MicroBlaze

ΘΈΜΑ ΠΤΥΧΙΑΚΉΣ ΕΡΓΑΣΊΑΣ :

Αξιολόγηση της ασφάλειας υλικού ενός MicroBlaze επεξεργαστή για επιθέσεις από σφάλματα.

ΑΡΙΘΜΌΣ ΜΗΤΡΏΟΥ :

2026202100134

ΟΝΟΜΑΤΕΠΏΝΥΜΟ :

ΑΝΑΣΤΆΣΙΟΣ ΠΟΥΤΑΧΊΔΗΣ

ΠΑΝΕΠΙΣΤΉΜΙΟ :

Πελοποννήσου.

ΤΜΉΜΑ :

Ψηφιακών Συστημάτων.

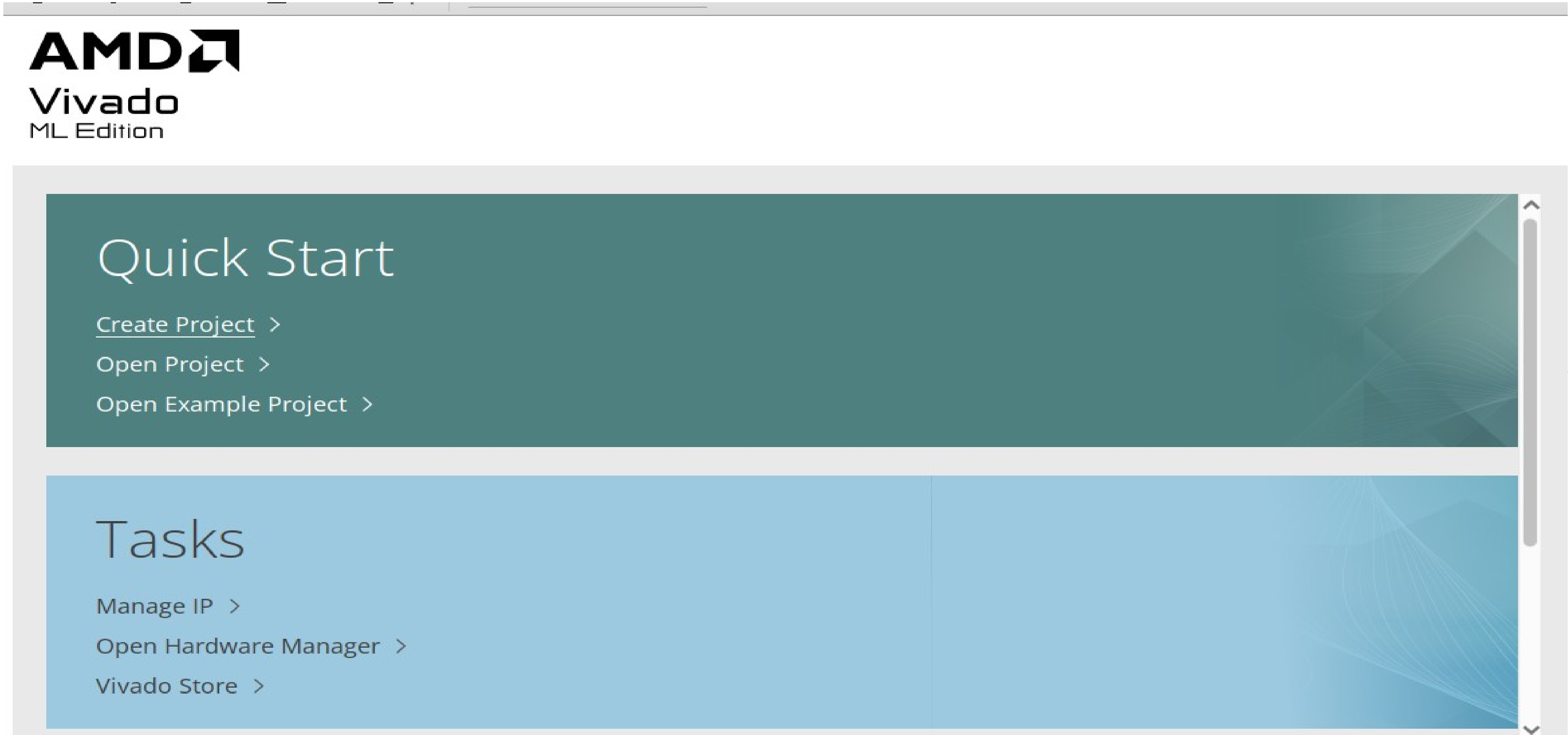
ΠΑΡΟΥΣΊΑΣΗ :

Πτυχιακής Εργασίας 20’.

ΕΡΓΑΛΕΊΑ ΥΛΟΠΟΊΗΣΗΣ :

Vitis(SW DEVELOPER)2023.2

Click : Create Project > .


AMD Vivado ML Edition

Quick Start

Create Project >

Open Project >

Open Example Project >

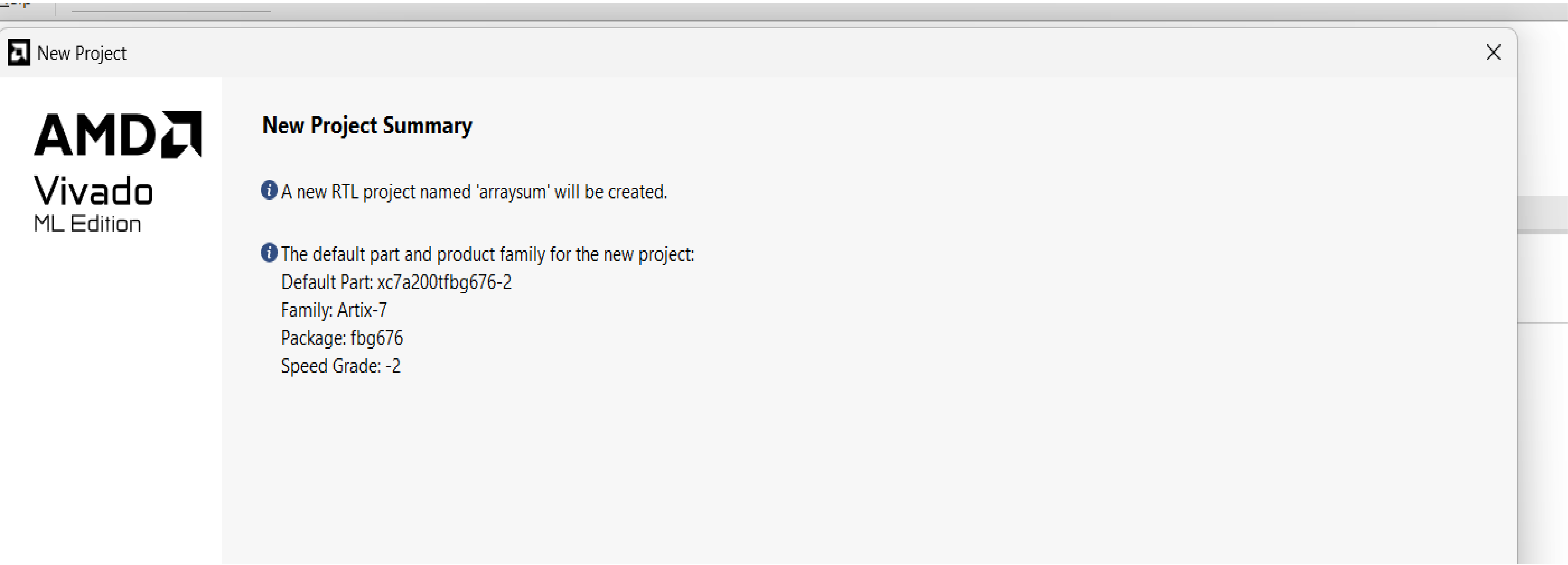
Tasks

Manage IP >

Open Hardware Manager >

Vivado Store >

Project Name : arraysum
Xilinx Board Part : xc7a200tfbg676-2

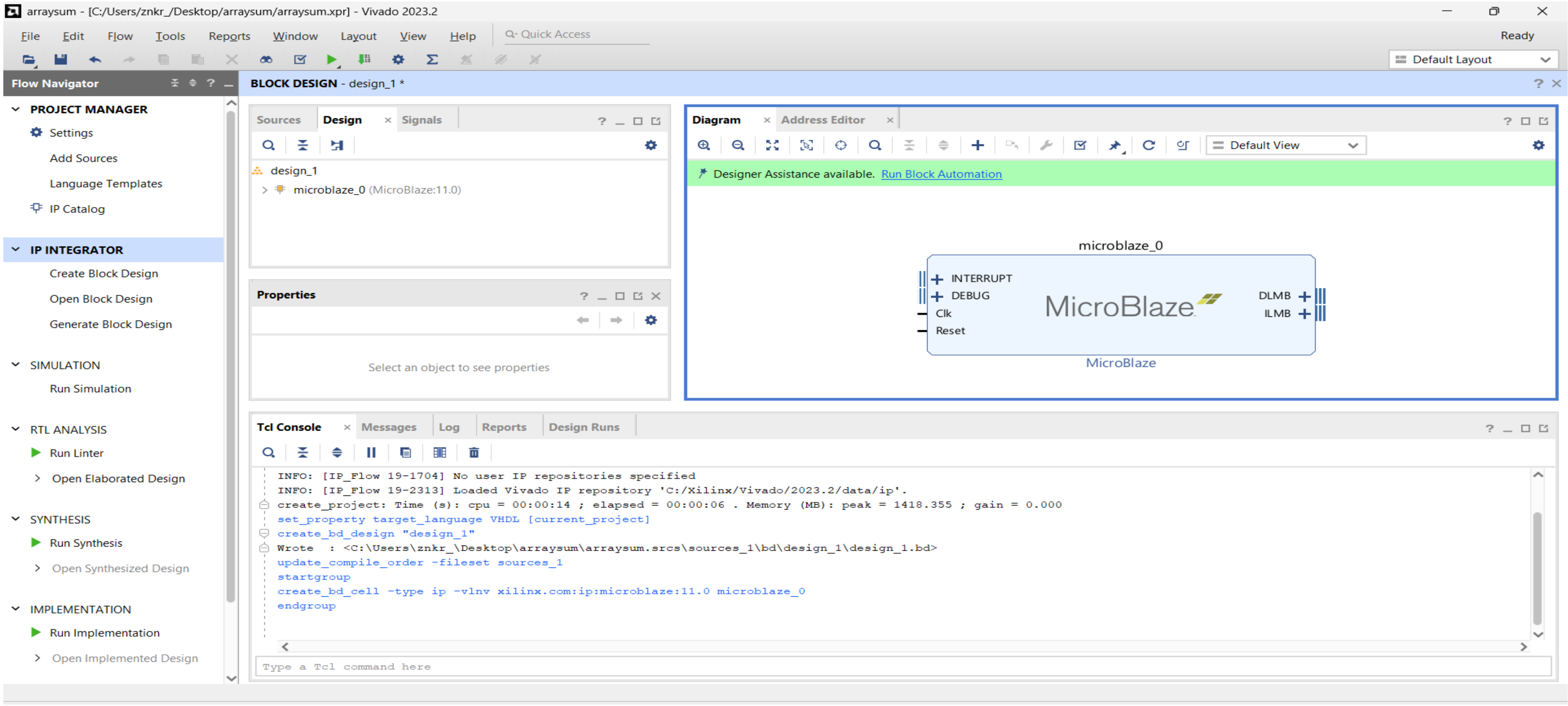


New Project Summary

A new RTL project named 'arraysum' will be created.

The default part and product family for the new project:
Default Part: xc7a200tfbg676-2
Family: Artix-7
Package: fbg676
Speed Grade: -2

Create Block Design : Add MicroBlaze and Run Block Automation.



design_1
microblaze_0 (MicroBlaze:11.0)

Select an object to see properties

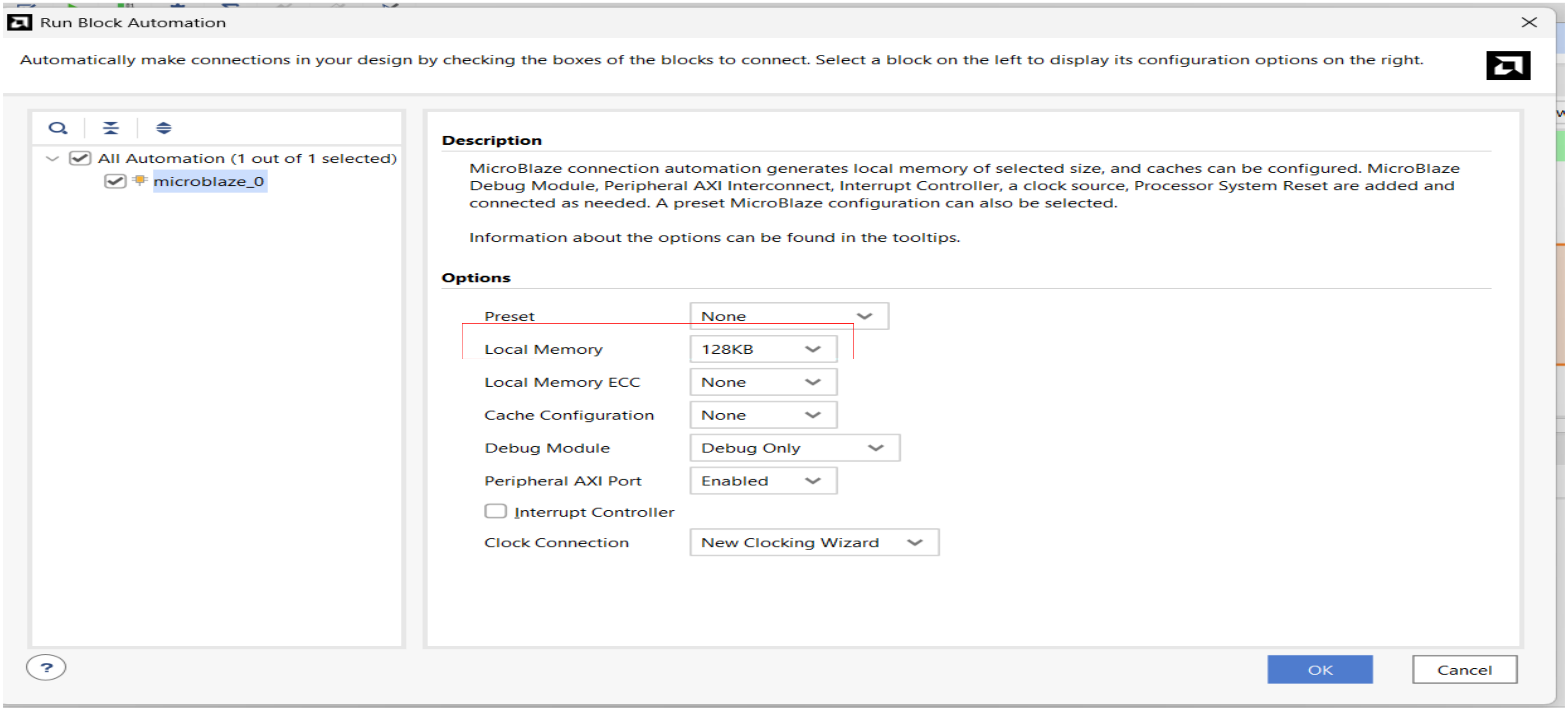
Designer Assistance available. Run Block Automation

```
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2023.2/data/ip'.
create_project: Time (s): cpu = 00:00:14 ; elapsed = 00:00:06 . Memory (MB): peak = 1418.355 ; gain = 0.000
set_property target_language VHDL [current_project]
create_bd_design "design_1"
Wrote  : <C:\Users\znkr_\Desktop\arraysum\arraysum.srcs\sources_1\bd\design_1\design_1.bd>
update_compile_order -fileset sources_1
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:microblaze:11.0 microblaze_0
endgroup
```

Type a Tcl command here

Set : Local Memory 128KB and Click OK.

(Improves Performance).



Run Block Automation

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

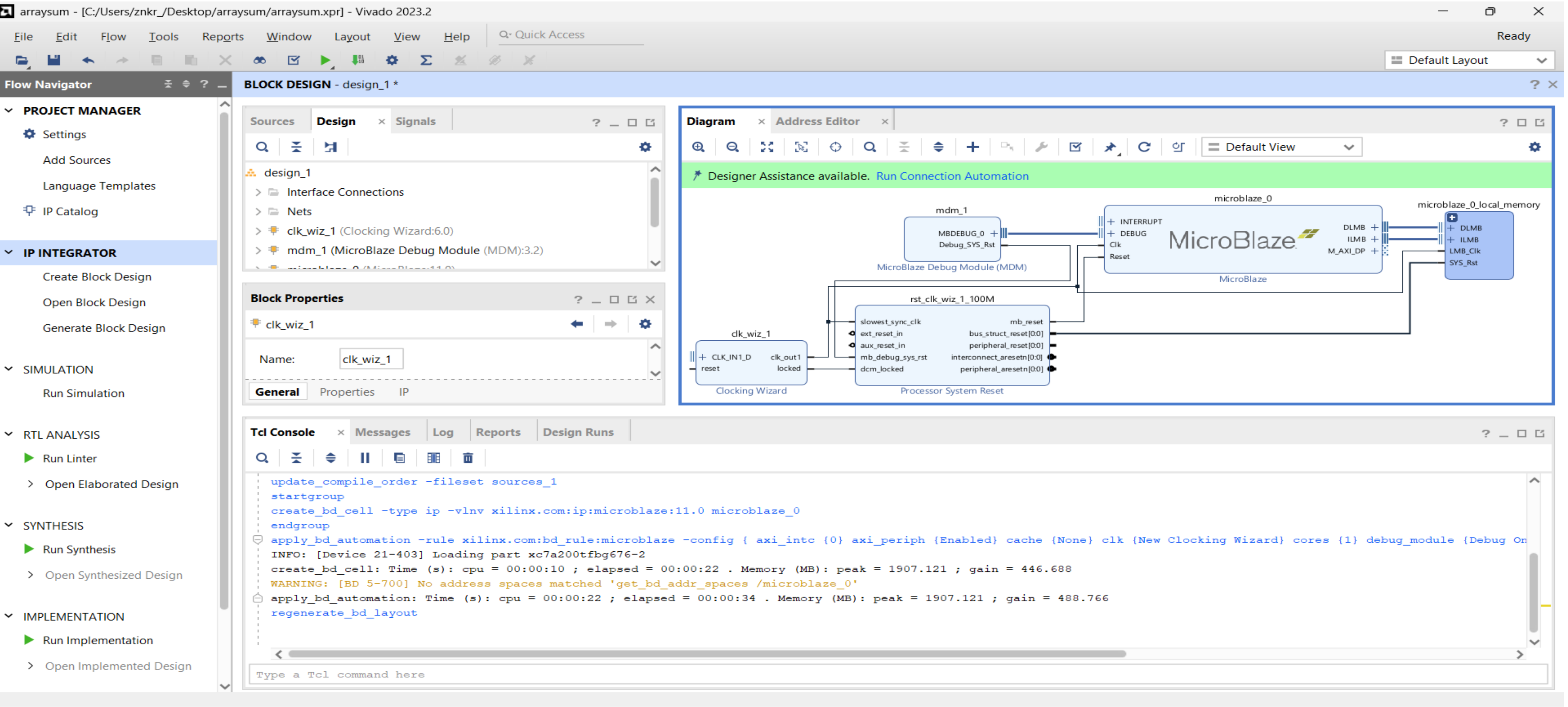
MicroBlaze connection automation generates local memory of selected size, and caches can be configured. MicroBlaze Debug Module, Peripheral AXI Interconnect, Interrupt Controller, a clock source, Processor System Reset are added and connected as needed. A preset MicroBlaze configuration can also be selected.

Information about the options can be found in the tooltips.

Options

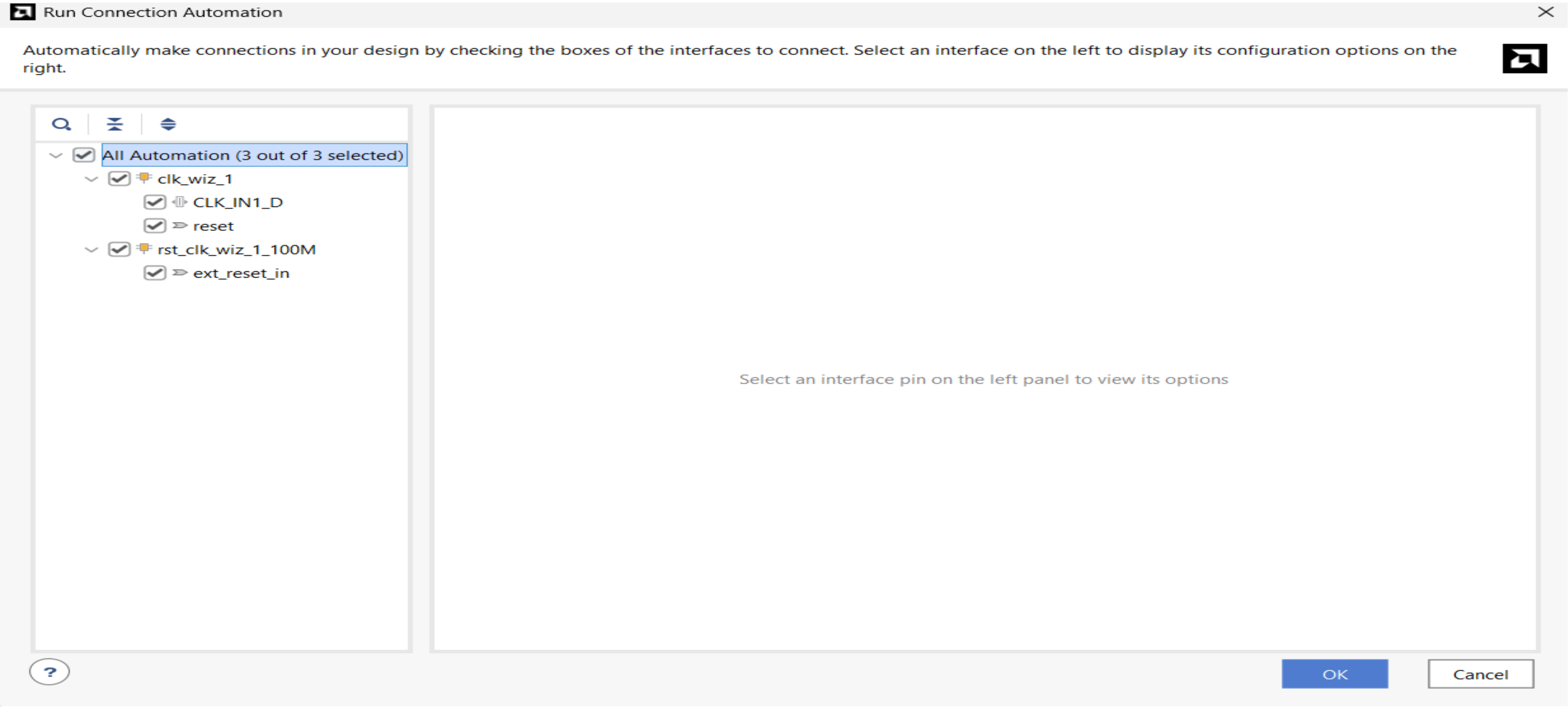
Preset	None
Local Memory	128KB
Local Memory ECC	None
Cache Configuration	None
Debug Module	Debug Only
Peripheral AXI Port	Enabled
Interrupt Controller	☐
Clock Connection	New Clocking Wizard

Click : Run Connection Automation.



```
update_compile_order -fileset sources_1
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:microblaze:11.0 microblaze_0
endgroup
apply_bd_automation -rule xilinx.com:bd_rule:microblaze -config { axi_intc {0} axi_periph {Enabled} cache {None} clk {New Clocking Wizard} cores {1} debug_module {Debug On
INFO: [Device 21-403] Loading part xc7a200tfbg676-2
create_bd_cell: Time (s): cpu = 00:00:10 ; elapsed = 00:00:22 . Memory (MB): peak = 1907.121 ; gain = 446.688
WARNING: [BD 5-700] No address spaces matched 'get_bd_addr_spaces /microblaze_0'
apply_bd_automation: Time (s): cpu = 00:00:22 ; elapsed = 00:00:34 . Memory (MB): peak = 1907.121 ; gain = 488.766
regenerate_bd_layout
```

Select : All(3 out of 3 selected) *and* ***Click OK.***


Run Connection Automation

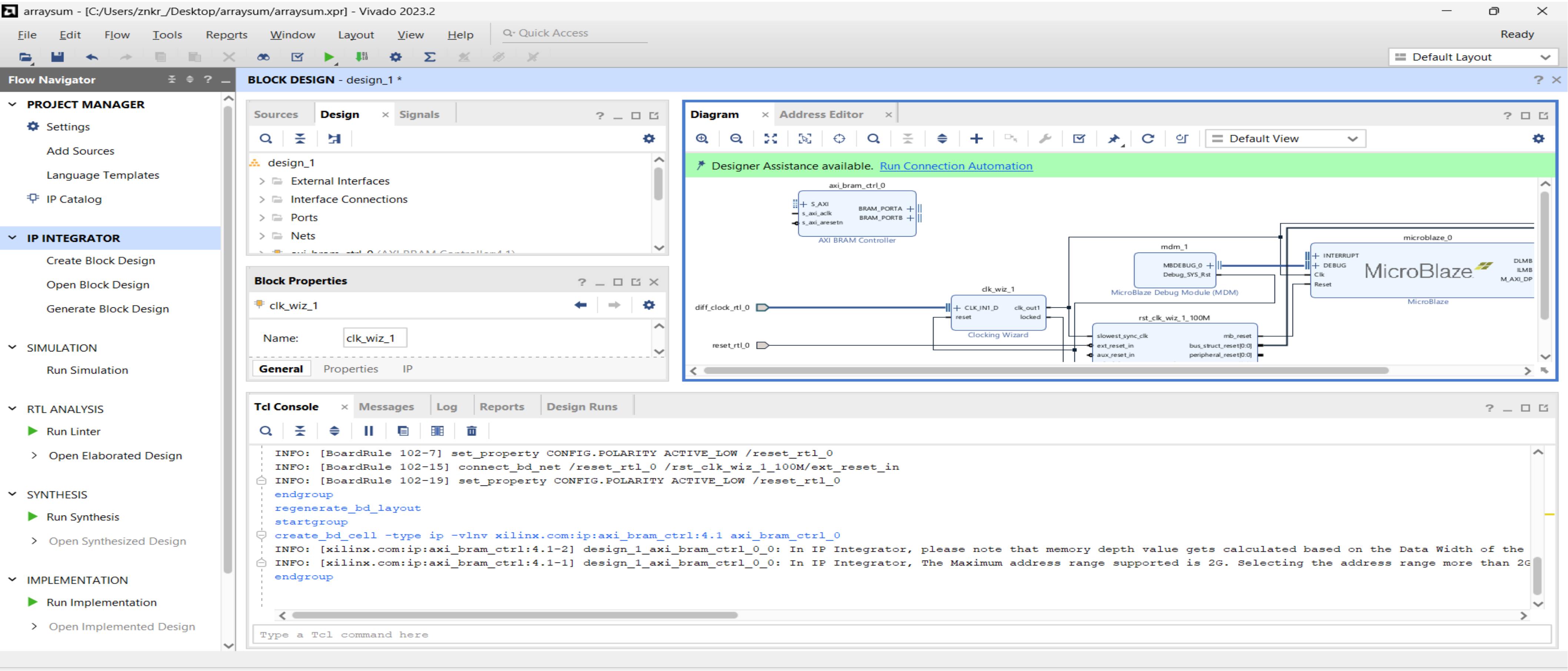
Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.

- ✓ All Automation (3 out of 3 selected)
  - ✓ clk_wiz_1
    - ✓ CLK_IN1_D
    - ✓ reset
  - ✓ rst_clk_wiz_1_100M
    - ✓ ext_reset_in

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

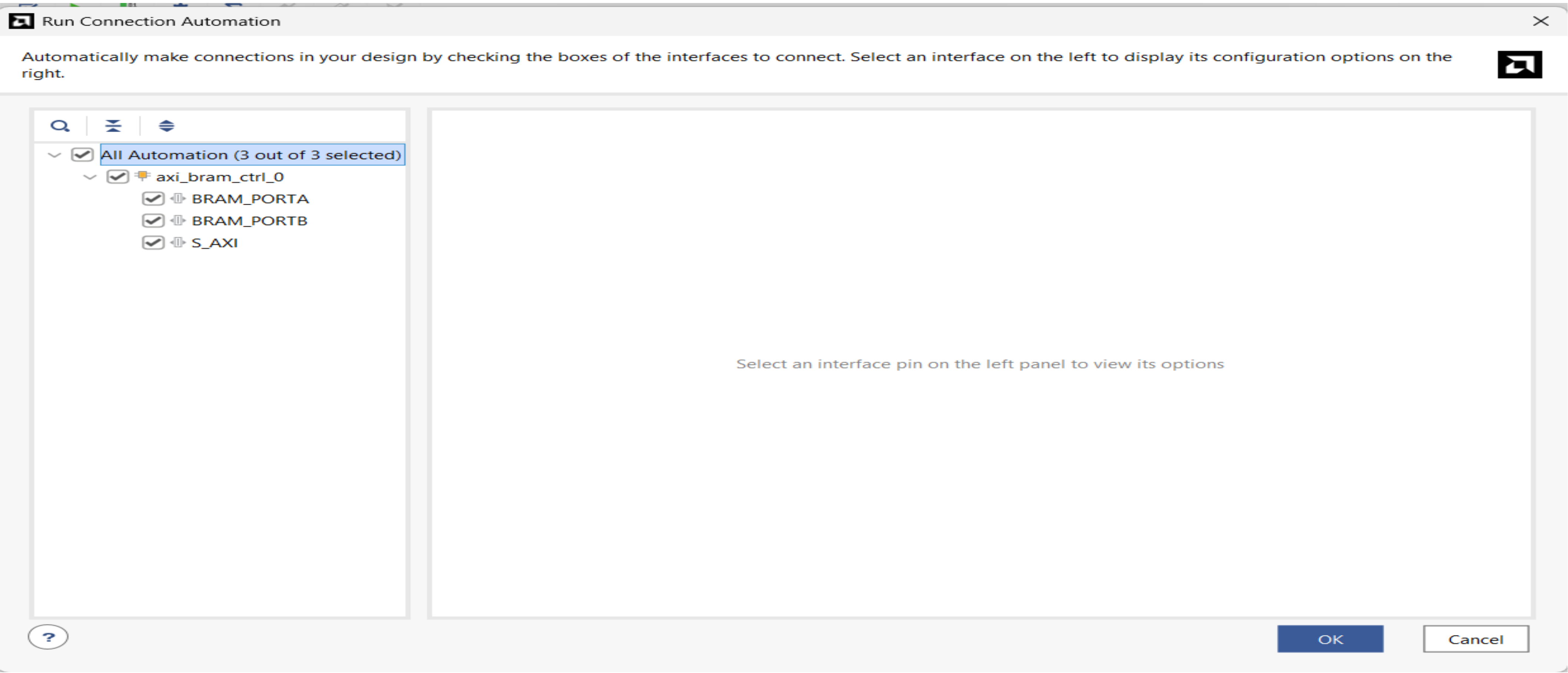
Add BRAM *and* Run Connection Automation.

(BRAM : boosts system performance by minimizing the delay in data access).



```
INFO: [BoardRule 102-7] set_property CONFIG.POLARITY ACTIVE_LOW /reset_rtl_0
INFO: [BoardRule 102-15] connect_bd_net /reset_rtl_0 /rst_clk_wiz_1_100M/ext_reset_in
INFO: [BoardRule 102-19] set_property CONFIG.POLARITY ACTIVE_LOW /reset_rtl_0
endgroup
regenerate_bd_layout
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_bram_ctrl:4.1 axi_bram_ctrl_0
INFO: [xilinx.com:ip:axi_bram_ctrl:4.1-2] design_1_axi_bram_ctrl_0_0: In IP Integrator, please note that memory depth value gets calculated based on the Data Width of the
INFO: [xilinx.com:ip:axi_bram_ctrl:4.1-1] design_1_axi_bram_ctrl_0_0: In IP Integrator, The Maximum address range supported is 2G. Selecting the address range more than 2G
endgroup
```

Select : All(3 out of 3 selected) and Click OK.



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 (3 out of 3 selected)
  - [✓] axi_bram_ctrl_0
    - [✓] BRAM_PORTA
    - [✓] BRAM_PORTB
    - [✓] S_AXI

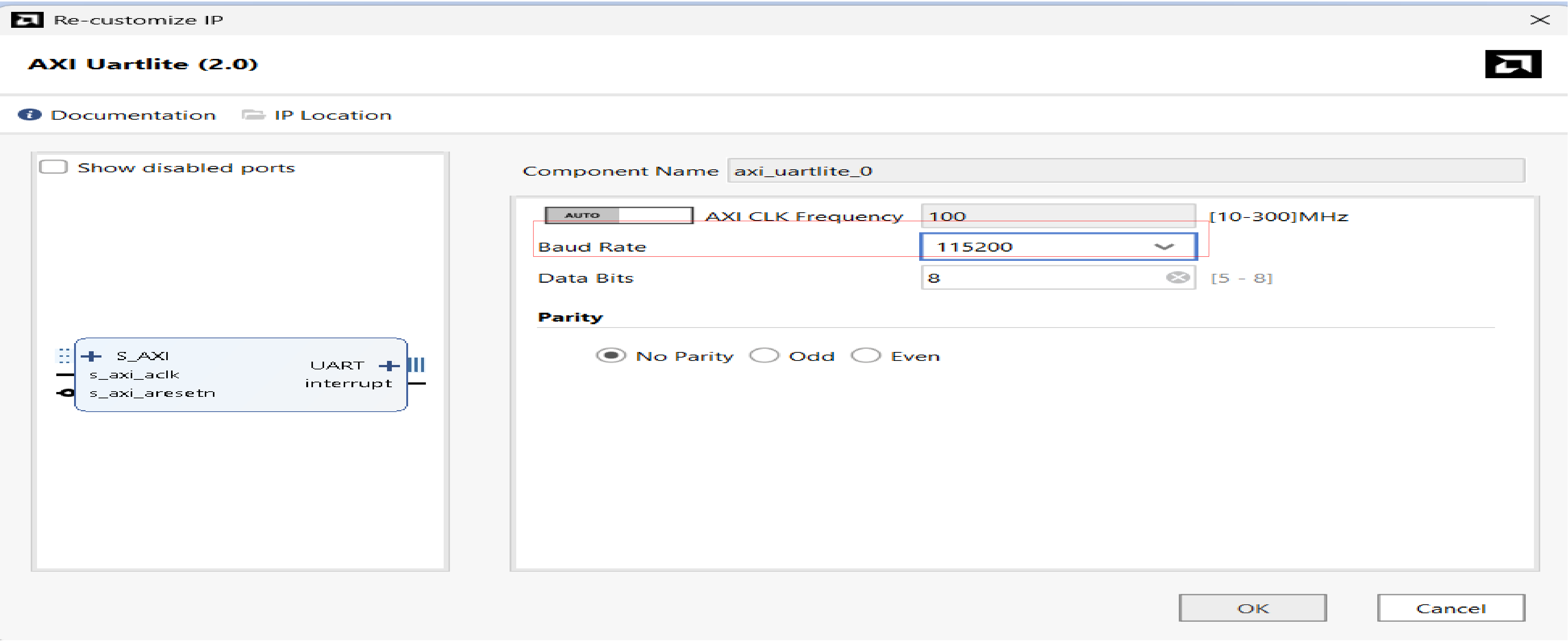
Select an interface pin on the left panel to view its options

Add UART and Set : Baud Rate 115200.

***Baud Rate:** Symbols/Second*

(My testbench is written for this baud rate).

(UART : MicroBlaze can send and receive serial data with external devices).



Re-customize IP

AXI Uartlite (2.0)

Documentation IP Location

Show disabled ports

Component Name: axi_uartlite_0

AUTO AXI CLK Frequency: 100 [10-300]MHz

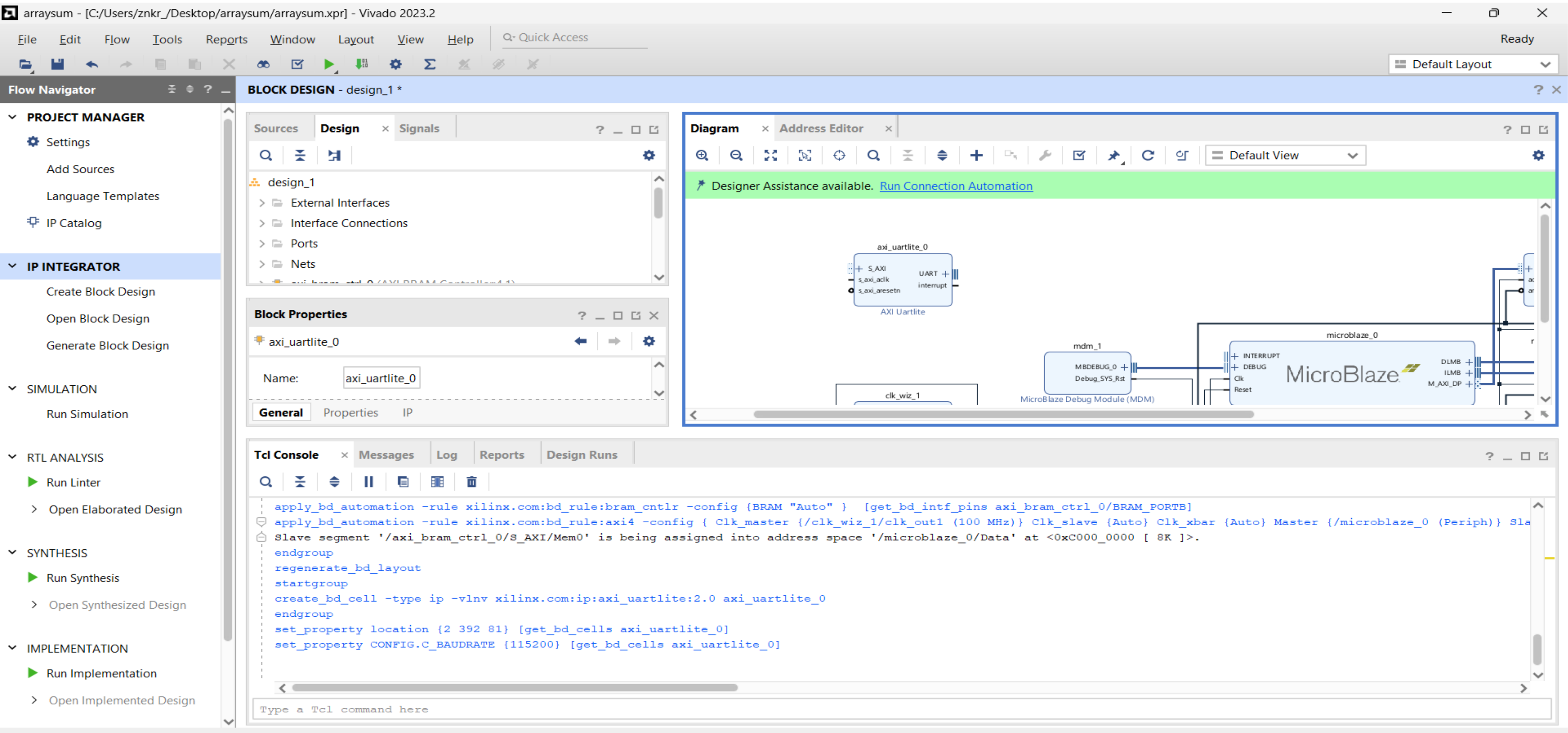
Baud Rate: 115200

Data Bits: 8 [5 - 8]

Parity

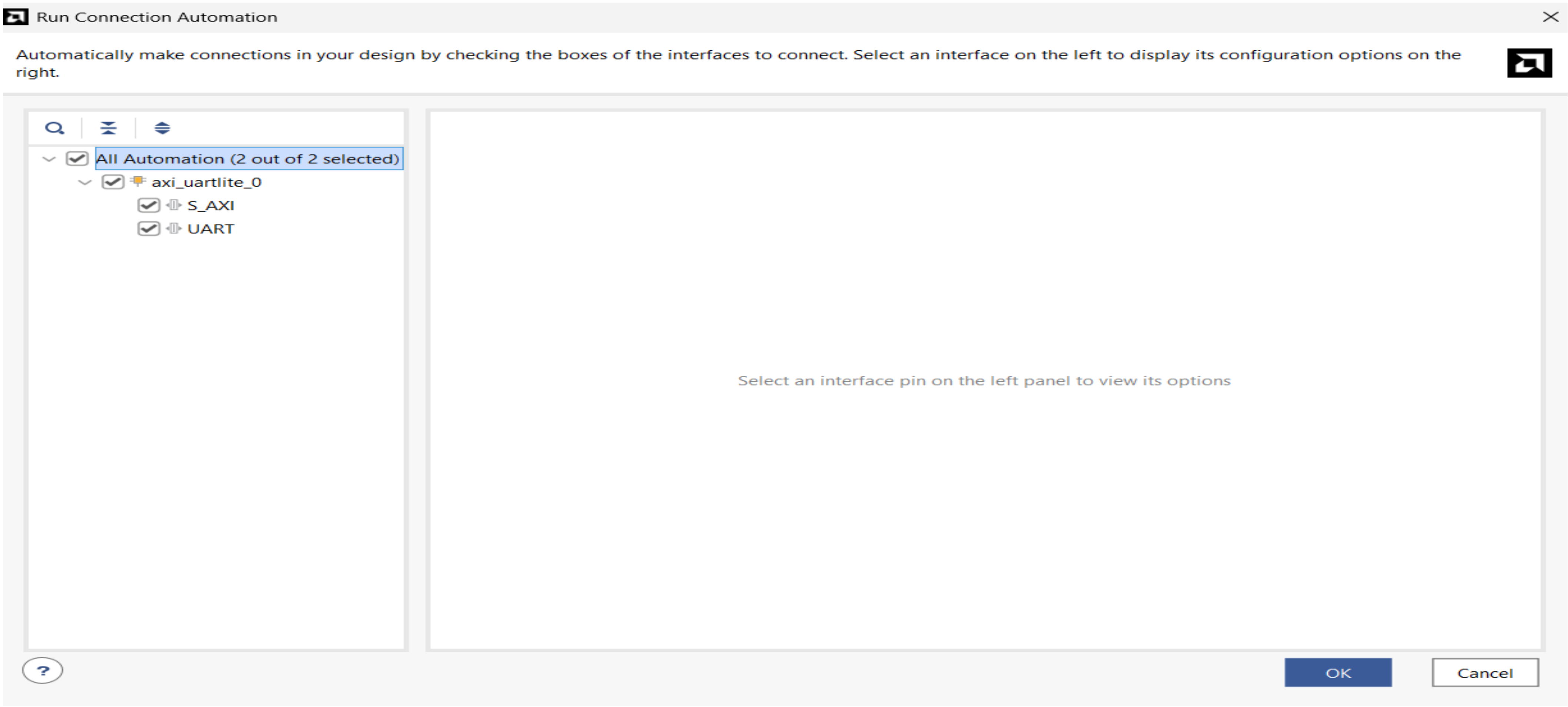
[● No Parity] ○ Odd ○ Even

Click : Run Connection Automation.



```
apply_bd_automation -rule xilinx.com:bd_rule:bram_cntlr -config {BRAM "Auto" }  [get_bd_intf_pins axi_bram_ctrl_0/BRAM_PORTB]
apply_bd_automation -rule xilinx.com:bd_rule:axi4 -config { Clk_master {/clk_wiz_1/clk_out1 (100 MHz)} Clk_slave {Auto} Clk_xbar {Auto} Master {/microblaze_0 (Periph)} Sla
Slave segment '/axi_bram_ctrl_0/S_AXI/Mem0' is being assigned into address space '/microblaze_0/Data' at <0xC000_0000 [ 8K ]>.
endgroup
regenerate_bd_layout
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:axi_uartlite:2.0 axi_uartlite_0
endgroup
set_property location {2 392 81} [get_bd_cells axi_uartlite_0]
set_property CONFIG.C_BAUDRATE {115200} [get_bd_cells axi_uartlite_0]
```

Select : All(3 out of 3 selected) *and* ***Click OK.***



Run Connection Automation

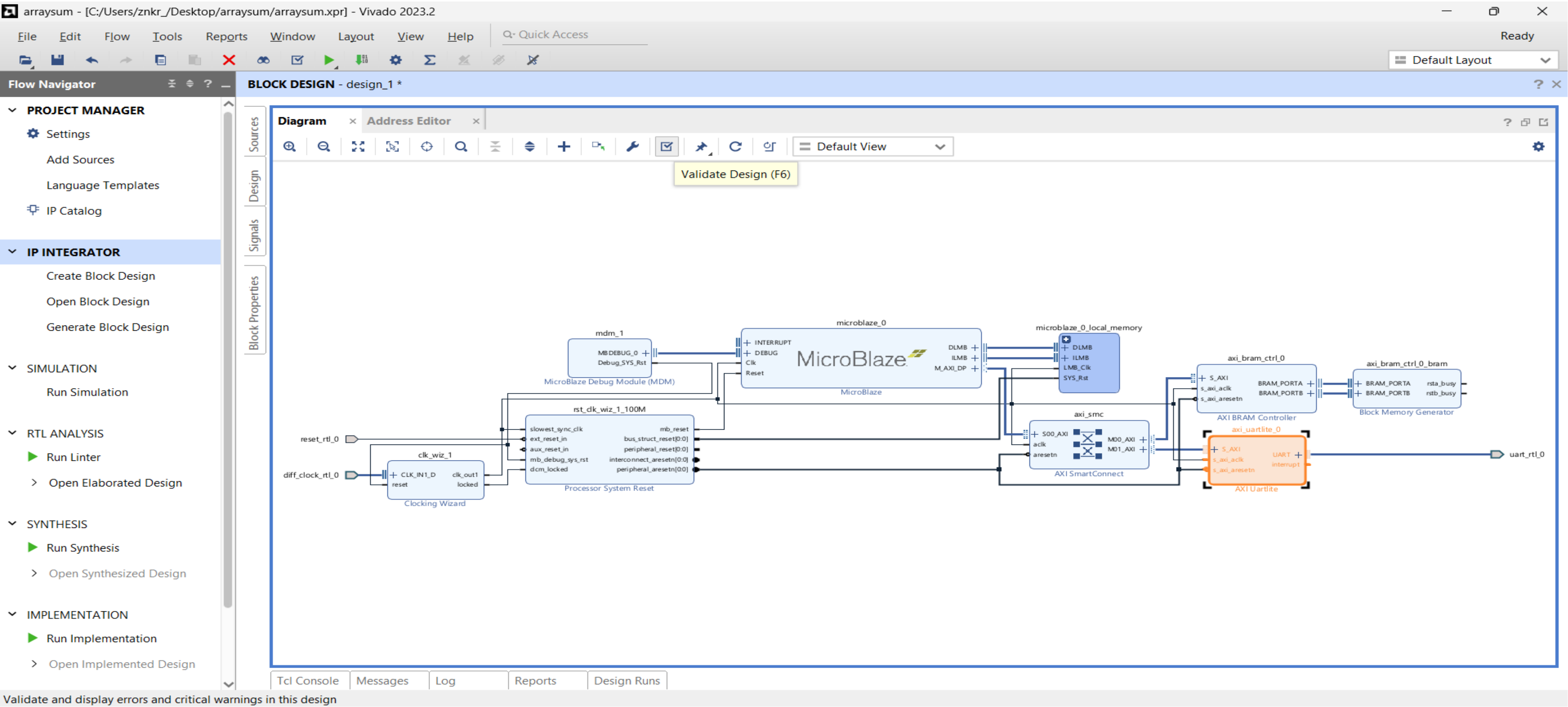
Automatically make connections in your design by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on the right.

- [x] All Automation (2 out of 2 selected)
  - [x] axi_uartlite_0
    - [x] S_AXI
    - [x] UART

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

Click: Validate Design (F6).

(Given : The Form Of The Block Design).

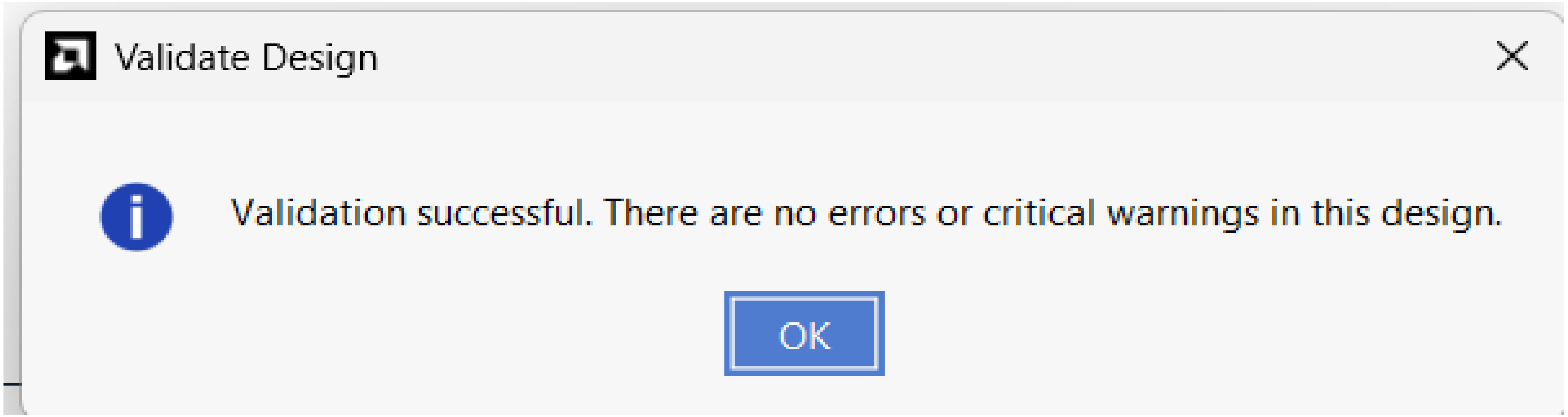

arraysum - [C:/Users/znkr_/Desktop/arraysum/arraysum.xpr] - Vivado 2023.2

BLOCK DESIGN - design_1 *

Validate and display errors and critical warnings in this design

Click : OK. *<u>If</u> <u>The</u>* ***Validation*** *<u>Is</u>* ***Successful.***

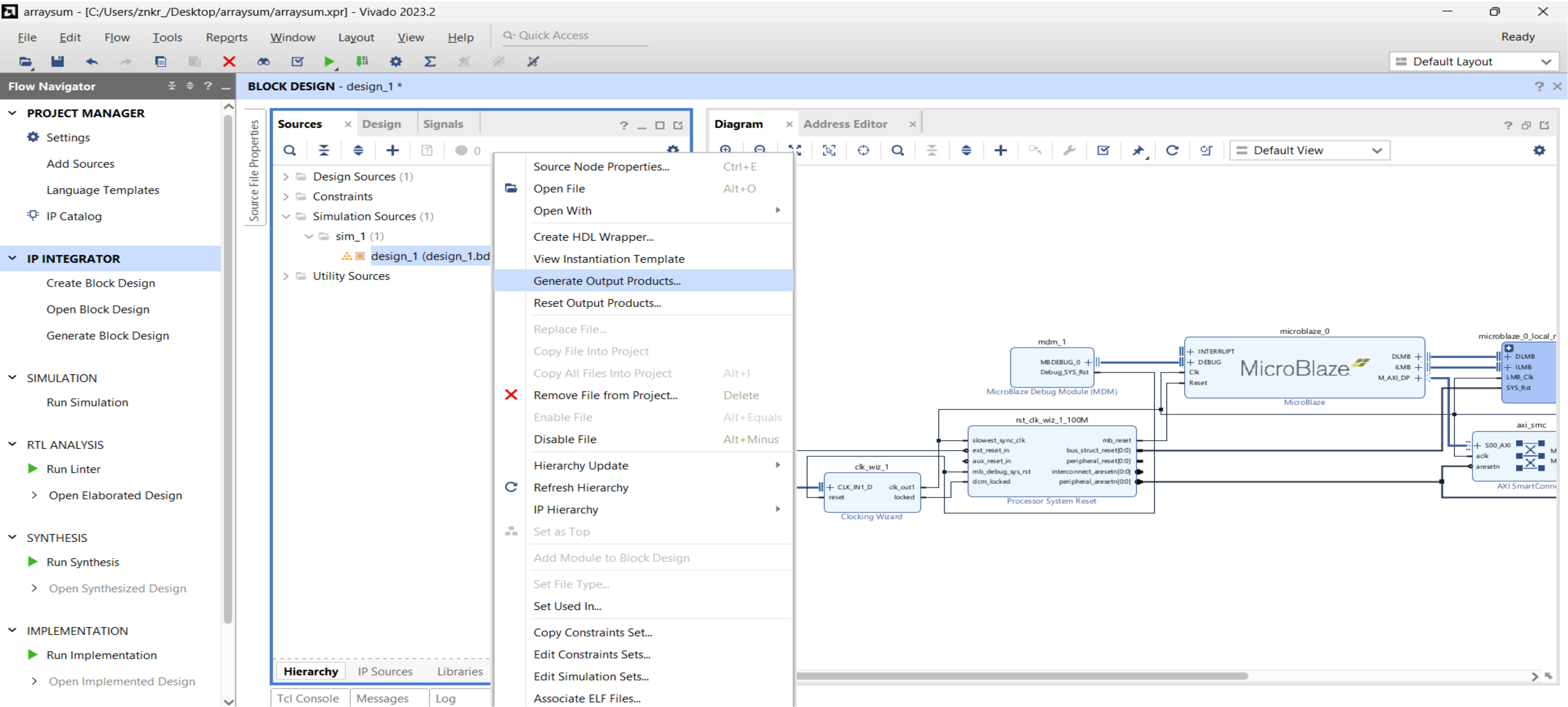
<u>(This window will open).</u>


Validate Design

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

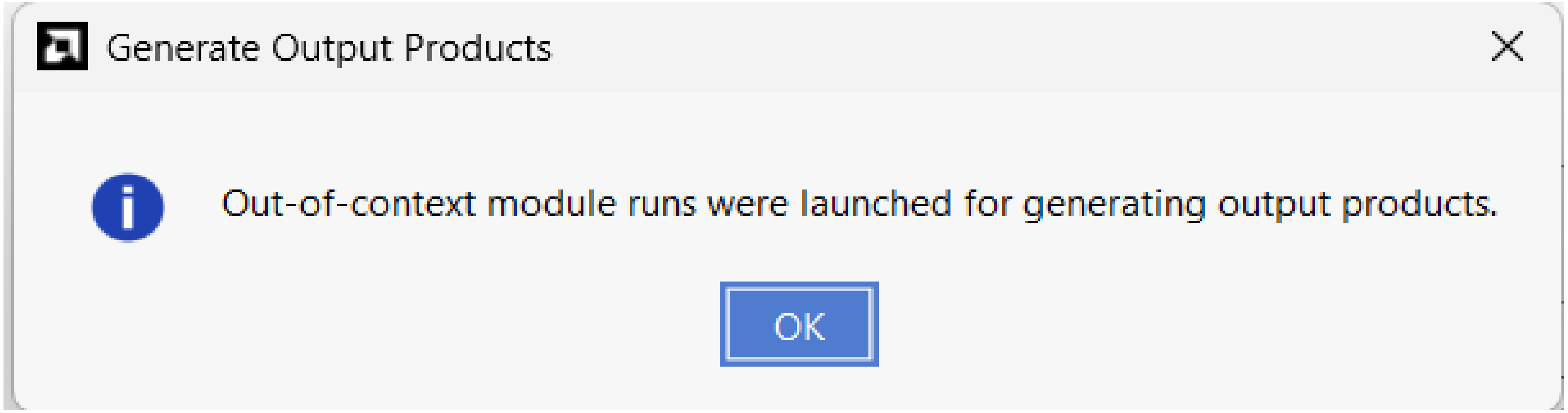
Click : Generate Output Products…

(For synthesizing the design and implementing it to test the system).



Click : OK. *If the* ***synthesis*** *and* ***implementation*** *are* ***successful.***

(This window will open).

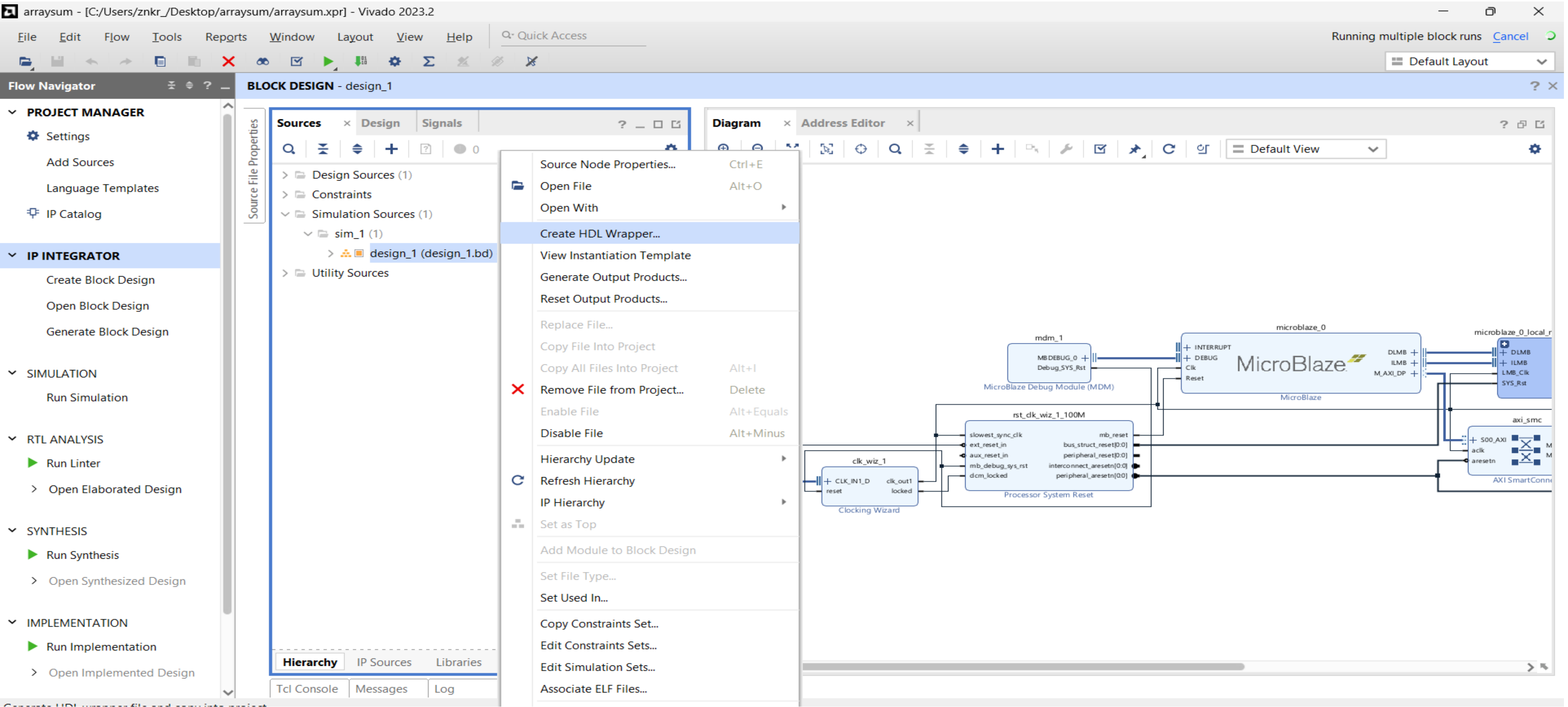


Generate Output Products

Out-of-context module runs were launched for generating output products.

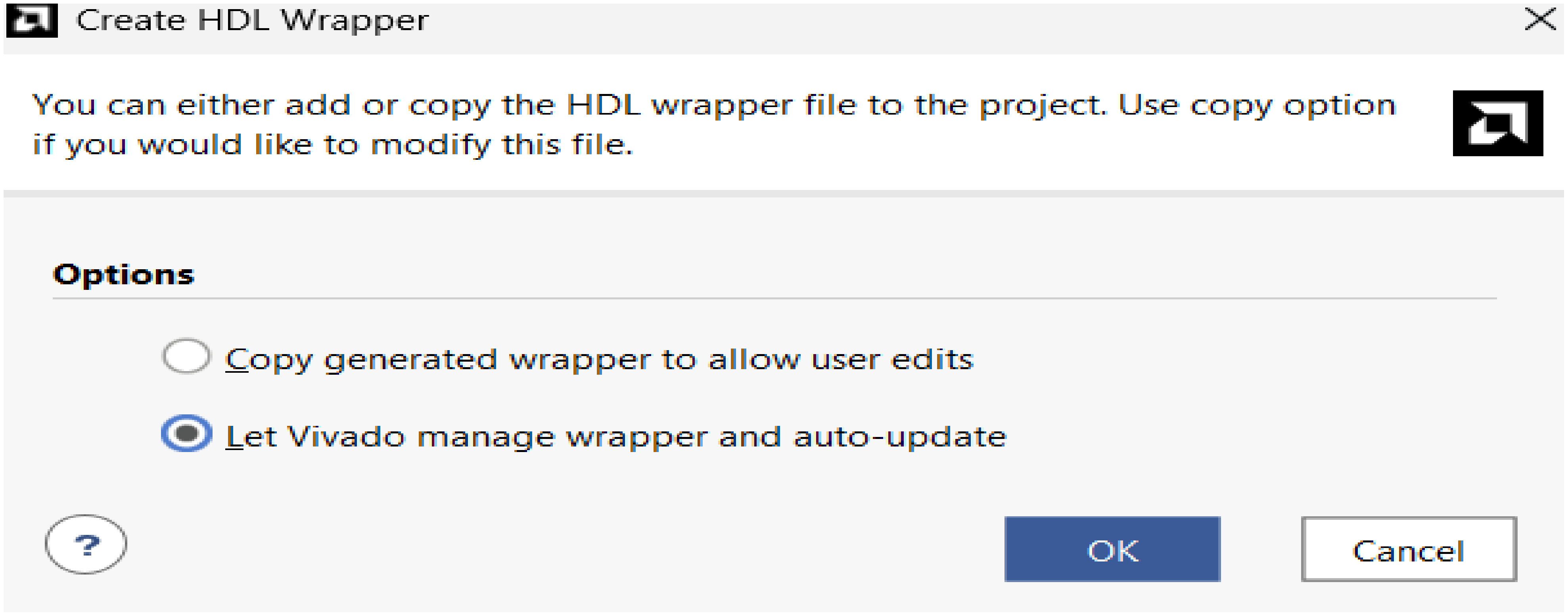
Click : Create HDL Wrapper…

(Manages the design within the FPGA).



Click : OK.

(Let Vivado manage wrapper and auto-update).



Create HDL Wrapper

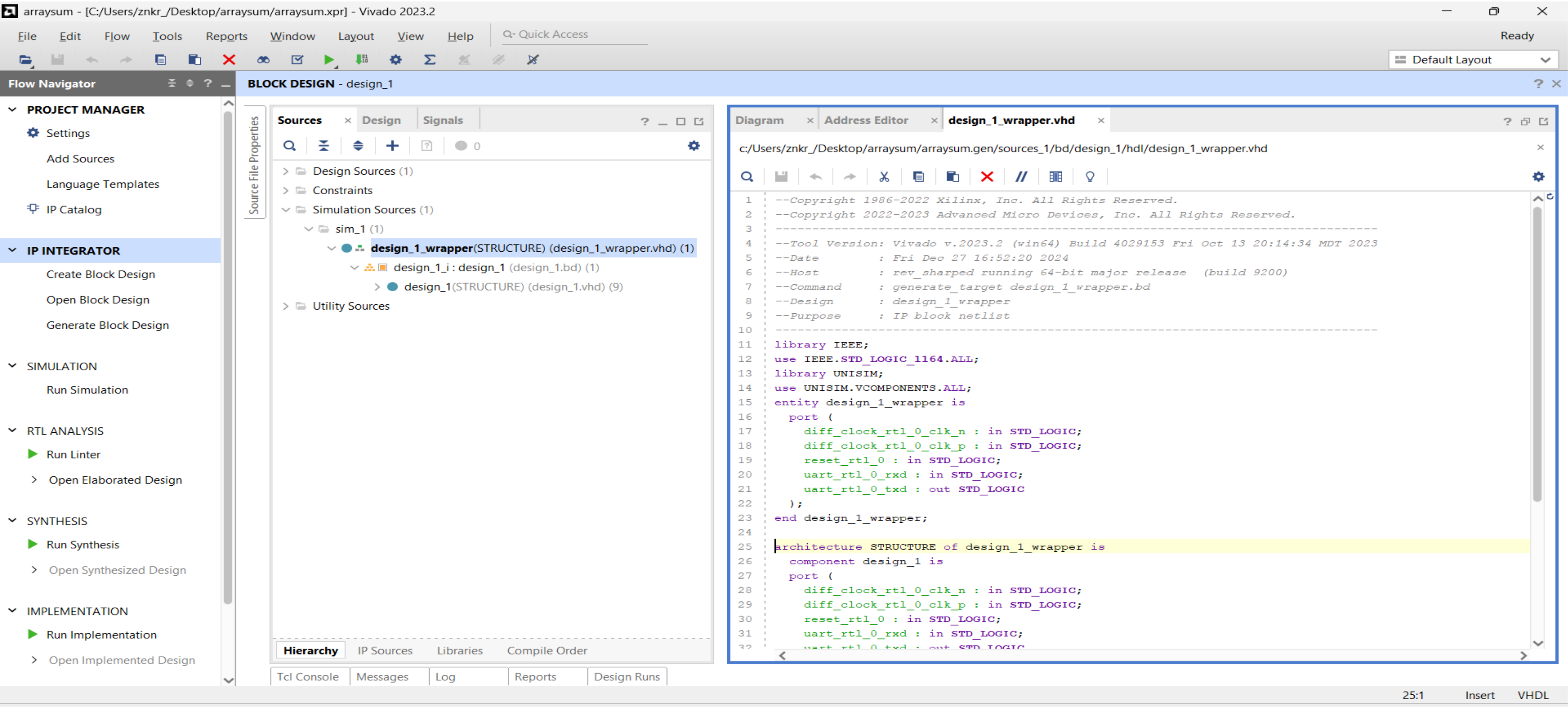
You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.

Options

- ◯ Copy generated wrapper to allow user edits
- ● Let Vivado manage wrapper and auto-update

HDL Wrapper Generated.

(With name : design_1_wrapper.vhd).



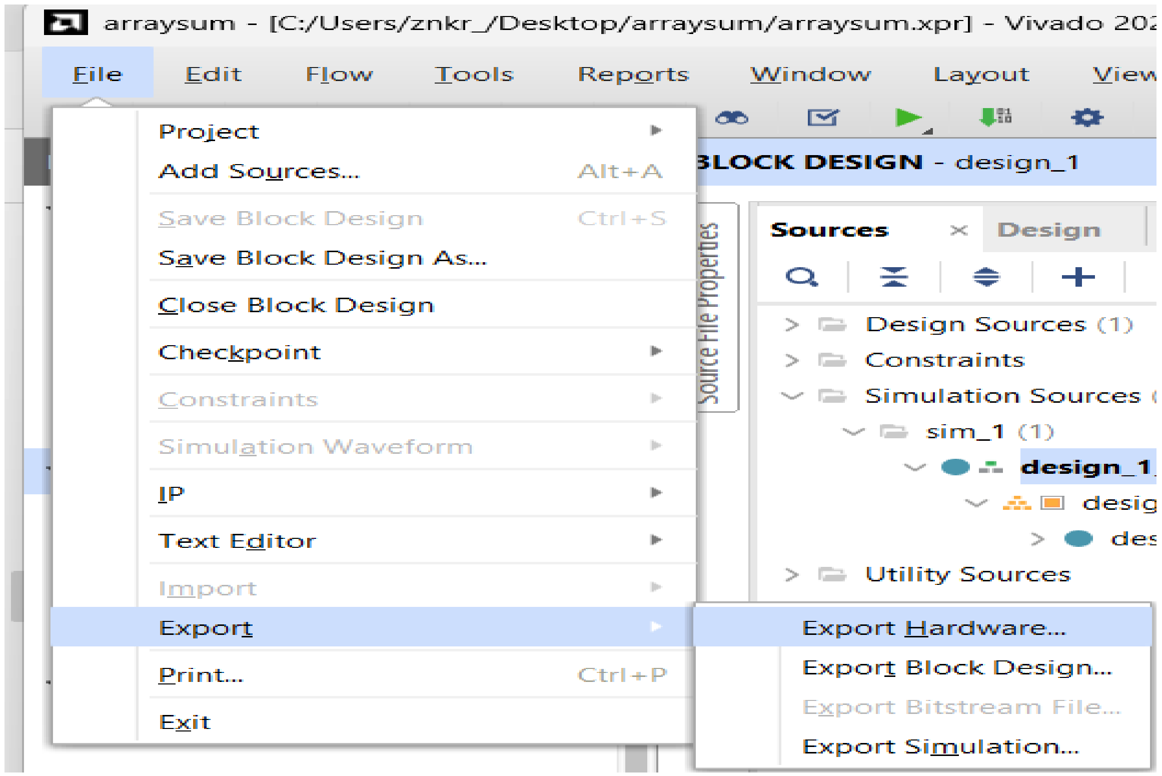
Design Sources (1)
Constraints
Simulation Sources (1)
sim_1 (1)
design_1_wrapper(STRUCTURE) (design_1_wrapper.vhd) (1)
design_1_i : design_1 (design_1.bd) (1)
design_1(STRUCTURE) (design_1.vhd) (9)
Utility Sources

c:/Users/znkr_/Desktop/arraysum/arraysum.gen/sources_1/bd/design_1/hdl/design_1_wrapper.vhd

```
--Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.
--Copyright 2022-2023 Advanced Micro Devices, Inc. All Rights Reserved.
----------------------------------------------------------------------------------
--Tool Version: Vivado v.2023.2 (win64) Build 4029153 Fri Oct 13 20:14:34 MDT 2023
--Date        : Fri Dec 27 16:52:20 2024
--Host        : rev_sharped running 64-bit major release  (build 9200)
--Command     : generate_target design_1_wrapper.bd
--Design      : design_1_wrapper
--Purpose     : IP block netlist
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity design_1_wrapper is
  port (
    diff_clock_rtl_0_clk_n : in STD_LOGIC;
    diff_clock_rtl_0_clk_p : in STD_LOGIC;
    reset_rtl_0 : in STD_LOGIC;
    uart_rtl_0_rxd : in STD_LOGIC;
    uart_rtl_0_txd : out STD_LOGIC
  );
end design_1_wrapper;

architecture STRUCTURE of design_1_wrapper is
  component design_1 is
  port (
    diff_clock_rtl_0_clk_n : in STD_LOGIC;
    diff_clock_rtl_0_clk_p : in STD_LOGIC;
    reset_rtl_0 : in STD_LOGIC;
    uart_rtl_0_rxd : in STD_LOGIC;
    uart_rtl_0_txd : out STD_LOGIC
```

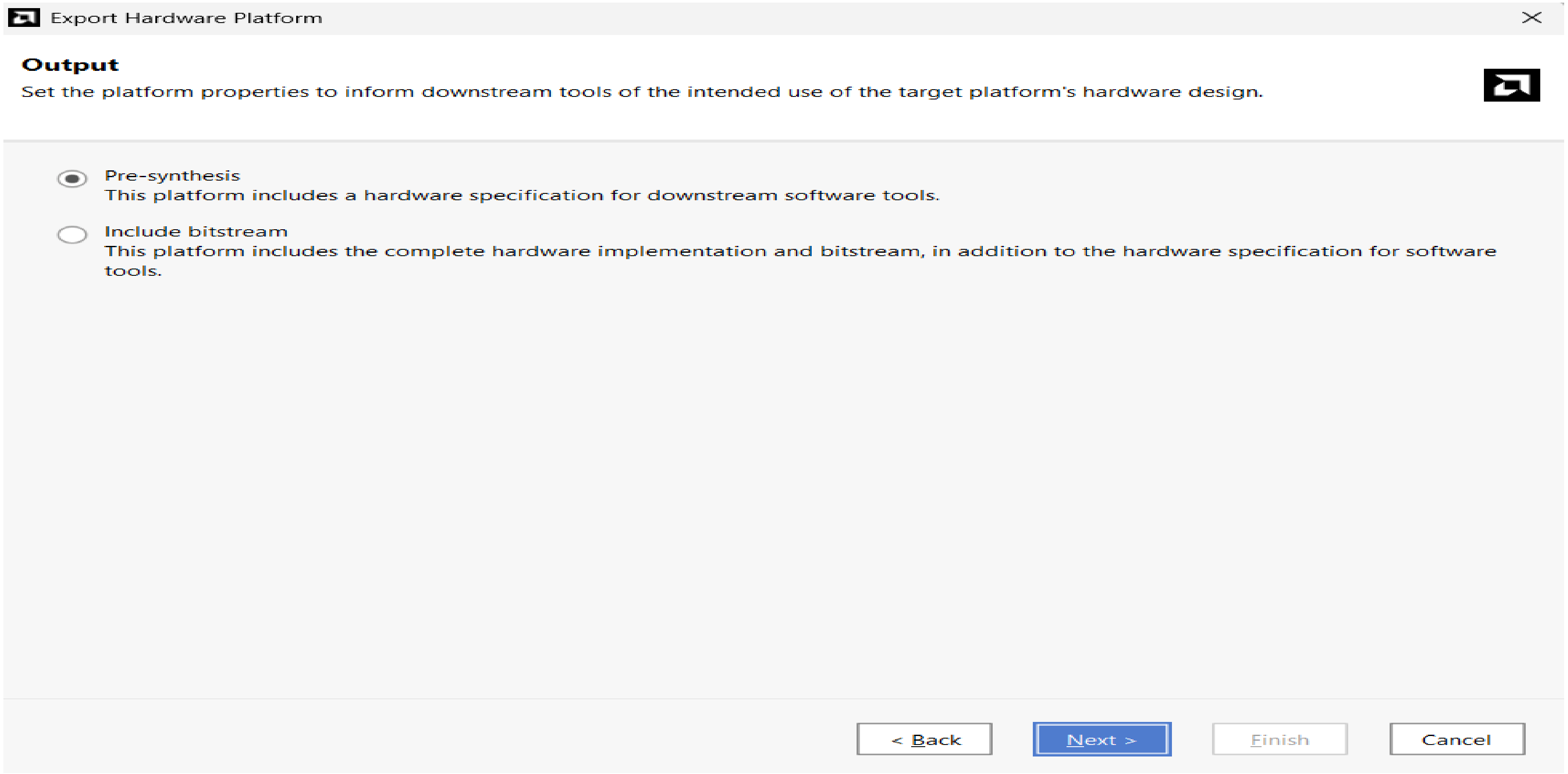
Click : Export Hardware…

(implementing the design on an FPGA).



Click : Next > .

<u>(pre-synthesis ensures that the hardware is ready for software development).</u>


Export Hardware Platform

Output

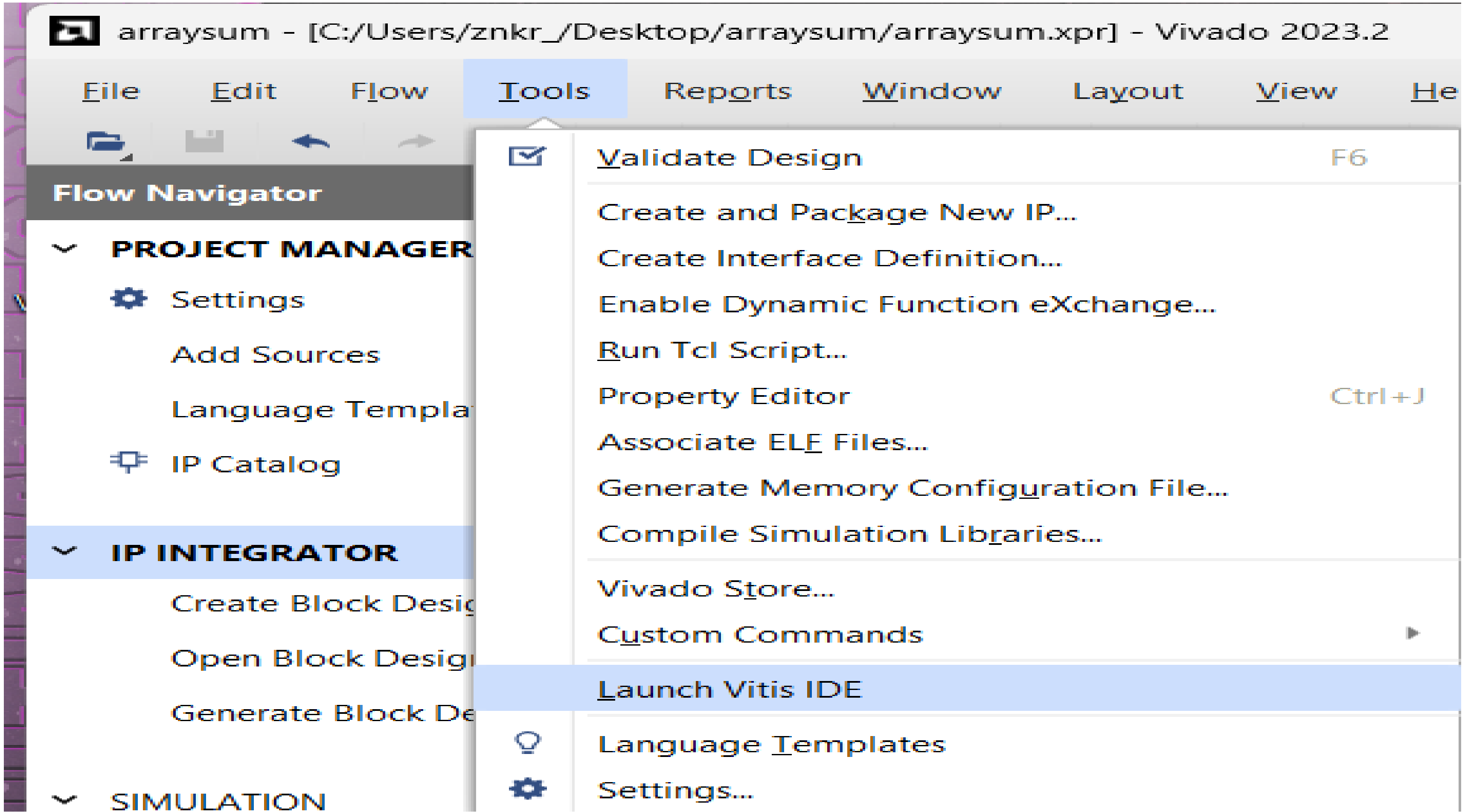
Set the platform properties to inform downstream tools of the intended use of the target platform's hardware design.

● Pre-synthesis
This platform includes a hardware specification for downstream software tools.

○ Include bitstream
This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for software tools.

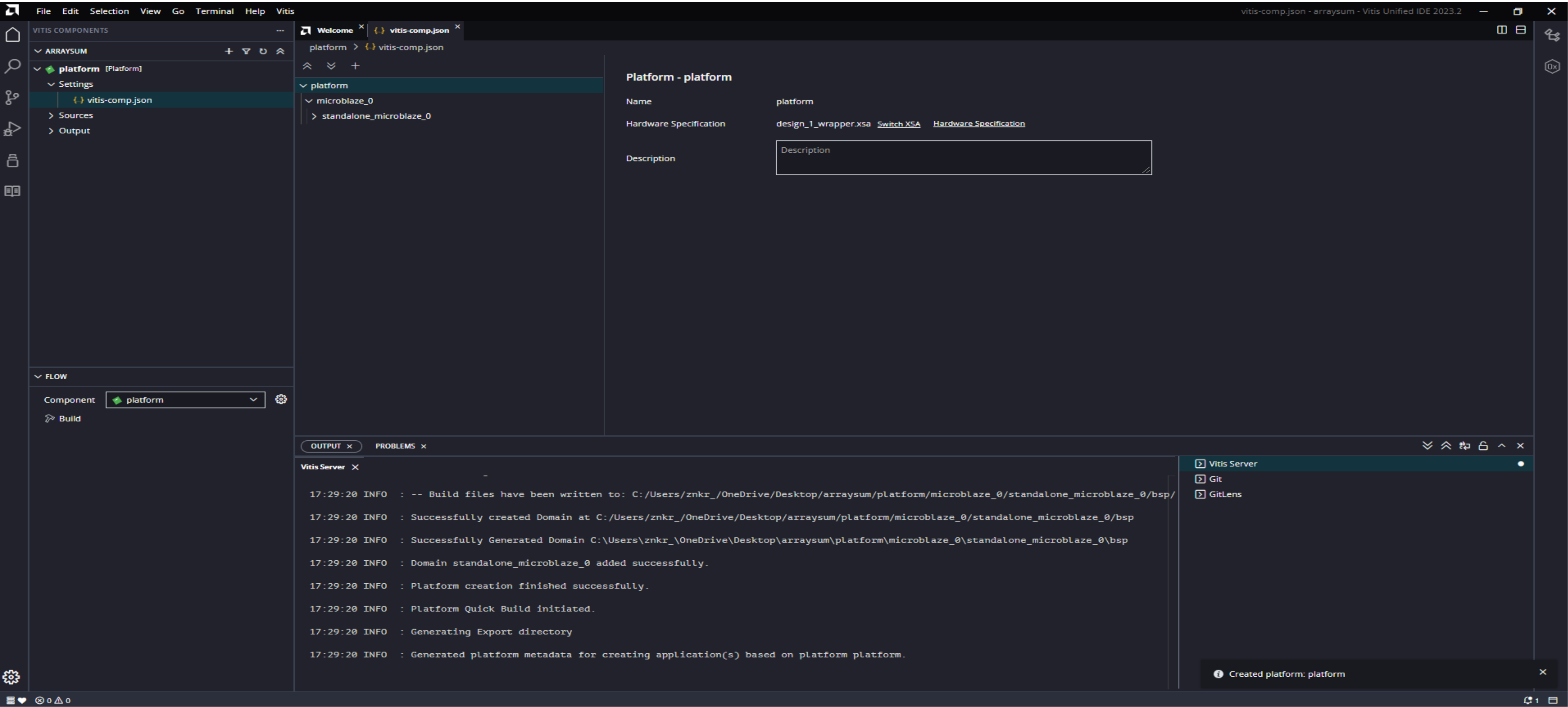
Click : Launch Vitis IDE .

(Make sure .xsa file is exported successfully).


arraysum - [C:/Users/znkr_/Desktop/arraysum/arraysum.xpr] - Vivado 2023.2

Open the project workspace and create a platform.

(Platform : based on the .xsa file).



Platform - platform

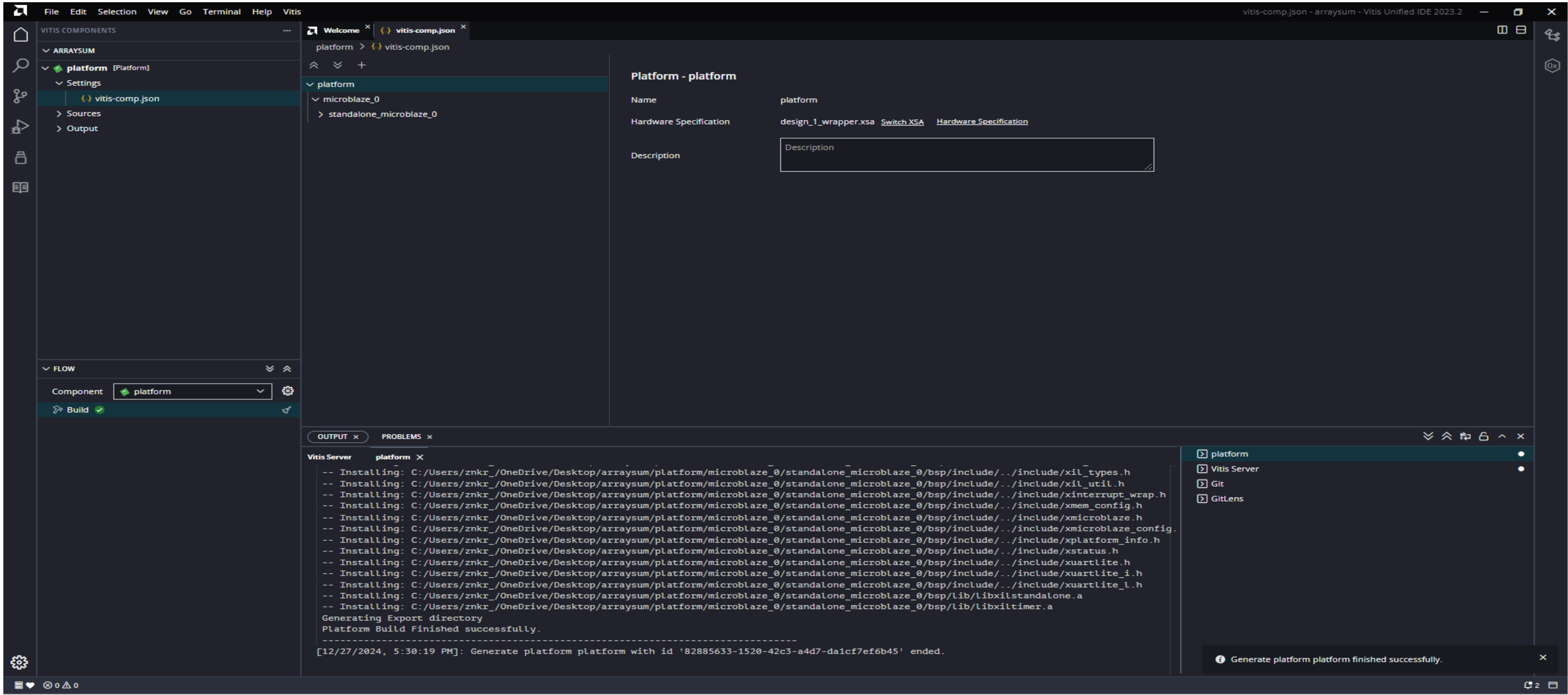
Name	platform
Hardware Specification	design_1_wrapper.xsa Switch XSA Hardware Specification
Description	Description

```
17:29:20 INFO  : -- Build files have been written to: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/
17:29:20 INFO  : Successfully created Domain at C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp
17:29:20 INFO  : Successfully Generated Domain C:\Users\znkr_\OneDrive\Desktop\arraysum\platform\microblaze_0\standalone_microblaze_0\bsp
17:29:20 INFO  : Domain standalone_microblaze_0 added successfully.
17:29:20 INFO  : Platform creation finished successfully.
17:29:20 INFO  : Platform Quick Build initiated.
17:29:20 INFO  : Generating Export directory
17:29:20 INFO  : Generated platform metadata for creating application(s) based on platform platform.
```

Created platform: platform

Click : Build.

(Build : To build the platform).



platform > vitis-comp.json

- platform
  - microblaze_0
    - standalone_microblaze_0

Platform - platform

Name	platform
Hardware Specification	design_1_wrapper.xsa Switch XSA Hardware Specification
Description	Description

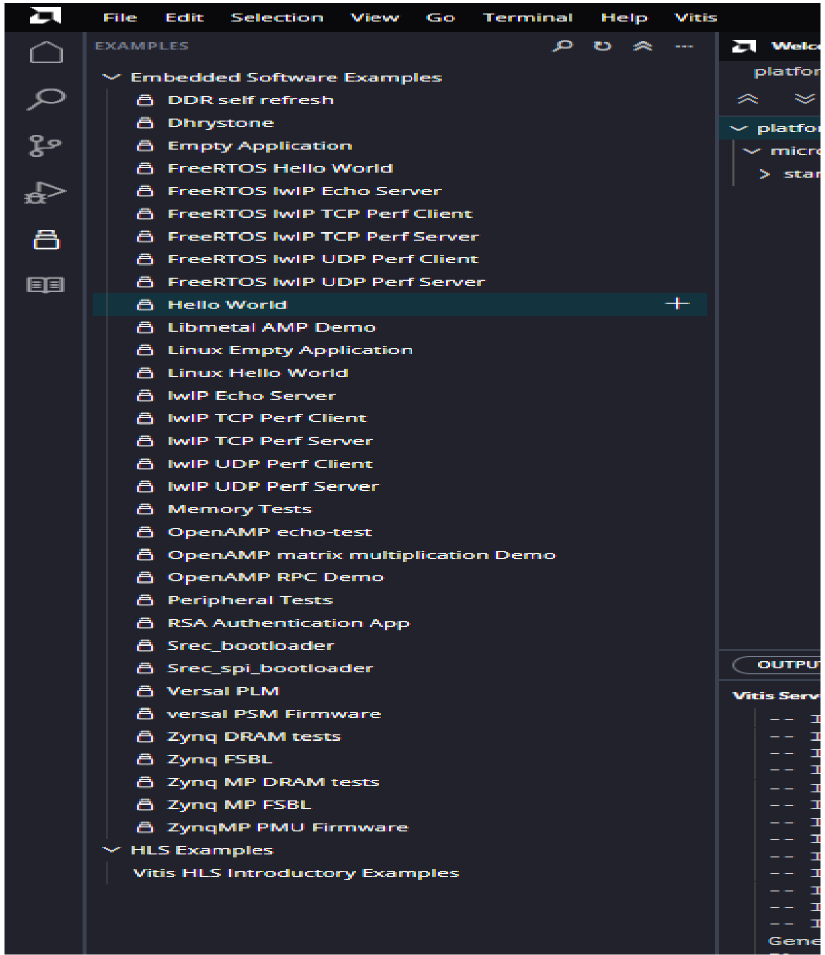
Component: platform

```
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xil_types.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xil_util.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xinterrupt_wrap.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xmem_config.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xmicroblaze.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xmicroblaze_config.
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xplatform_info.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xstatus.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xuartlite.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xuartlite_i.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/include/../include/xuartlite_l.h
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/lib/libxilstandalone.a
-- Installing: C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/microblaze_0/standalone_microblaze_0/bsp/lib/libxiltimer.a
Generating Export directory
Platform Build Finished successfully.
------------------------------------------------------------------------------
[12/27/2024, 5:30:19 PM]: Generate platform platform with id '82885633-1520-42c3-a4d7-da1cf7ef6b45' ended.
```

Generate platform platform finished successfully.

Select : Hello World Application From Templates.

(Renamed: Array_sum_sdk).



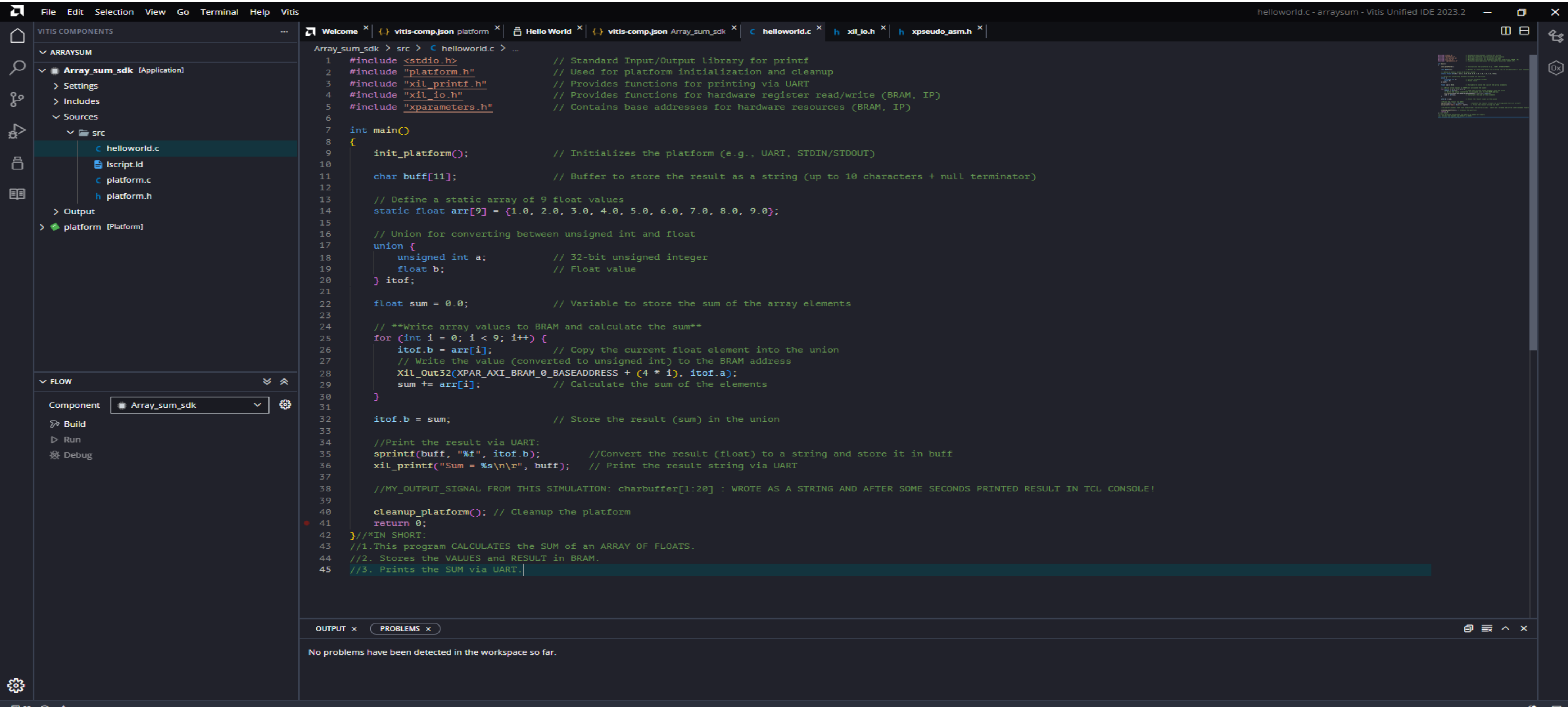
Embedded Software Examples
- DDR self refresh
- Dhrystone
- Empty Application
- FreeRTOS Hello World
- FreeRTOS lwIP Echo Server
- FreeRTOS lwIP TCP Perf Client
- FreeRTOS lwIP TCP Perf Server
- FreeRTOS lwIP UDP Perf Client
- FreeRTOS lwIP UDP Perf Server
- Hello World
- Libmetal AMP Demo
- Linux Empty Application
- Linux Hello World
- lwIP Echo Server
- lwIP TCP Perf Client
- lwIP TCP Perf Server
- lwIP UDP Perf Client
- lwIP UDP Perf Server
- Memory Tests
- OpenAMP echo-test
- OpenAMP matrix multiplication Demo
- OpenAMP RPC Demo
- Peripheral Tests
- RSA Authentication App
- Srec_bootloader
- Srec_spi_bootloader
- Versal PLM
- versal PSM Firmware
- Zynq DRAM tests
- Zynq FSBL
- Zynq MP DRAM tests
- Zynq MP FSBL
- ZynqMP PMU Firmware

HLS Examples
- Vitis HLS Introductory Examples

Function Sprintf:
Converts:
The result (float) to a string.

Program : helloworld.c

(Calculates the sum of an array of nine floats, stores the result in BRAM and prints the sum via UART).

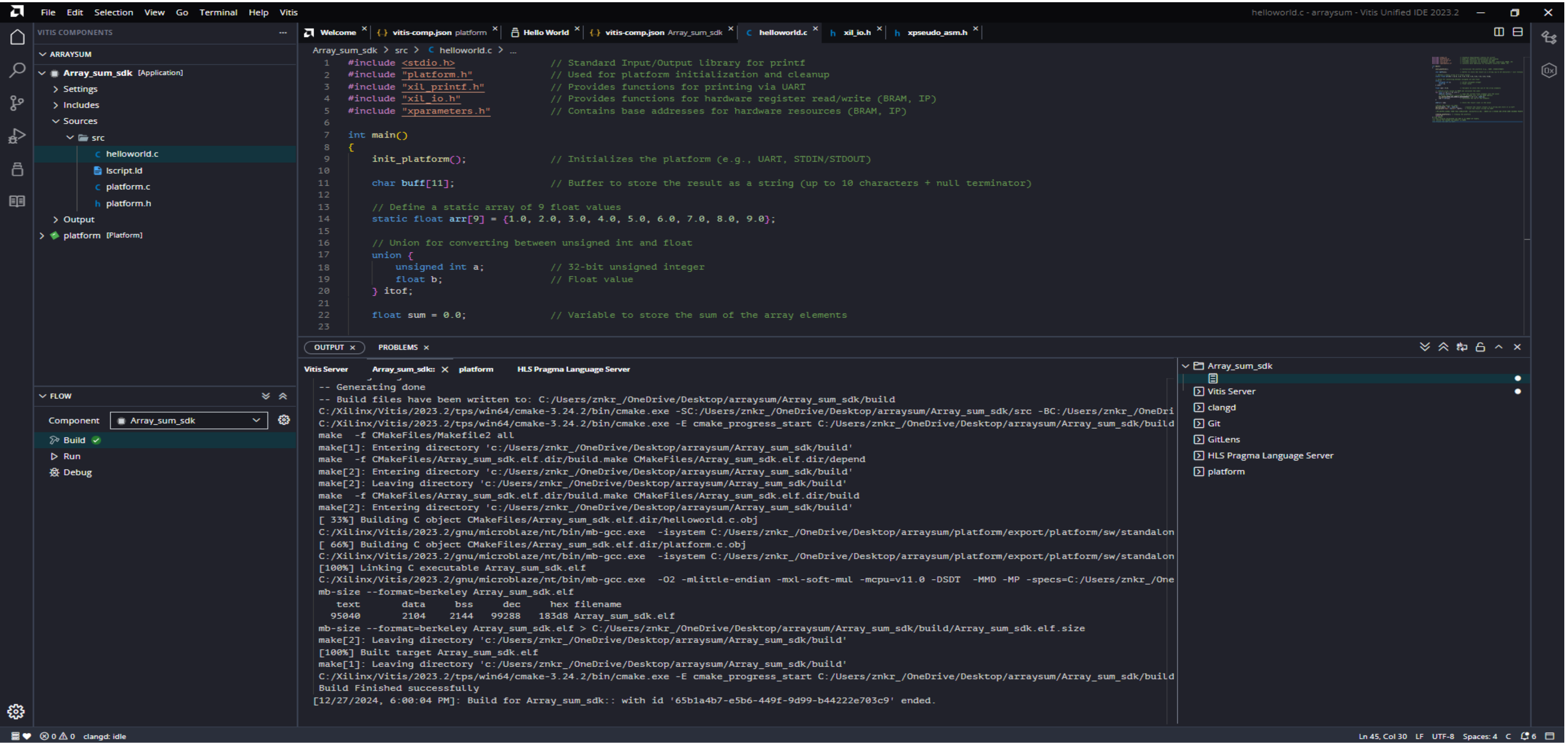


```c
#include <stdio.h>                  // Standard Input/Output library for printf
#include "platform.h"               // Used for platform initialization and cleanup
#include "xil_printf.h"             // Provides functions for printing via UART
#include "xil_io.h"                 // Provides functions for hardware register read/write (BRAM, IP)
#include "xparameters.h"            // Contains base addresses for hardware resources (BRAM, IP)

int main()
{
    init_platform();                // Initializes the platform (e.g., UART, STDIN/STDOUT)

    char buff[11];                  // Buffer to store the result as a string (up to 10 characters + null terminator)

    // Define a static array of 9 float values
    static float arr[9] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0};

    // Union for converting between unsigned int and float
    union {
        unsigned int a;             // 32-bit unsigned integer
        float b;                    // Float value
    } itof;

    float sum = 0.0;                // Variable to store the sum of the array elements

    // **Write array values to BRAM and calculate the sum**
    for (int i = 0; i < 9; i++) {
        itof.b = arr[i];            // Copy the current float element into the union
        // Write the value (converted to unsigned int) to the BRAM address
        Xil_Out32(XPAR_AXI_BRAM_0_BASEADDRESS + (4 * i), itof.a);
        sum += arr[i];              // Calculate the sum of the elements
    }

    itof.b = sum;                   // Store the result (sum) in the union

    //Print the result via UART:
    sprintf(buff, "%f", itof.b);         //Convert the result (float) to a string and store it in buff
    xil_printf("Sum = %s\n\r", buff);    // Print the result string via UART

    //MY_OUTPUT_SIGNAL FROM THIS SIMULATION: charbuffer[1:20] : WROTE AS A STRING AND AFTER SOME SECONDS PRINTED RESULT IN TCL CONSOLE!

    cleanup_platform(); // Cleanup the platform
    return 0;
}//*IN SHORT:
//1.This program CALCULATES the SUM of an ARRAY OF FLOATS.
//2. Stores the VALUES and RESULT in BRAM.
//3. Prints the SUM via UART.
```

No problems have been detected in the workspace so far.

Click : Build.

(Build : To build the Application and generate ELF File).

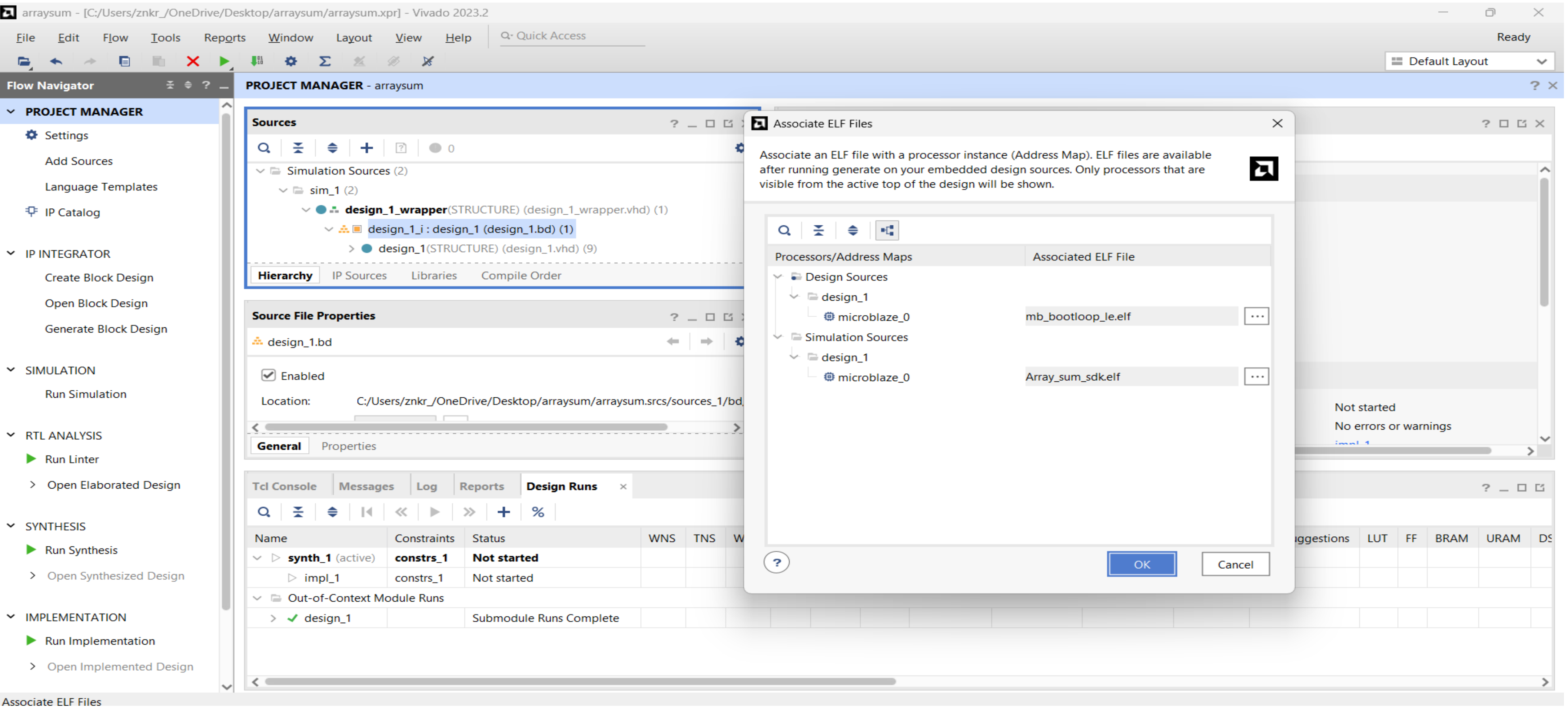


```c
#include <stdio.h>              // Standard Input/Output library for printf
#include "platform.h"           // Used for platform initialization and cleanup
#include "xil_printf.h"         // Provides functions for printing via UART
#include "xil_io.h"             // Provides functions for hardware register read/write (BRAM, IP)
#include "xparameters.h"        // Contains base addresses for hardware resources (BRAM, IP)

int main()
{
    init_platform();            // Initializes the platform (e.g., UART, STDIN/STDOUT)

    char buff[11];              // Buffer to store the result as a string (up to 10 characters + null terminator)

    // Define a static array of 9 float values
    static float arr[9] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0};

    // Union for converting between unsigned int and float
    union {
        unsigned int a;         // 32-bit unsigned integer
        float b;                // Float value
    } itof;

    float sum = 0.0;            // Variable to store the sum of the array elements
```

```
-- Generating done
-- Build files have been written to: C:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build
C:/Xilinx/Vitis/2023.2/tps/win64/cmake-3.24.2/bin/cmake.exe -SC:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/src -BC:/Users/znkr_/OneDri
C:/Xilinx/Vitis/2023.2/tps/win64/cmake-3.24.2/bin/cmake.exe -E cmake_progress_start C:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build
make  -f CMakeFiles/Makefile2 all
make[1]: Entering directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
make  -f CMakeFiles/Array_sum_sdk.elf.dir/build.make CMakeFiles/Array_sum_sdk.elf.dir/depend
make[2]: Entering directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
make[2]: Leaving directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
make  -f CMakeFiles/Array_sum_sdk.elf.dir/build.make CMakeFiles/Array_sum_sdk.elf.dir/build
make[2]: Entering directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
[ 33%] Building C object CMakeFiles/Array_sum_sdk.elf.dir/helloworld.c.obj
C:/Xilinx/Vitis/2023.2/gnu/microblaze/nt/bin/mb-gcc.exe  -isystem C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/export/platform/sw/standalon
[ 66%] Building C object CMakeFiles/Array_sum_sdk.elf.dir/platform.c.obj
C:/Xilinx/Vitis/2023.2/gnu/microblaze/nt/bin/mb-gcc.exe  -isystem C:/Users/znkr_/OneDrive/Desktop/arraysum/platform/export/platform/sw/standalon
[100%] Linking C executable Array_sum_sdk.elf
C:/Xilinx/Vitis/2023.2/gnu/microblaze/nt/bin/mb-gcc.exe  -O2 -mlittle-endian -mxl-soft-mul -mcpu=v11.0 -DSDT  -MMD -MP -specs=C:/Users/znkr_/One
mb-size --format=berkeley Array_sum_sdk.elf
   text       data     bss     dec     hex filename
  95040       2104    2144   99288   183d8 Array_sum_sdk.elf
mb-size --format=berkeley Array_sum_sdk.elf > C:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build/Array_sum_sdk.elf.size
make[2]: Leaving directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
[100%] Built target Array_sum_sdk.elf
make[1]: Leaving directory 'c:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build'
C:/Xilinx/Vitis/2023.2/tps/win64/cmake-3.24.2/bin/cmake.exe -E cmake_progress_start C:/Users/znkr_/OneDrive/Desktop/arraysum/Array_sum_sdk/build
Build Finished successfully
[12/27/2024, 6:00:04 PM]: Build for Array_sum_sdk:: with id '65b1a4b7-e5b6-449f-9d99-b44222e703c9' ended.
```

Right-click : Block design <u>and</u> associate ELF files.

<u>(Apply the ELF file from SDK only to the simulation sources).</u>



Sources:

- Simulation Sources (2)
  - sim_1 (2)
    - **design_1_wrapper**(STRUCTURE) (design_1_wrapper.vhd) (1)
      - design_1_i : design_1 (design_1.bd) (1)
        - design_1(STRUCTURE) (design_1.vhd) (9)

Source File Properties: design_1.bd

Enabled ✓

Location: C:/Users/znkr_/OneDrive/Desktop/arraysum/arraysum.srcs/sources_1/bd

Associate ELF Files

Associate an ELF file with a processor instance (Address Map). ELF files are available after running generate on your embedded design sources. Only processors that are visible from the active top of the design will be shown.

Processors/Address Maps	Associated ELF File
Design Sources	
design_1	
microblaze_0	mb_bootloop_le.elf
Simulation Sources	
design_1	
microblaze_0	Array_sum_sdk.elf

Design Runs:

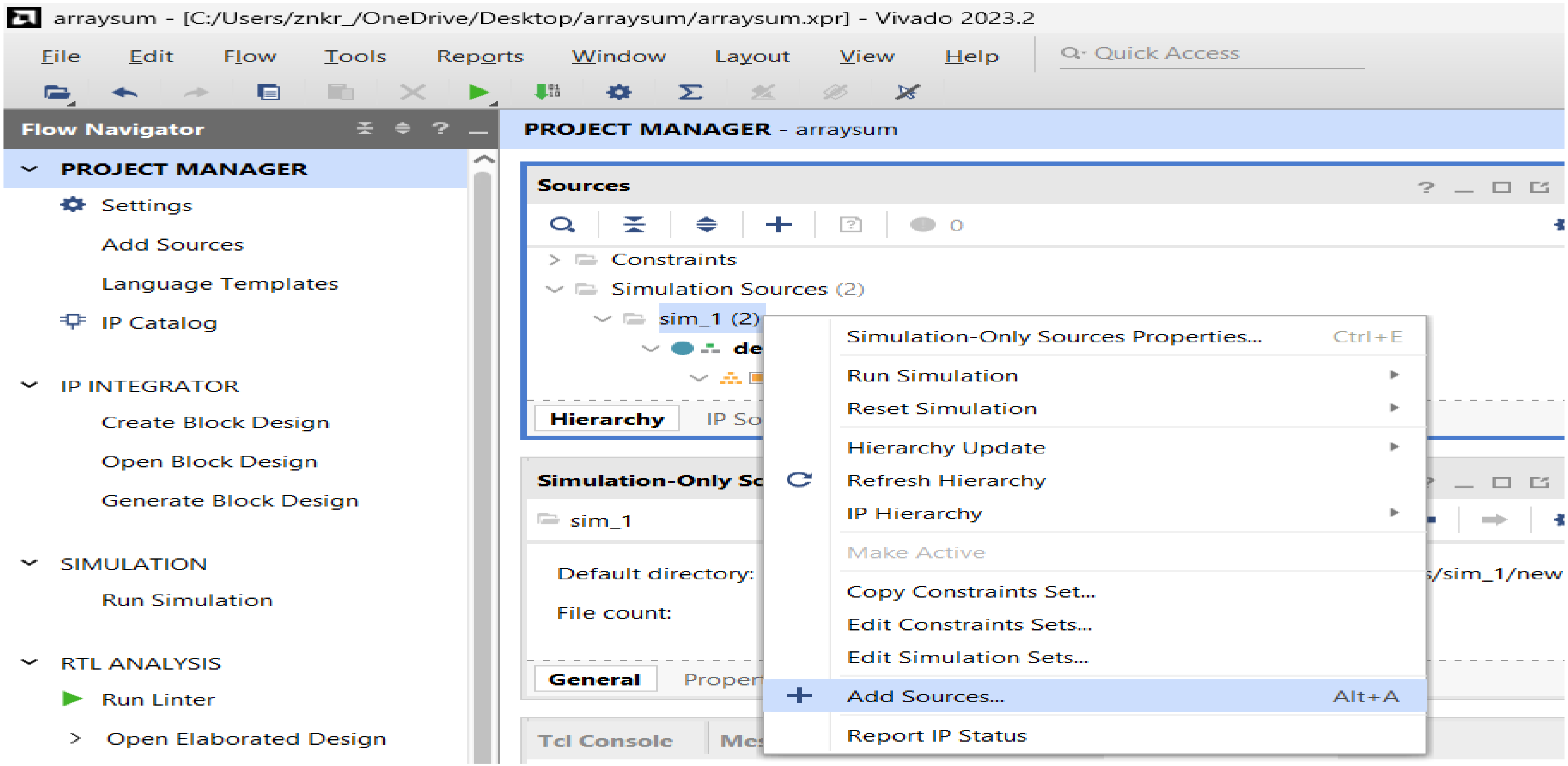
Name	Constraints	Status
synth_1 (active)	constrs_1	Not started
impl_1	constrs_1	Not started
Out-of-Context Module Runs		
design_1		Submodule Runs Complete

Columns: WNS, TNS, W, ...ggestions, LUT, FF, BRAM, URAM, DS

Not started

No errors or warnings

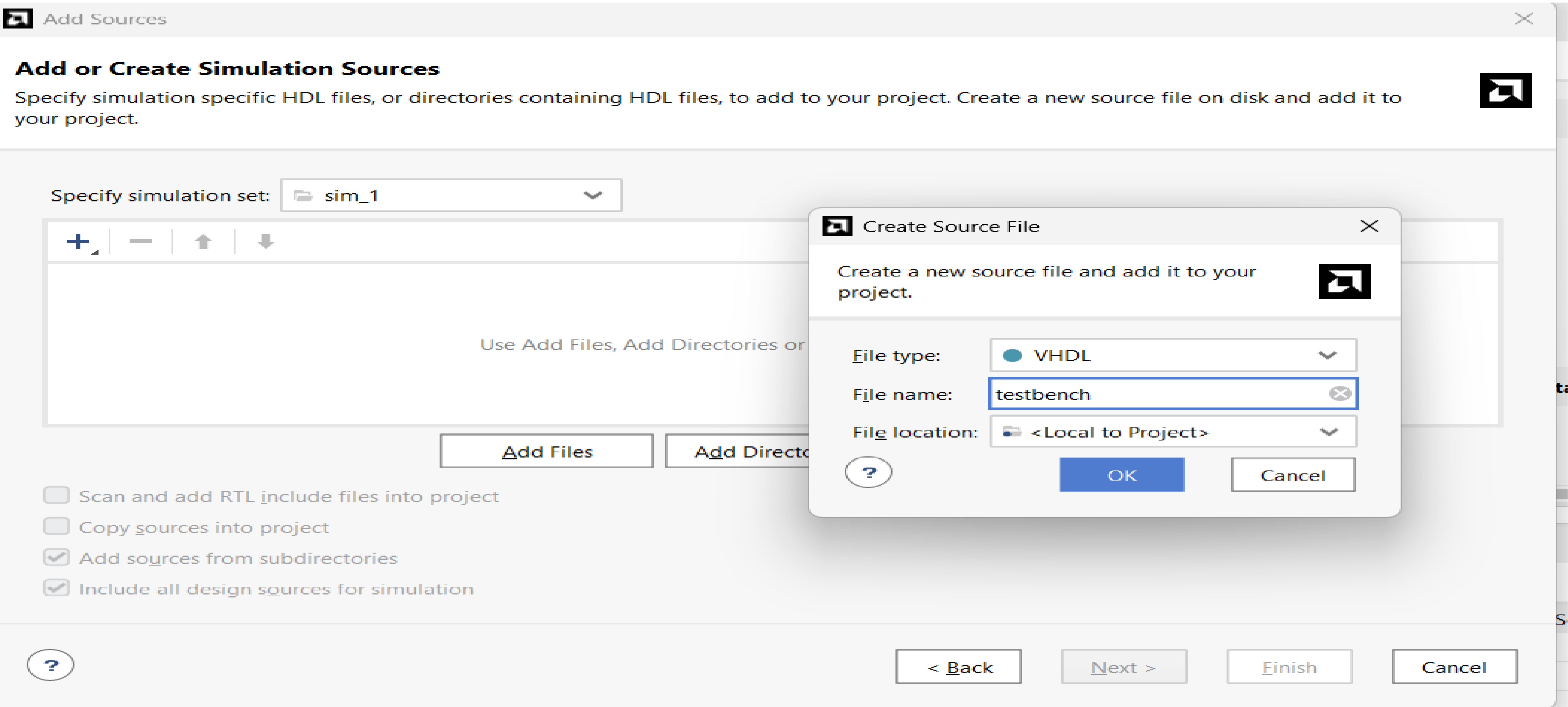
Right-click : Simulation Sources <u>and</u> Add Sources…



arraysum - [C:/Users/znkr_/OneDrive/Desktop/arraysum/arraysum.xpr] - Vivado 2023.2

Click : Create Source File.

(File name : testbench).



Add Sources

Add or Create Simulation Sources

Specify simulation specific HDL files, or directories containing HDL files, to add to your project. Create a new source file on disk and add it to your project.

Specify simulation set: sim_1

Use Add Files, Add Directories or

Add Files | Add Directo

- [ ] Scan and add RTL include files into project
- [ ] Copy sources into project
- [x] Add sources from subdirectories
- [x] Include all design sources for simulation

< Back | Next > | Finish | Cancel

Create Source File

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

File type: VHDL

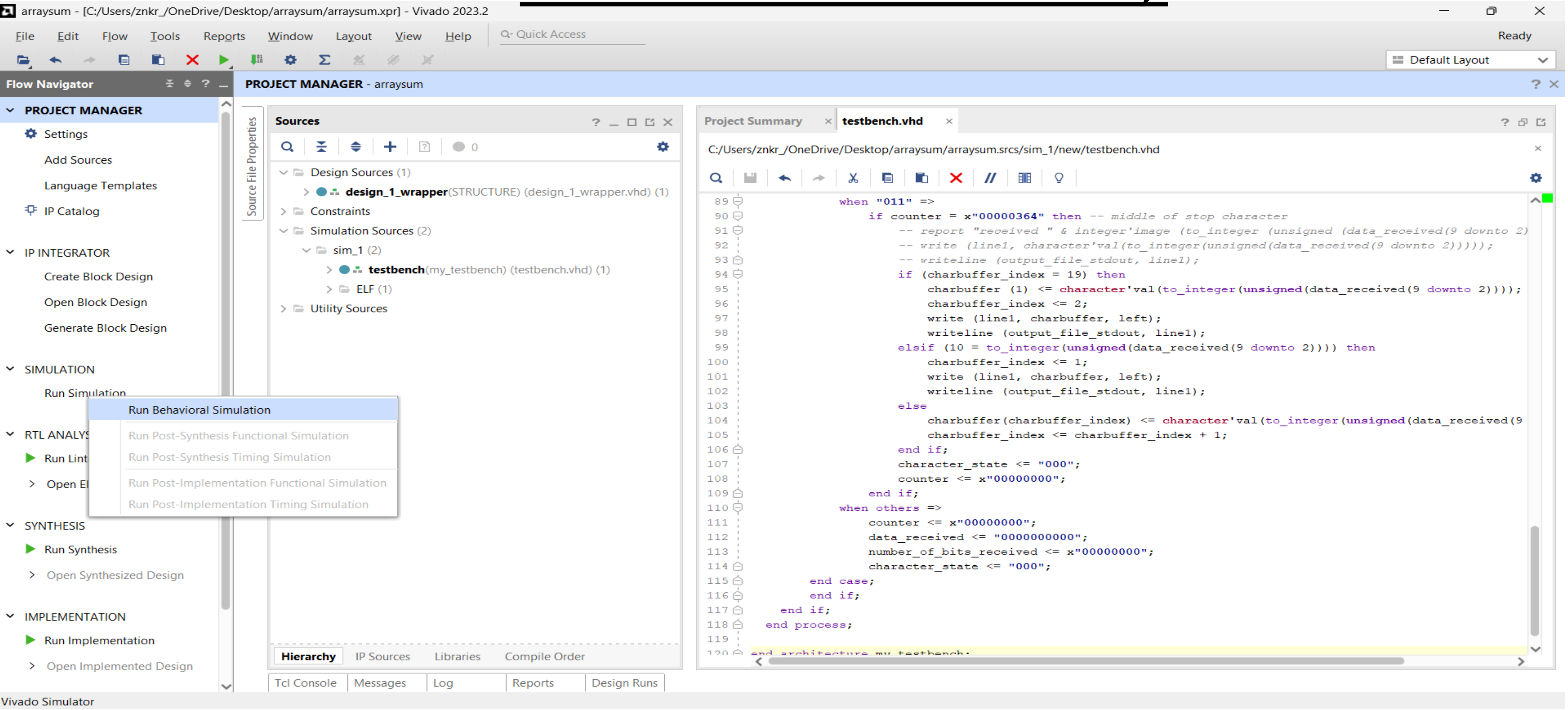
File name: testbench

File location: <Local to Project>

OK | Cancel

Click : Run Behavioral Simulation.

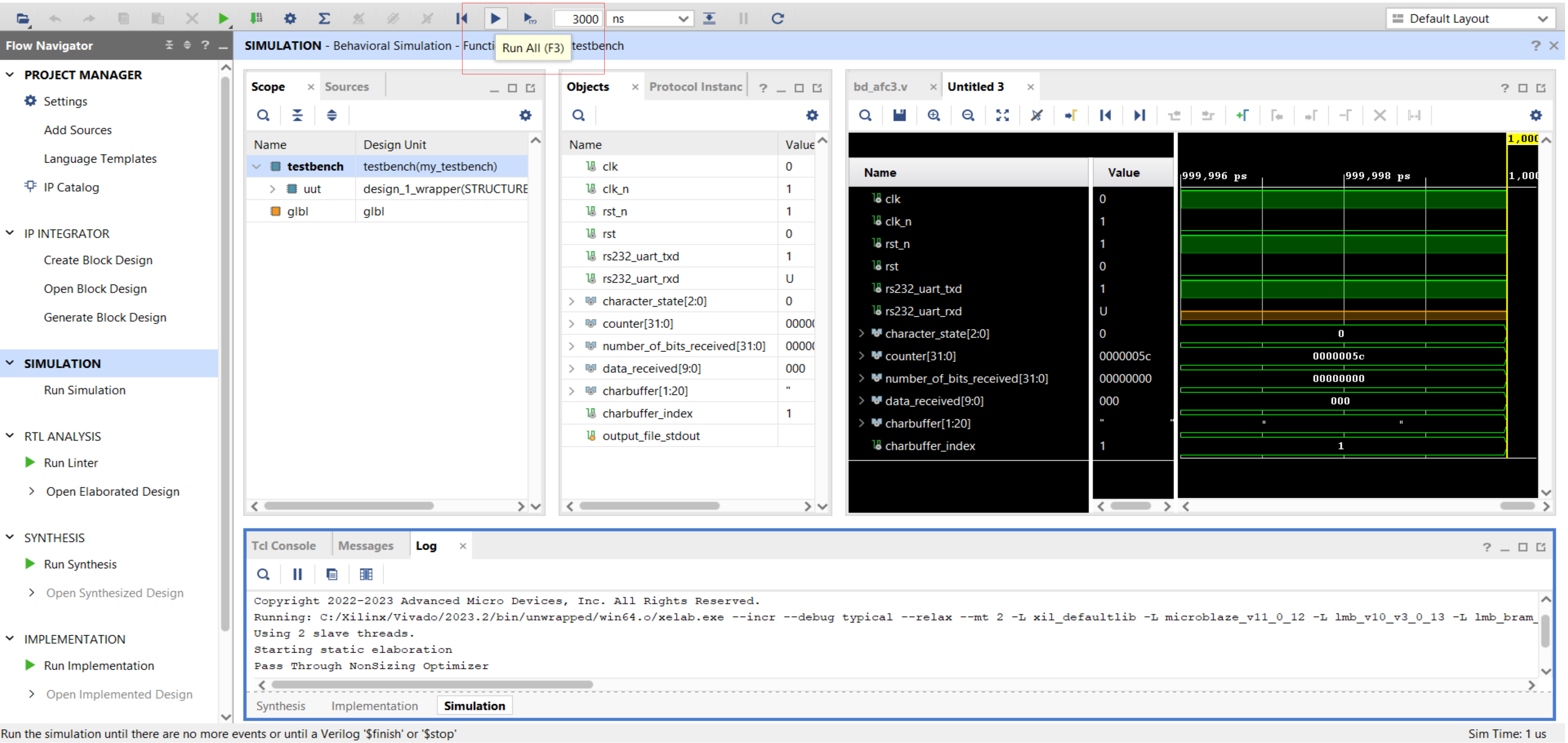
(The testbench contains VHDL code for UART communication and displaying received data in the TCL Console).


arraysum - [C:/Users/znkr_/OneDrive/Desktop/arraysum/arraysum.xpr] - Vivado 2023.2
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Ready
Default Layout
Flow Navigator
PROJECT MANAGER - arraysum
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
Run Linter
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional Simulation
Run Post-Implementation Timing Simulation
Source File Properties
Sources
Design Sources (1)
design_1_wrapper(STRUCTURE) (design_1_wrapper.vhd) (1)
Constraints
Simulation Sources (2)
sim_1 (2)
testbench(my_testbench) (testbench.vhd) (1)
ELF (1)
Utility Sources
Hierarchy IP Sources Libraries Compile Order
Tcl Console Messages Log Reports Design Runs
Project Summary
testbench.vhd
C:/Users/znkr_/OneDrive/Desktop/arraysum/arraysum.srcs/sim_1/new/testbench.vhd
Vivado Simulator

```
            when "011" =>
                if counter = x"00000364" then -- middle of stop character
                    -- report "received " & integer'image (to_integer (unsigned (data_received(9 downto 2)
                    -- write (line1, character'val(to_integer(unsigned(data_received(9 downto 2)))));
                    -- writeline (output_file_stdout, line1);
                    if (charbuffer_index = 19) then
                        charbuffer (1) <= character'val(to_integer(unsigned(data_received(9 downto 2))));
                        charbuffer_index <= 2;
                        write (line1, charbuffer, left);
                        writeline (output_file_stdout, line1);
                    elsif (10 = to_integer(unsigned(data_received(9 downto 2)))) then
                        charbuffer_index <= 1;
                        write (line1, charbuffer, left);
                        writeline (output_file_stdout, line1);
                    else
                        charbuffer(charbuffer_index) <= character'val(to_integer(unsigned(data_received(9
                        charbuffer_index <= charbuffer_index + 1;
                    end if;
                    character_state <= "000";
                    counter <= x"00000000";
                end if;
            when others =>
                counter <= x"00000000";
                data_received <= "0000000000";
                number_of_bits_received <= x"00000000";
                character_state <= "000";
        end case;
        end if;
    end if;
end process;

end architecture my_testbench;
```

Click : Run All (F3).

(Simulation launched).



SIMULATION - Behavioral Simulation - Functi… testbench

Name	Design Unit
testbench	testbench(my_testbench)
uut	design_1_wrapper(STRUCTURE
glbl	glbl

Name	Value
clk	0
clk_n	1
rst_n	1
rst	0
rs232_uart_txd	1
rs232_uart_rxd	U
character_state[2:0]	0
counter[31:0]	0000005c
number_of_bits_received[31:0]	00000000
data_received[9:0]	000
charbuffer[1:20]	"
charbuffer_index	1
output_file_stdout	

999,996 ps 999,998 ps 1,000

```
Copyright 2022-2023 Advanced Micro Devices, Inc. All Rights Reserved.
Running: C:/Xilinx/Vivado/2023.2/bin/unwrapped/win64.o/xelab.exe --incr --debug typical --relax --mt 2 -L xil_defaultlib -L microblaze_v11_0_12 -L lmb_v10_v3_0_13 -L lmb_bram_
Using 2 slave threads.
Starting static elaboration
Pass Through NonSizing Optimizer
```

Run the simulation until there are no more events or until a Verilog '$finish' or '$stop'

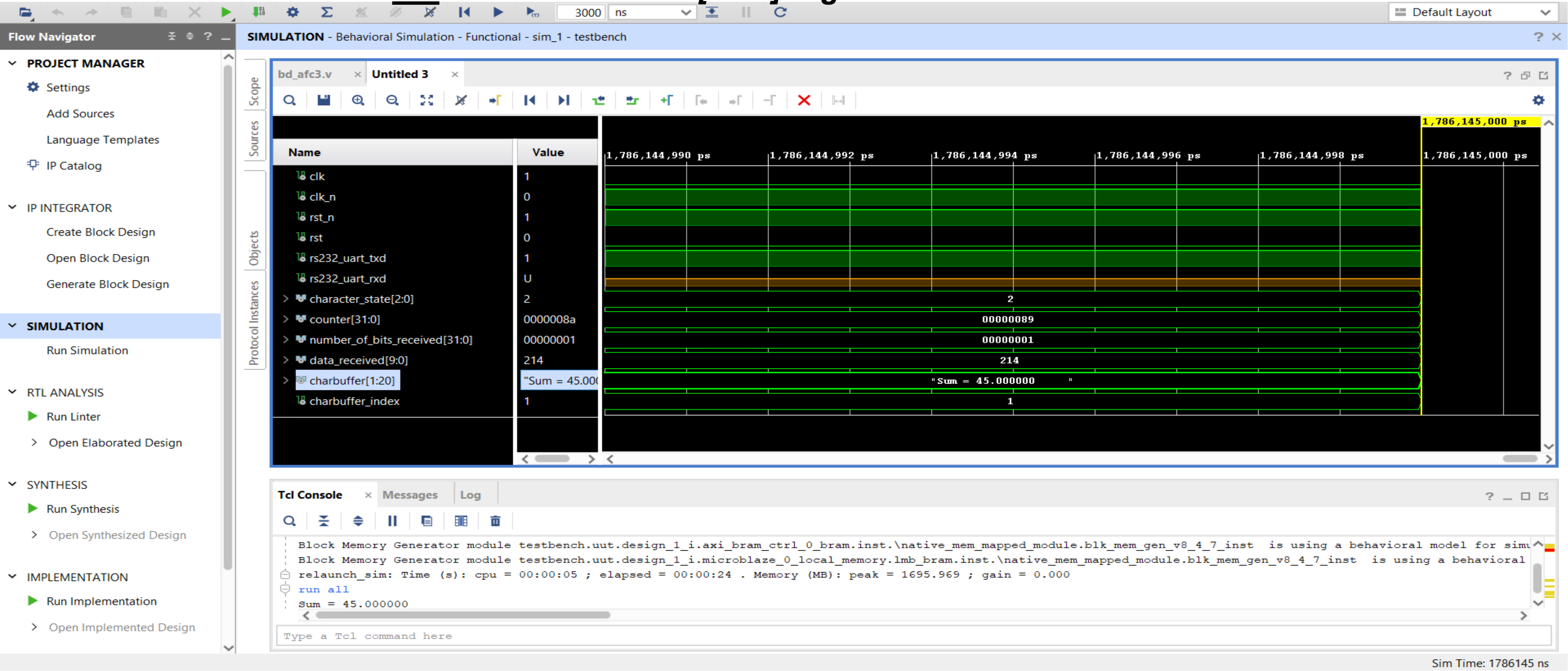
Sim Time: 1 us

It may take some time; please be patient.

Click : TCL Console to see the result.

(Sum = 45.000000).

OR Click: charbuffer[1:20] Signal in the waveform viewer.



SIMULATION - Behavioral Simulation - Functional - sim_1 - testbench

Name	Value
clk	1
clk_n	0
rst_n	1
rst	0
rs232_uart_txd	1
rs232_uart_rxd	U
character_state[2:0]	2
counter[31:0]	0000008a
number_of_bits_received[31:0]	00000001
data_received[9:0]	214
charbuffer[1:20]	"Sum = 45.00(
charbuffer_index	1

```
1,786,145,000 ps
1,786,144,990 ps   1,786,144,992 ps   1,786,144,994 ps   1,786,144,996 ps   1,786,144,998 ps   1,786,145,000 ps
2
00000089
00000001
214
"Sum = 45.000000      "
1
```

```
Block Memory Generator module testbench.uut.design_1_i.axi_bram_ctrl_0_bram.inst.\native_mem_mapped_module.blk_mem_gen_v8_4_7_inst  is using a behavioral model for simu
Block Memory Generator module testbench.uut.design_1_i.microblaze_0_local_memory.lmb_bram.inst.\native_mem_mapped_module.blk_mem_gen_v8_4_7_inst  is using a behavioral
relaunch_sim: Time (s): cpu = 00:00:05 ; elapsed = 00:00:24 . Memory (MB): peak = 1695.969 ; gain = 0.000
run all
Sum = 45.000000
```

Sim Time: 1786145 ns

Thank you for your participation.