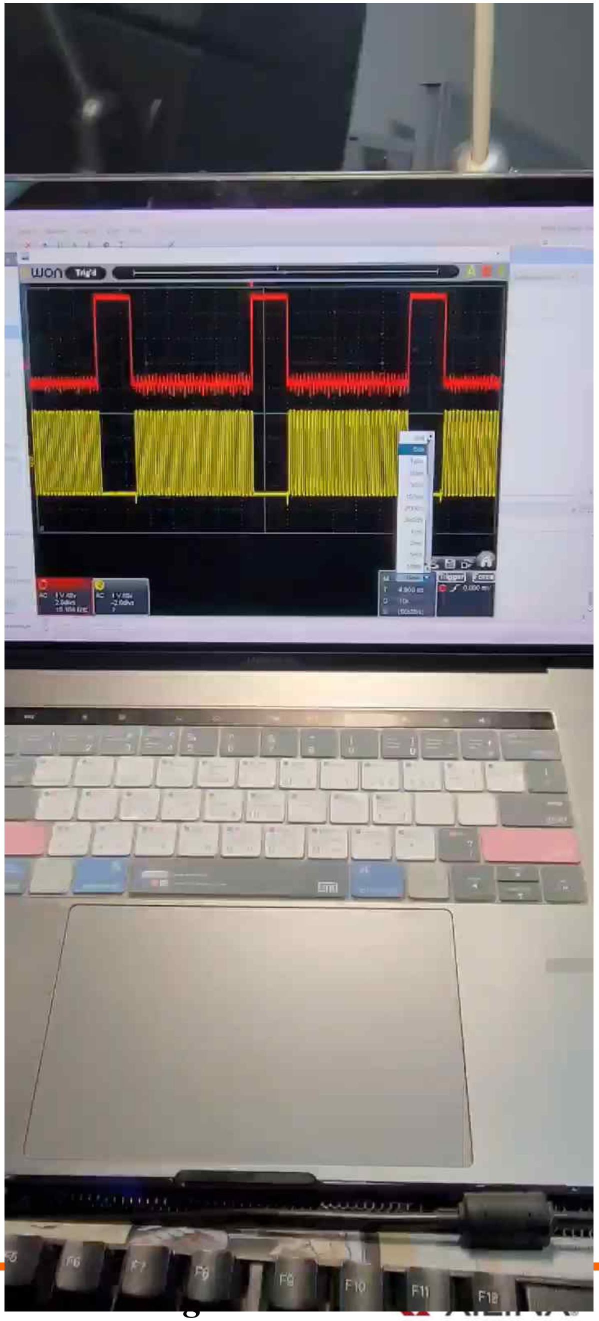
M 5us /div
T 0.0 ns
D 10k
S (100MS/s)

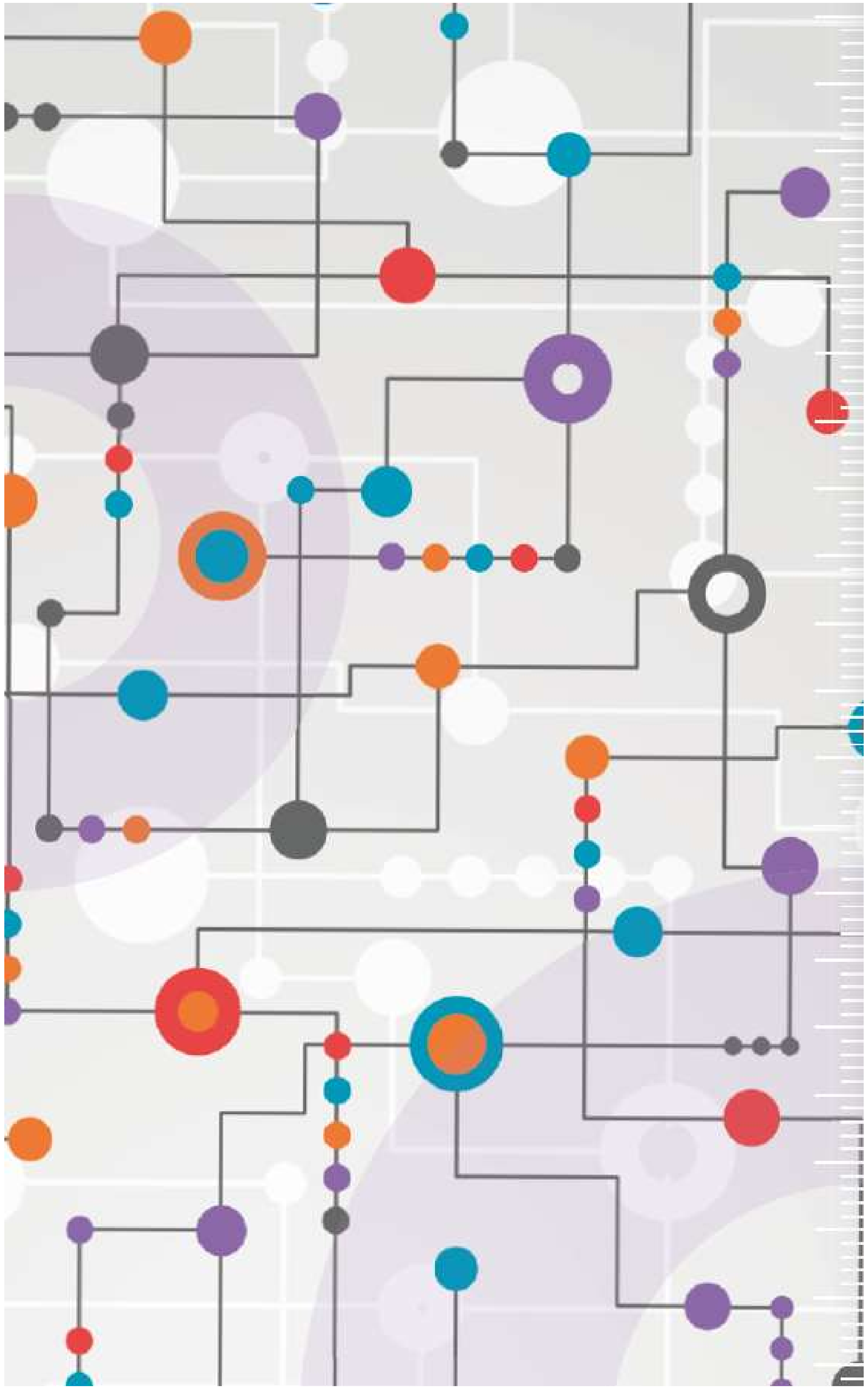
0.000 mV

User Logic Interface Implementation

spi_master_ip_test.xpr







수고하셨습니다.