

Versal Dhrystone Benchmark user guide

Table of Contents

- [Introduction](#)
- [Before You Begin](#)
- [Tutorial Requirements](#)
- [Build Tutorial Design](#)
- [Create a new application project for Dhrystone](#)
- [Build Dhrystone application](#)
- [Run the Dhrystone application](#)
- [Performance calculation](#)

Introduction

Versal™ ACAP combines adaptable processing and acceleration engines with programmable logic and configurable connectivity to enable custom, heterogeneous hardware solutions for a wide variety of applications in data center, automotive, 5G wireless, wired network, and defense.

This tutorial provides step-by-step instructions for generating a reference design for the Dhrystone benchmark and building and running the Dhrystone application.

Objectives

After completing this tutorial, users can:

- Generate programmable device image (PDI) for tutorial design.
- Build a Dhrystone application and execute it on the VCK190 evaluation kit.
- Calculate Dhrystone performance number.

Directory structure

```

├── Dhrystone
│   ├── Design.....Contains Vivado design scripts
│   │   └── design.tcl.....Generates reference design
PDI/XSA
│   └── run.tcl.....Top tcl for project setup,
calls design.tcl
│   ├── Images.....Contains images that appear in README.md
│   │   ├── apu_clock_configuration.png.....APU clock configuration
│   │   ├── axi_noc_0_configuration_ddr_basic.png.....DDR basic configuration
│   │   └── axi_noc_0_connectivity.png.....NoC0 connectivity

```

```

|   |─ axi_noc_0_ddr_configuration.png.....DDR memory configuration
|   |─ axi_noc_0_general.png.....NoC0 general configuration
|   |─ axi_noc_0_inputs.png.....NoC0 input clock
configuration
|   |─ browse_and_add_xsa.png.....Add XSA
|   |─ browse_import_source_code_finish.png.....Complete importing source
code
|   |─ build_complete.png.....Build complete
|   |─ build_project.png.....Build project
|   |─ configure_domain_settings.png.....Domain settings
|   |─ create_a_new_application_project.png.....Start new application
project
|   |─ create_application_project.png.....Create new application
|   |─ create_empty_application_template.png.....Create application
template
|   |─ create_hardware_description.png.....Hardware description
window
|   |─ debug_level_none.png.....Add debug level
|   |─ download_and_run_dhrystone_application.png..Download and run Dhrystone
application
|   |─ expand_and_view_source_files.png.....Browse and view source
code
|   |─ import_source_code.png.....Import source code
|   |─ launch_xsct_and_connect_board.png.....Launch XSCT and connect to
board
|   |─ name_new_application_project.png.....Added new application
|   |─ optimization_optimize_most_03.png.....Add optimization level
|   |─ optimization_properties.png.....Go to optimization
properties
|   |─ processor_cips_block_diagram.png.....CIPS block diagram
|   |─ program_pdi.png.....Load the PDI over JTAG
|   |─ project_template.png.....Added project template
|   |─ select_a72_0_target_and_reset.png.....Select A72_0 and Reset
|   |─ source_run_tcl.png.....Source run.tcl in Vivado
|   |─ vck190_sw1_jtag_bootmode.png.....VCK190 JTAG boot mode
settings on SW1
|   |─ vck190_targets_list.png.....List the VCK190 targets
|─ README.md.....Includes tutorial overview
|─ Source_code.....Source code for Dhrystone application
|   |─ dhry_1.c
|   |─ dhry_2.c
|   |─ dhry.h
|   |─ LICENSE
|   |─ README.md

```

Prerequisites

Recommended general knowledge of:

- VCK190 evaluation board
- Versal JTAG boot mode
- Xilinx® Vivado® Design Suite

- Vitis™ Unified Software Platform Tool

Key Versal reference documents

- VCK190 Evaluation Board User Guide ([UG1366](#))
- Versal ACAP Technical Reference Manual ([AM011](#))
- Versal ACAP System Software Developers Guide ([UG1304](#))
- Control Interfaces and Processing System v3.0 (CIPS) ([PG352](#))

Key Dhrystone documents

- Dhrystone Benchmarking for ARM® Cortex® Processors - <https://developer.arm.com/documentation/dai0273/a/>
- Dhrystone Benchmark - https://www.eembc.org/techlit/datasheets/dhrystone_wp.pdf

Tutorial Requirements

This tutorial is demonstrated on the VCK190 evaluation kit. Install the necessary licenses for Vivado, Vitis, and XSCT/XSDB tools. Contact your Xilinx sales representative in case of any license issues. For more information, see <https://www.xilinx.com/products/boards-and-kits/vck190.html>.

Hardware Requirements

- A host machine with an operating system supported by Vivado Design Suite, Vitis tool, and XSCT/XSDB.
- VCK190 EV2 evaluation board, which includes:
 - Versal ACAP EK-VCK190-G-ED.
 - AC power adapter (100-240VAC input, 12VDC 15.0A output).
 - System controller microSD card in the socket (J302).
 - USB Type-C cable (for JTAG and UART communications).

Software Requirements

The following tools are necessary to build the tutorial design and execute the Dhrystone application:

- Vivado Design Suite and Vitis tool
 - For the latest tool version details, see <https://www.xilinx.com/support/download.html>.
 - For more information on installation, see [UG1400 Vitis Unified Software Platform Embedded Software Development](#).
- The [Build Tutorial Design](#) section of this document provides the scripts to create the tutorial design.
- UART serial terminal recommended:
 - Vitis serial Terminal or a terminal emulator program for UART (Putty or Tera Term) can be used to display the valuable PLM log boot status and the Dhrystone Benchmark logs.

Build the Tutorial Design

Follow these steps to build the Dhrystone Benchmark design and create the PDI/XSA.

1. Copy the Design directory and files to a local project directory. The following is a snippet of the top-level directory [Performance_Benchmark/Dhrystone/](#)

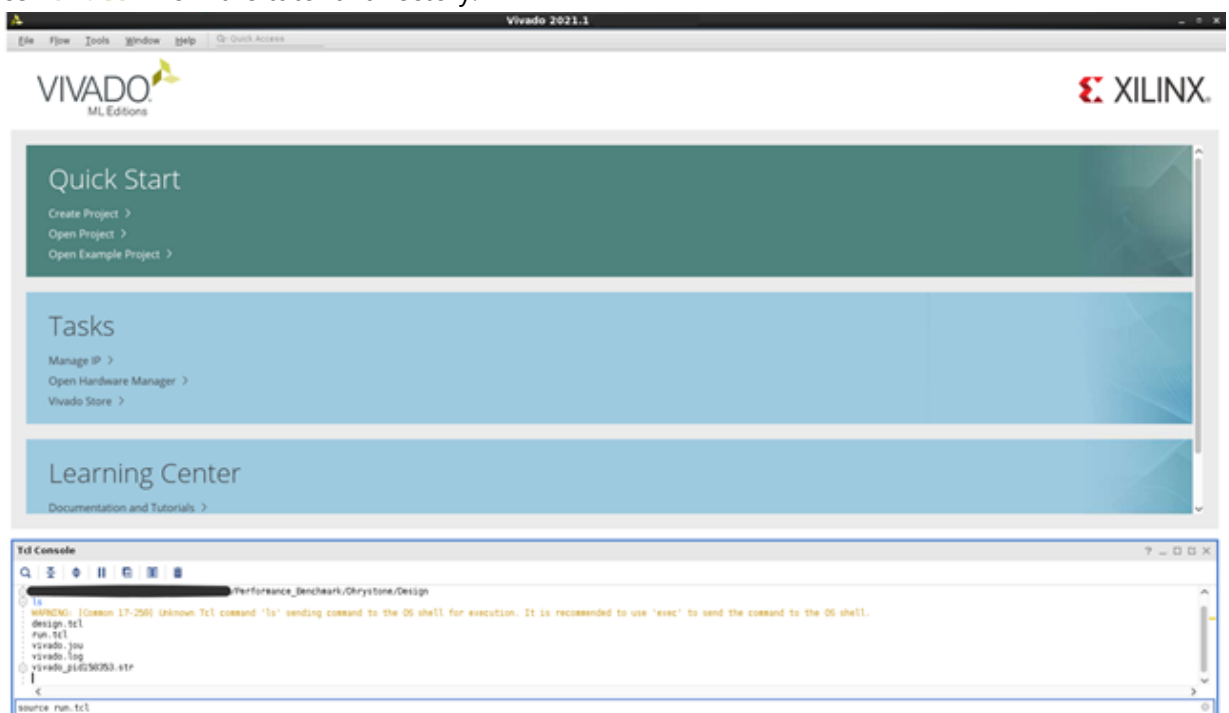
```
└─ Dhrystone
   └─ Design
      ├── design.tcl
      └─ run.tcl
```

2. Launch Vivado Design Suite.

3. In the Vivado Tcl console, cd to the tutorial directory:

```
/<Path to workspace>/Performance_Benchmark/Dhrystone/Design/).
```

4. Source `run.tcl` from the tutorial directory.



Sourcing the `run.tcl` script does the following:

- Creates a project directory
- Sources and runs the `design.tcl`, which does the following:
 - Selects the target Versal VC1902 device
 - Creates IPs and ports
 - Creates blocks
 - Configures and connects IP (Control, interfaces, and processing system (CIPS), Smartconnect)
 - Runs placement and routing
 - Creates a programmable device image (PDI) and Xilinx Support Archive (XSA).

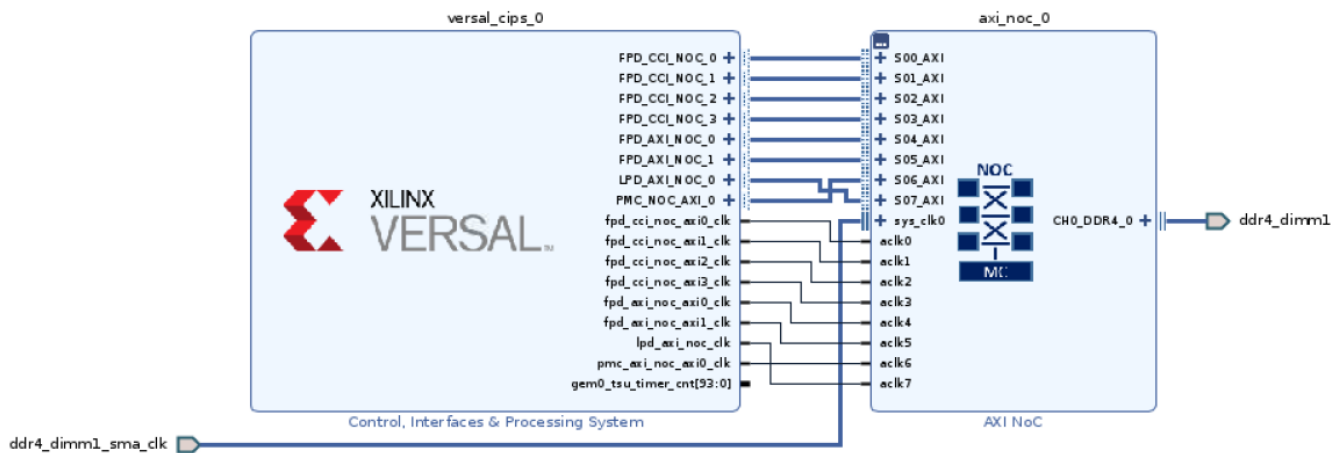
You can find PDI and XSA at:

```
PDI - /<path for
workspace>/Performance_Benchmark/Dhrystone/Design/runs/dperf_<*>/dhrystone_tutoria
```

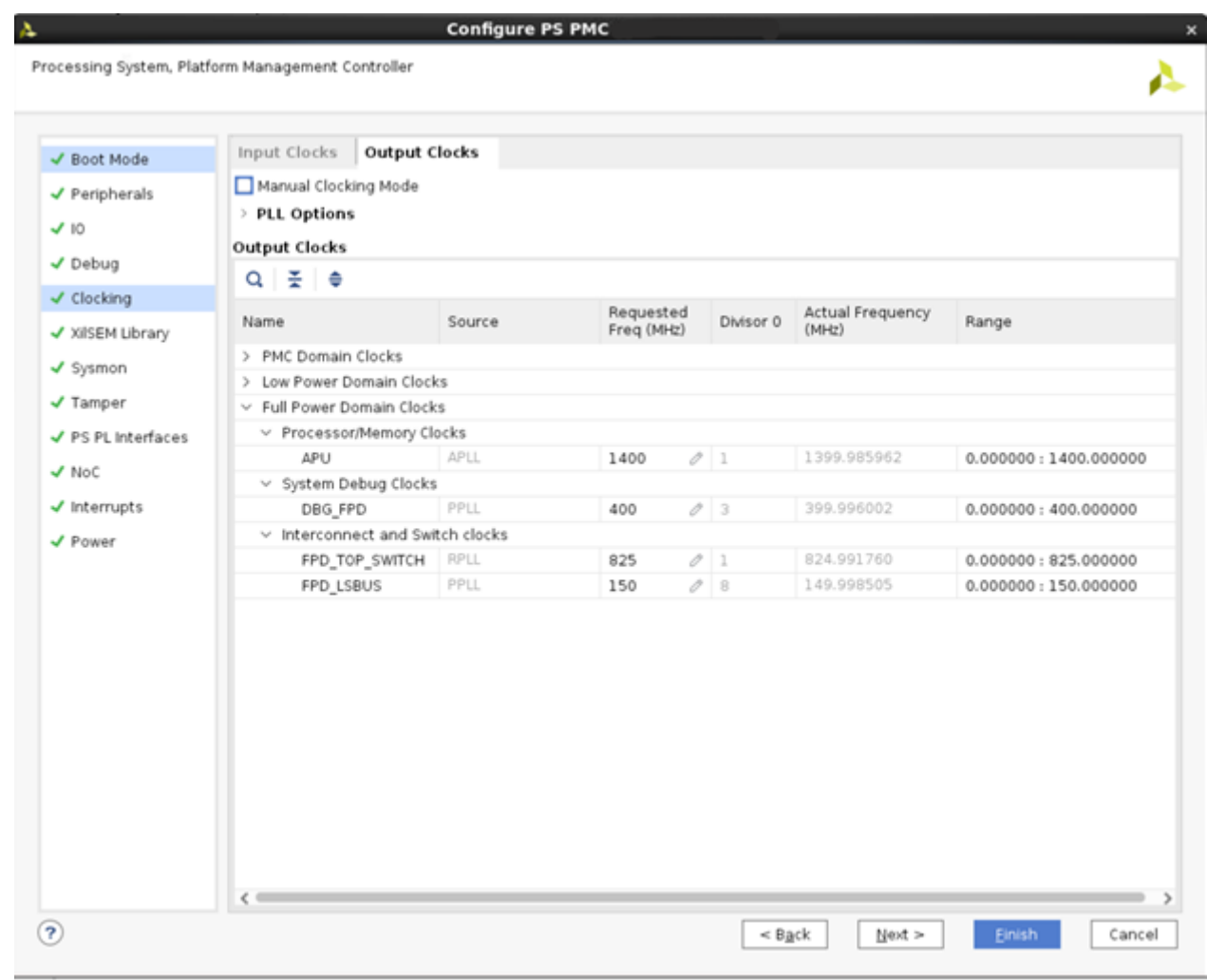
```
l.runs/impl_1/dhrystone_perf_wrapper.pdi
XSA - /<path for
workspace>/Performance_Benchmark/Dhrystone/Design/dhrystone_tutorial.xsa
```

Hardware Design Details

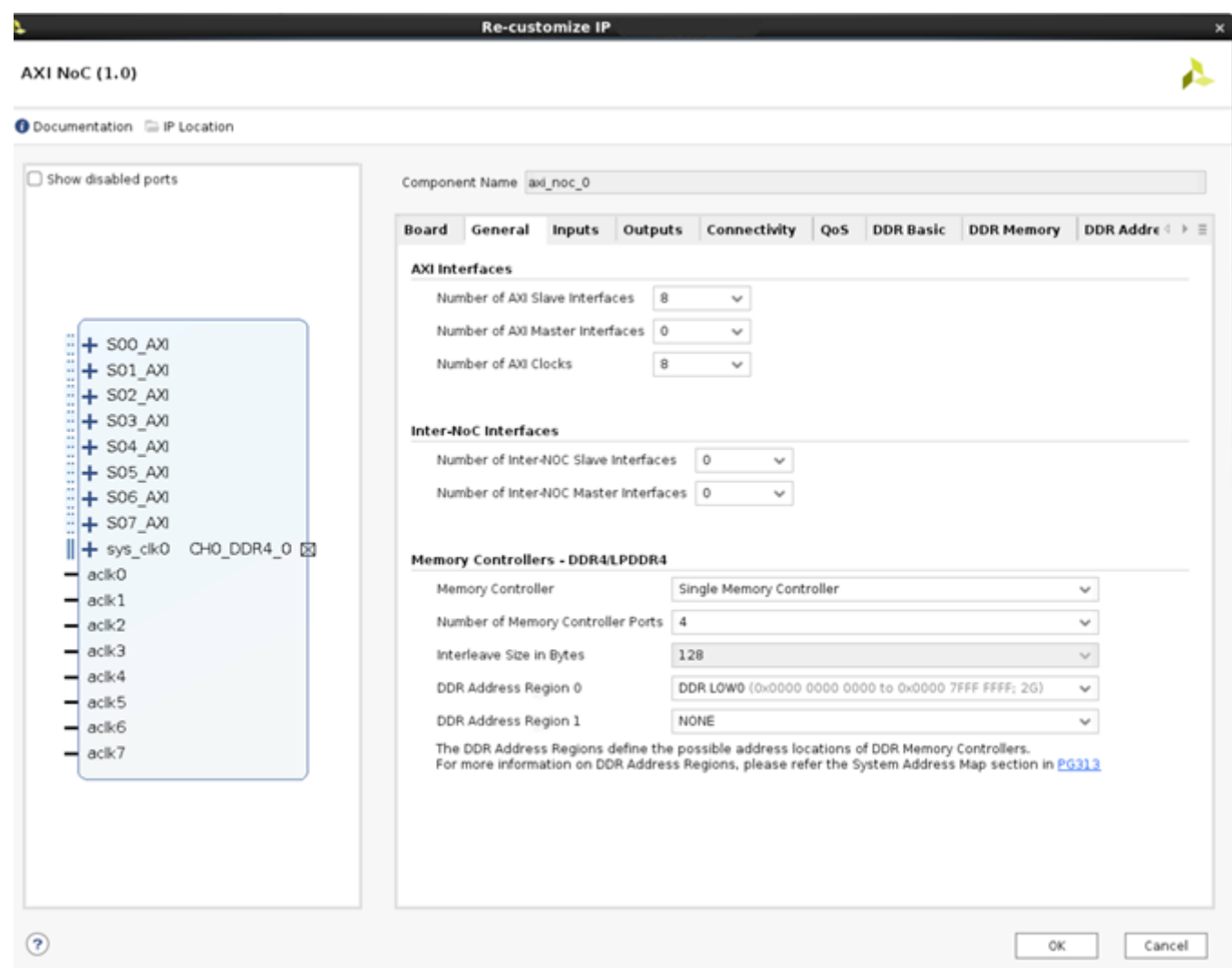
The tutorial design creates a block design with CIPS-IP and NoC IP upon sourcing the `run.tcl` script. The following image shows the details of the IP configuration.



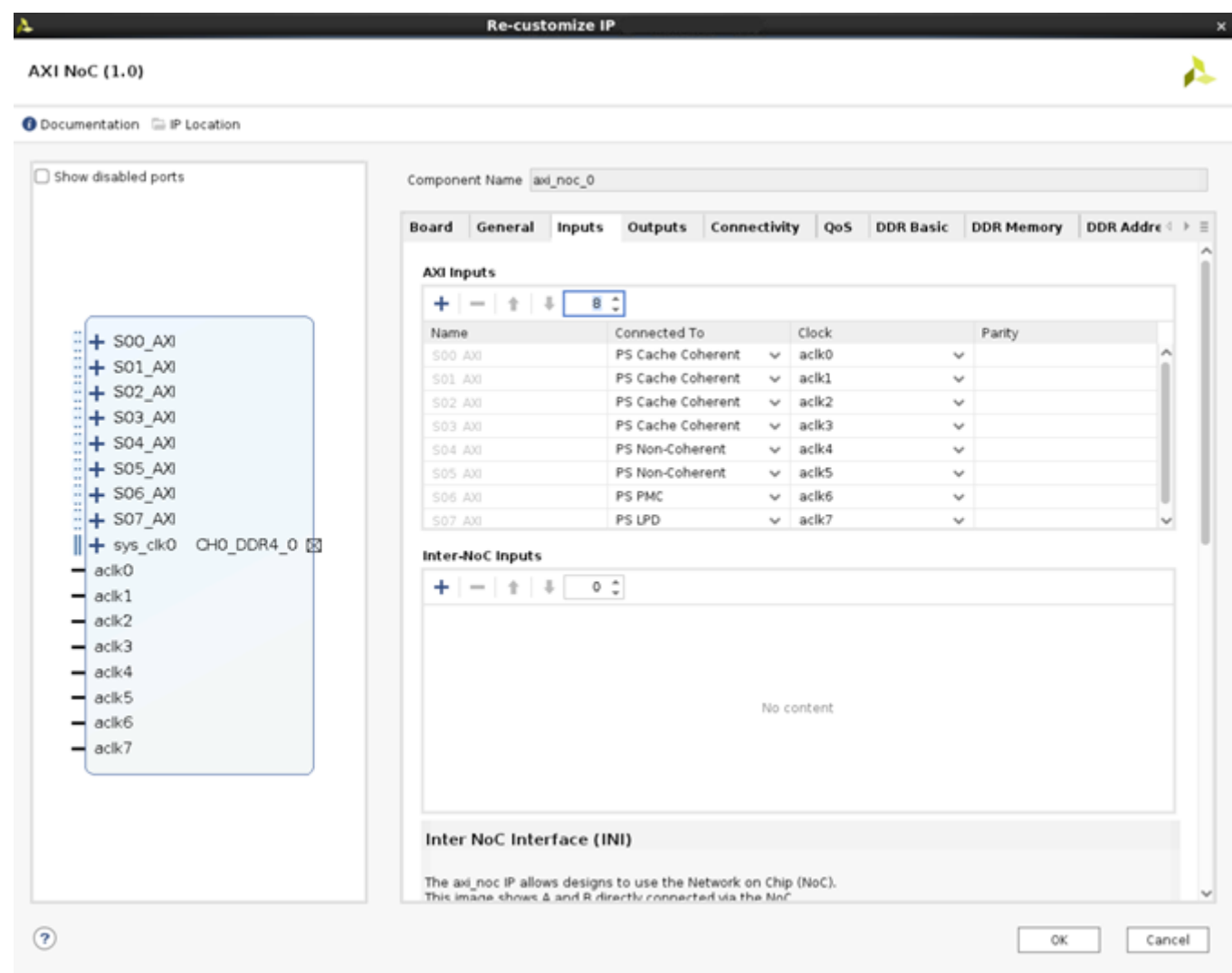
APU clock configuration



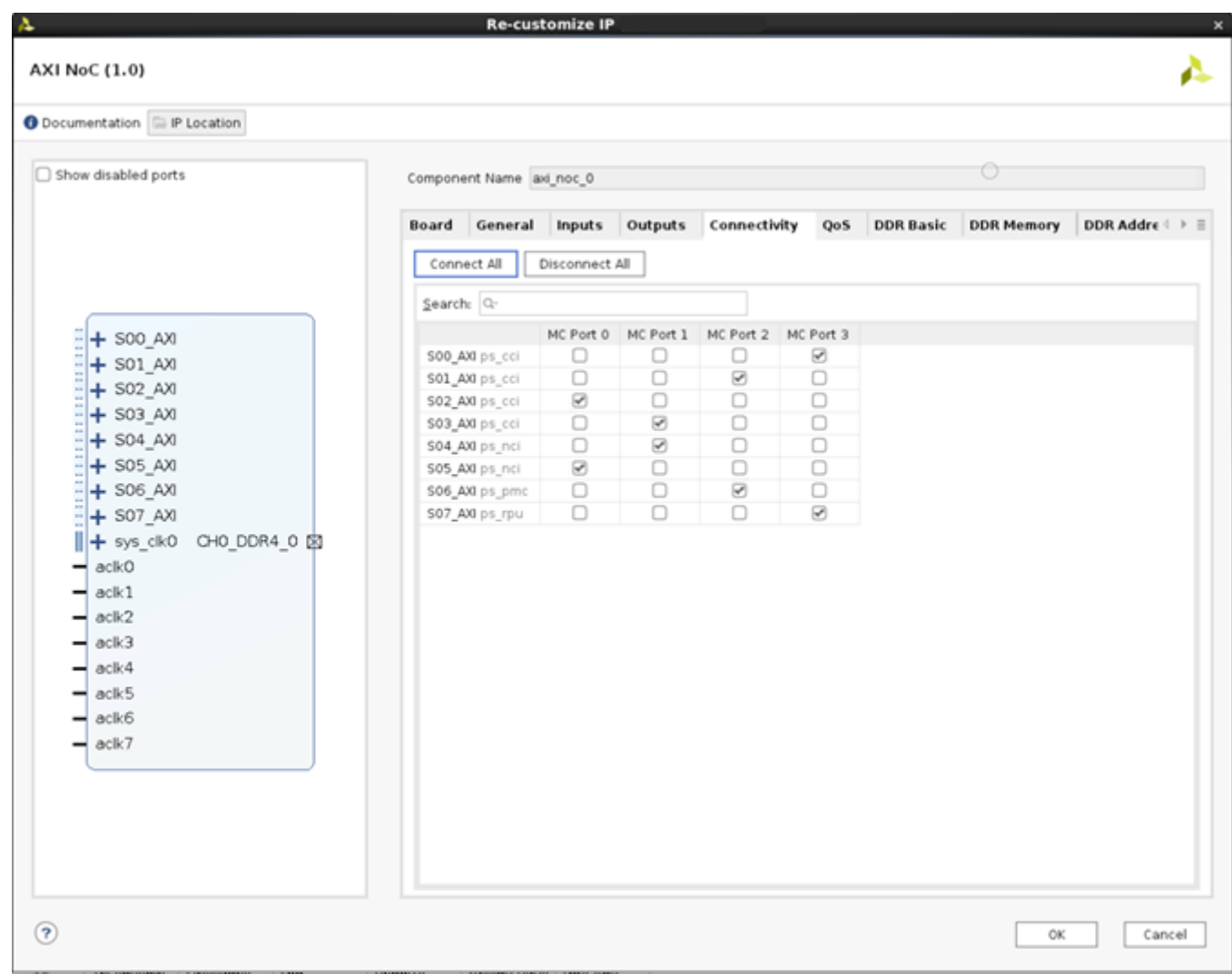
NoC interfaces details



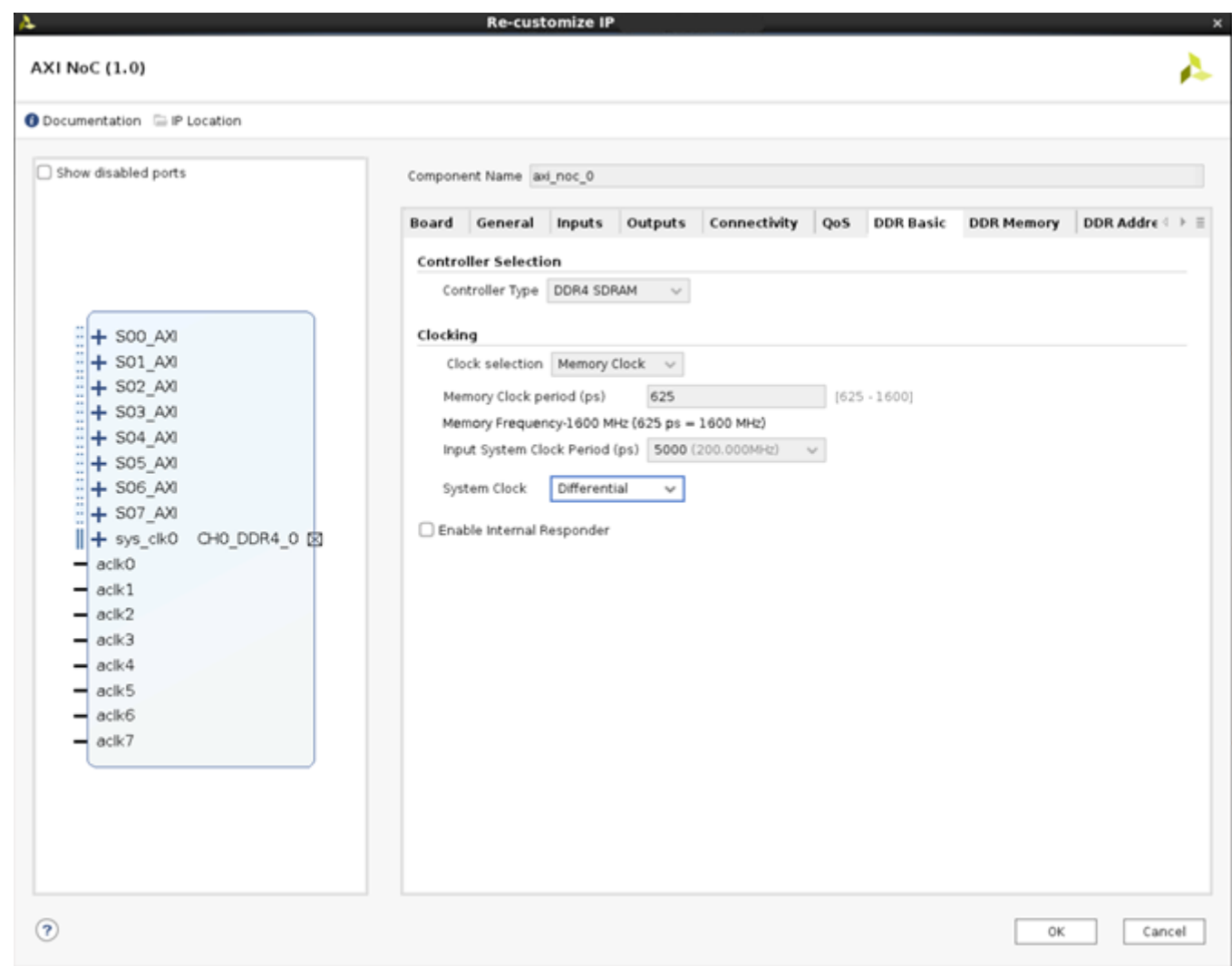
NoC inputs



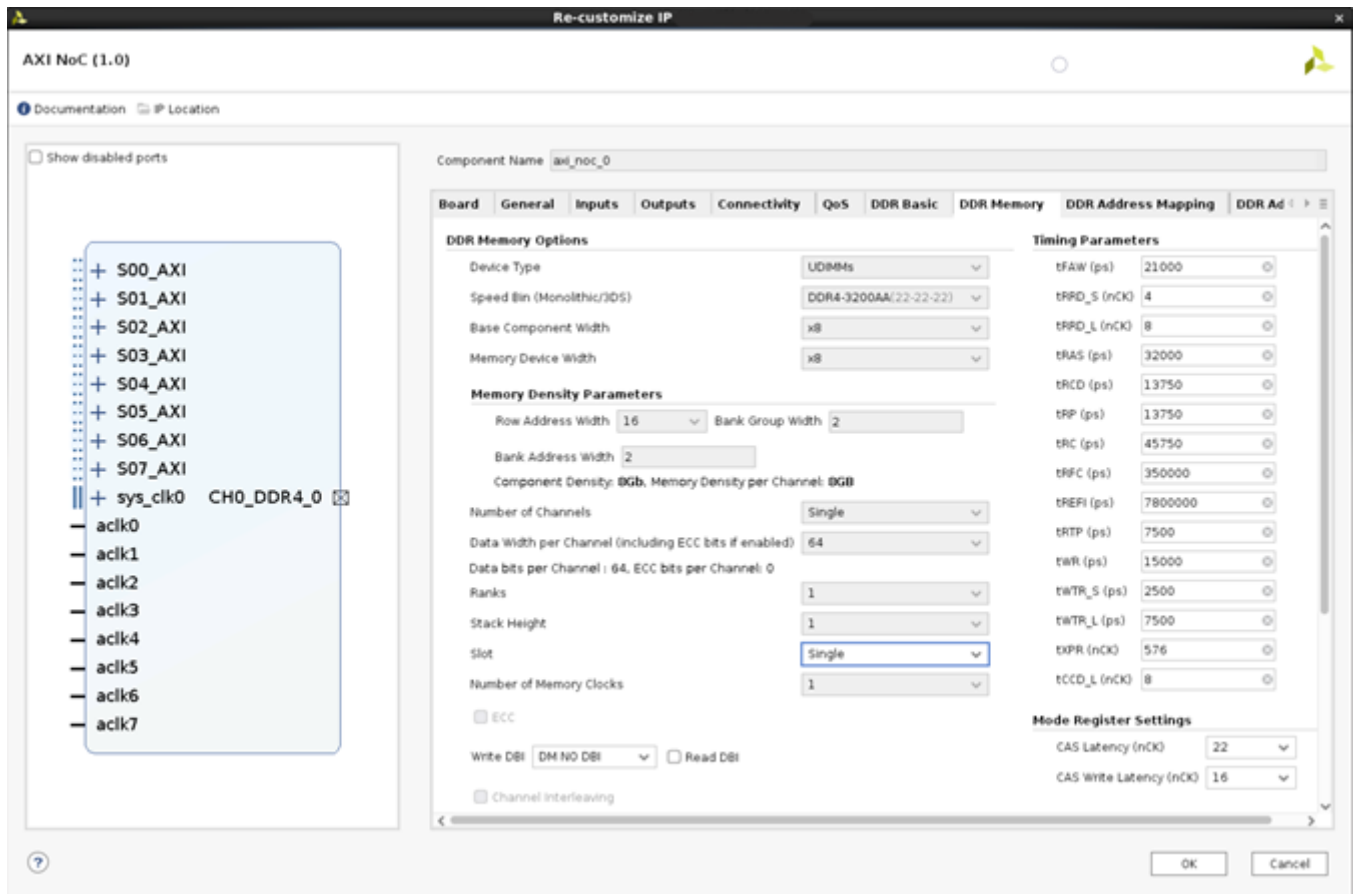
NoC port connectivity



DDR configurations



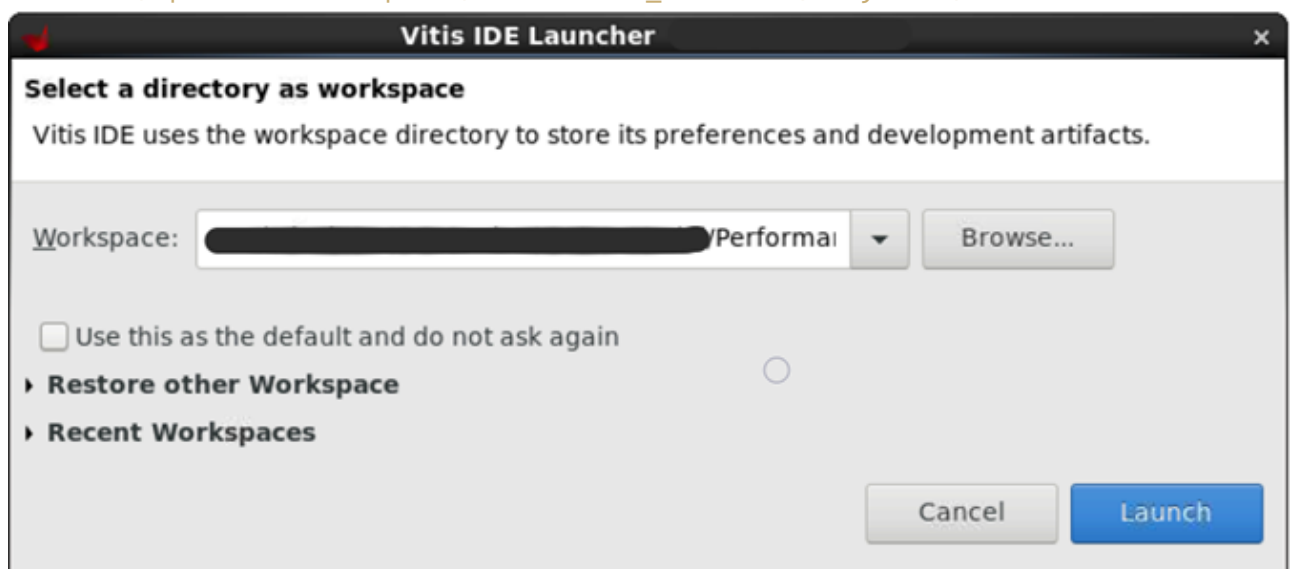
DDR memory options



Create a New Application Project for Dhrystone

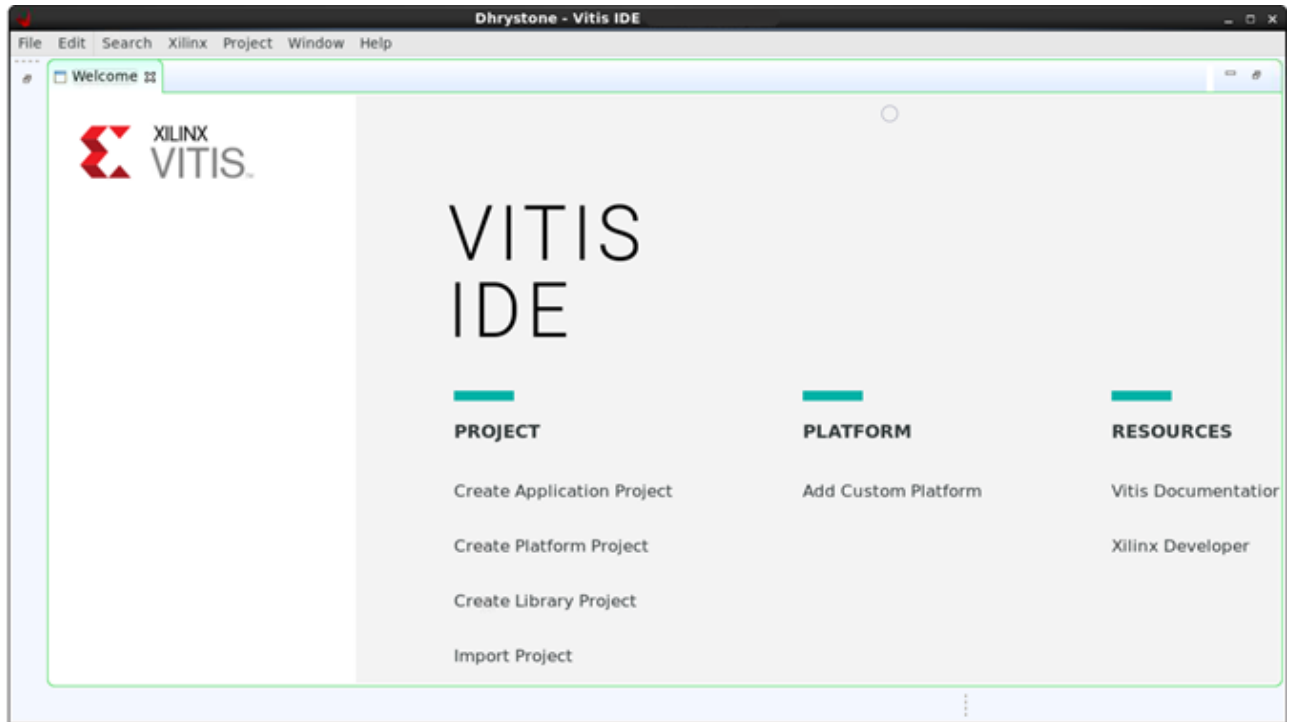
Step 1: Create and Browse to the Workspace

1. Create a workspace and launch the Vitis tool.
2. Browse to the workspace.
3. Click **Launch** to open the VITIS IDE wizard.
4. Browse to `/<path for workspace>/Performance_Benchmark/Dhrystone/`

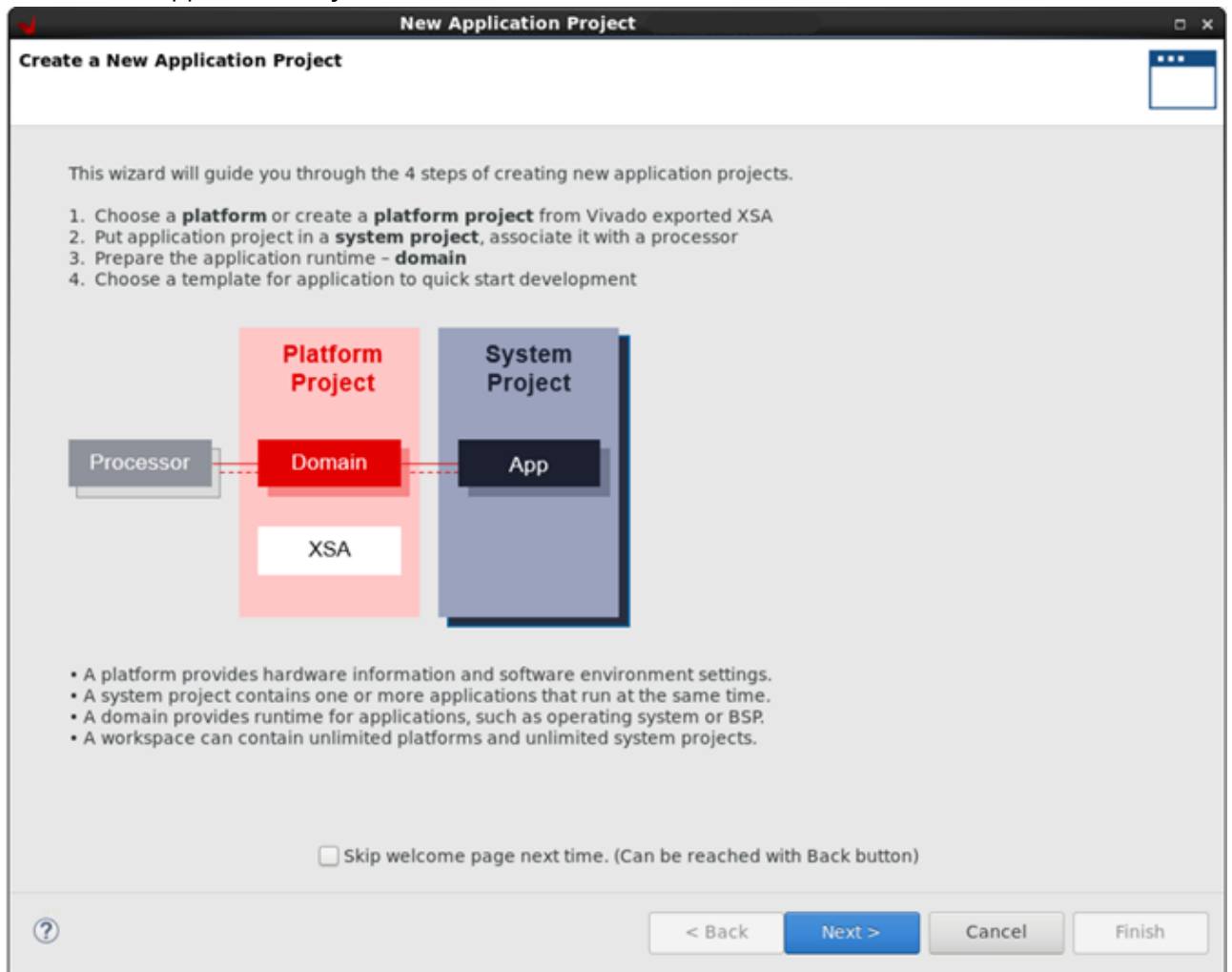


Step 2: Create the Application Project

1. Select **File->New->Application Project** from the menu. Alternatively, you can create the project by clicking **Create Application Project**.

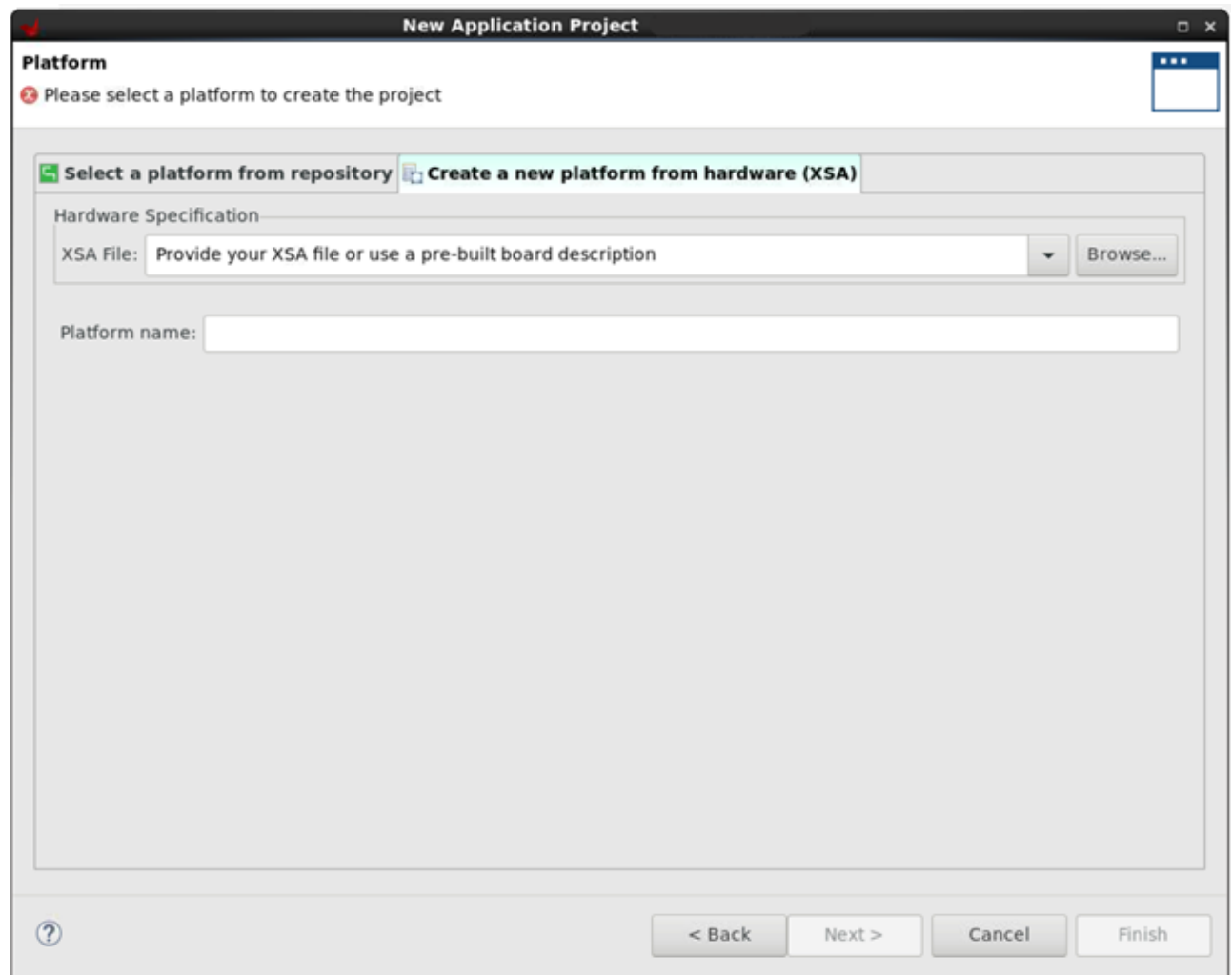


2. On the New Application Project wizard, click **Next**.



Step 3: Add Hardware Description File

1. To add the hardware description file (XSA), select **Create a new platform from hardware (XSA)**.



2. Browse to the XSA file and click **Next** to open the New Application Project wizard.

```
/<path for  
workspace>/Performance_Benchmark/Dhrystone/Design/dhrystone_tutorial.xsa
```

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: Performance_Benchmark/Dhrystone/Design/d Browse...

Platform name: dhrystone_tutorial

< Back Next > Cancel Finish

Step 4: Create Dhrystone application

Set Domain Configuration

1. For the **Application project name**, enter **Dhrystone**.

2. Select the **psv_cortexa72_0** processor, and click **Next**.

New Application Project

Application Project Details

Specify the application project name and its system project properties

Application project name: Dhrystone_Benchmark

System Project

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

Select a system project

+ Create new...

System project details

System project name: Dhrystone_Benchmark_system

Target processor

Select target processor for the Application project.

Processor	Associated applications
versal_cips_0_pspmc_0_psv_cortexa72_0	Dhrystone_Benchmark
versal_cips_0_pspmc_0_psv_cortexa72_1	
versal_cips_0_pspmc_0_psv_cortexr5_0	
versal_cips_0_pspmc_0_psv_cortexr5_1	
versal_cips_0_pspmc_0_psv_pmc_0	
versal_cips_0_pspmc_0_psv_psm_0	
psv_cortexa72 SMP	

Show all processors in the hardware specification

< Back

Next >

Cancel

Finish

3. Select the domain details and then click **Next**.

The screenshot shows the 'New Application Project' wizard, specifically the 'Domain' step. The window title is 'New Application Project'. The main heading is 'Domain', with the instruction 'Select a domain for your project or create a new domain'. Below this, it says 'Select the domain that the application would link to or create a new domain' and a note: 'Note: New domain created by this wizard will have all the requirements of the application template selected in the next step'.

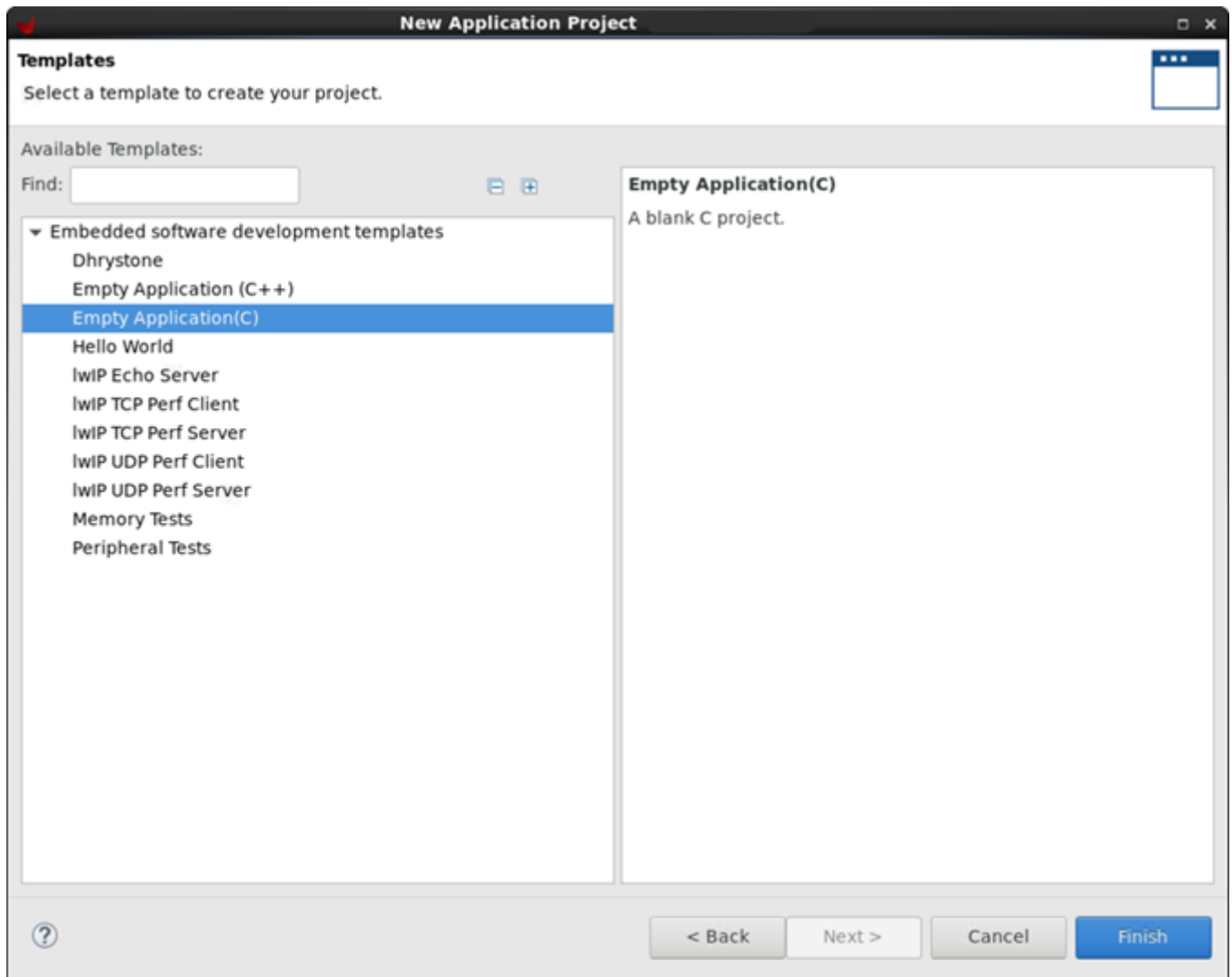
On the left, there is a list box titled 'Select a domain' with a single item: '+ Create new...'. On the right, under 'Domain details', there are five fields:

- Name: standalone_psv_cortexa72_0
- Display Name: standalone_psv_cortexa72_0
- Operating System: standalone (dropdown menu)
- Processor: versal_cips_0_pspmc_0_psv_cortexa72_0
- Architecture: 64-bit (dropdown menu)

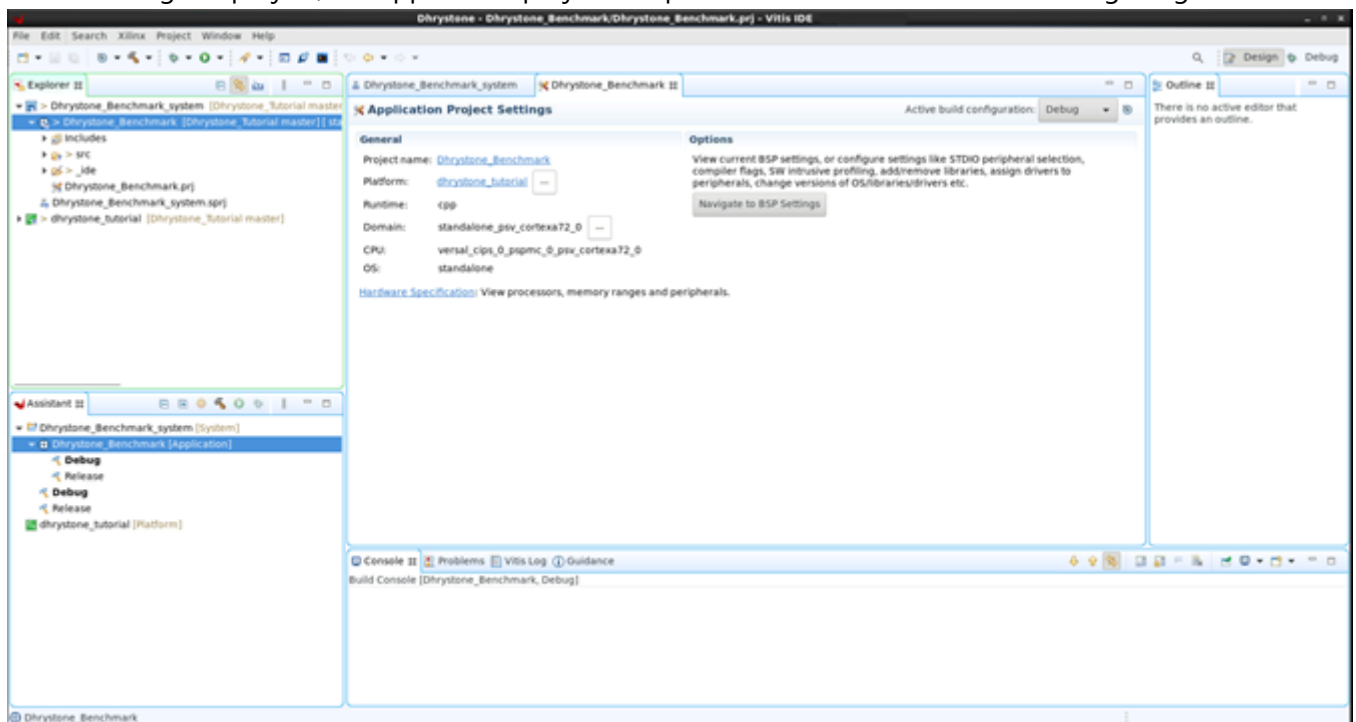
At the bottom, there are four buttons: '?', '< Back', 'Next >' (highlighted in blue), 'Cancel', and 'Finish'.

Create Empty Application Template

1. Select the Empty Application template (blank c project) for Dhrystone application.
2. Click **Finish**.

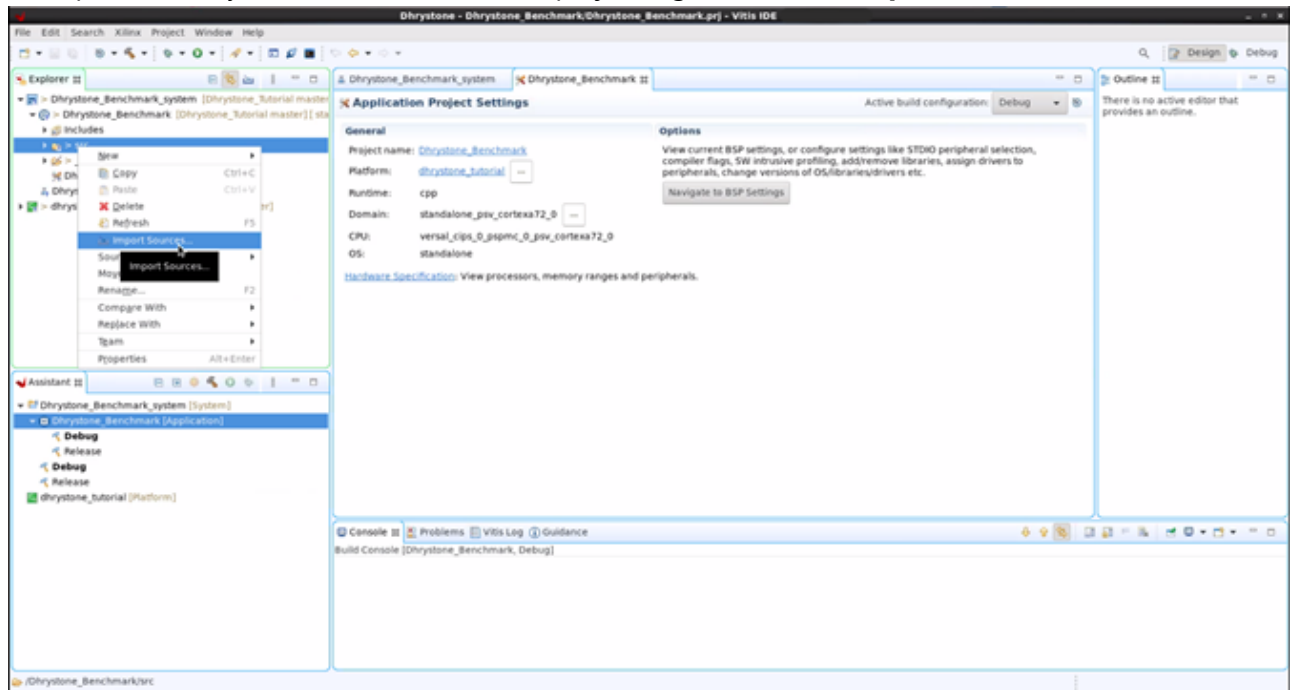


After creating the project, the application project template looks as shown in the following image.



Import Dhrystone Source Code

1. To import the Dhrystone source code to the project, right-click **src**->**import sources**.



2. Browse to the source path `/<Path to workspace>/Performance_Benchmark/Dhrystone/Source_code/`
3. Select all the source files and click **Finish** to import.
4. Click **Yes to All**.

Import Sources

File system

Import resources from the local file system.

From directory:

/Performance

Browse...

☒

Source_code

☒ dhry_1.c

☒ dhry_2.c

☒ dhry.h

☒ LICENSE

☒ README.md

Filter Types...

Select All

Deselect All

Into folder:

Dhrystone_Benchmark/src

Browse...

Options

☐ Overwrite existing resources without warning

☐ Create top-level folder

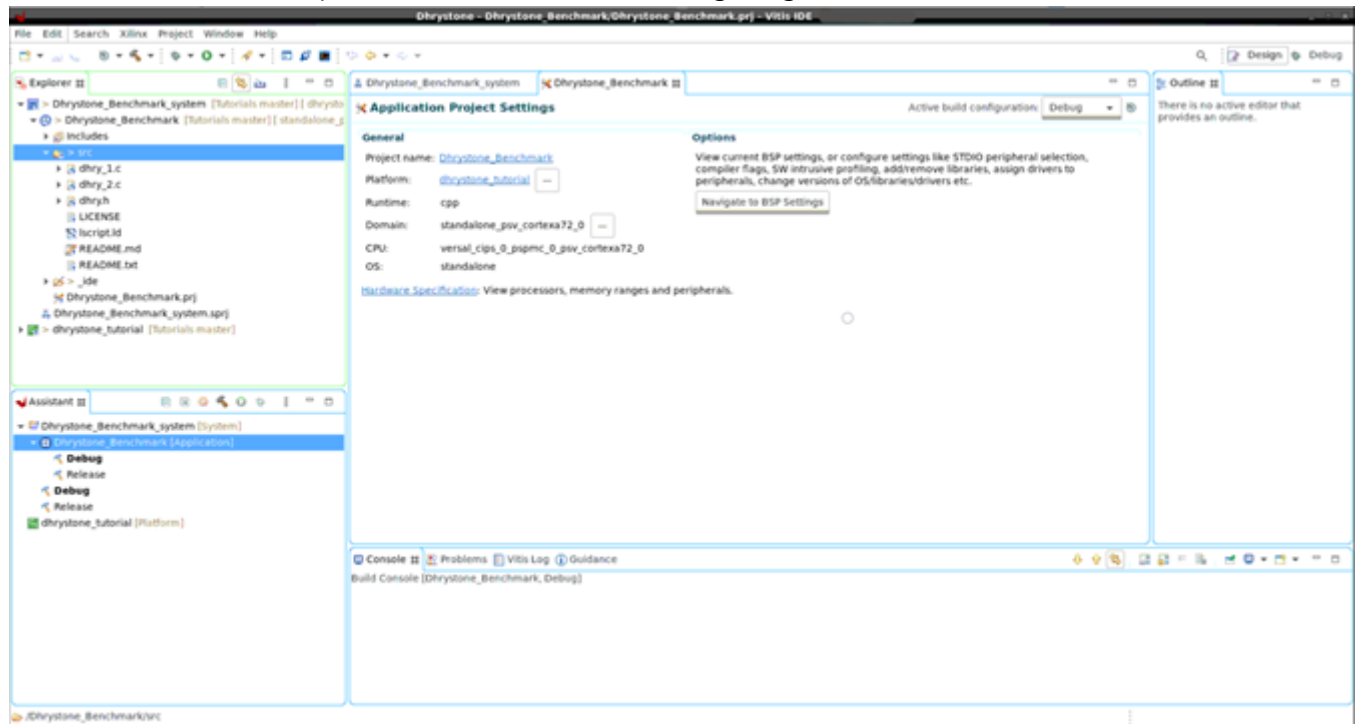
Advanced >>

?

Cancel

Finish

All the source files are imported, as shown in the following image.

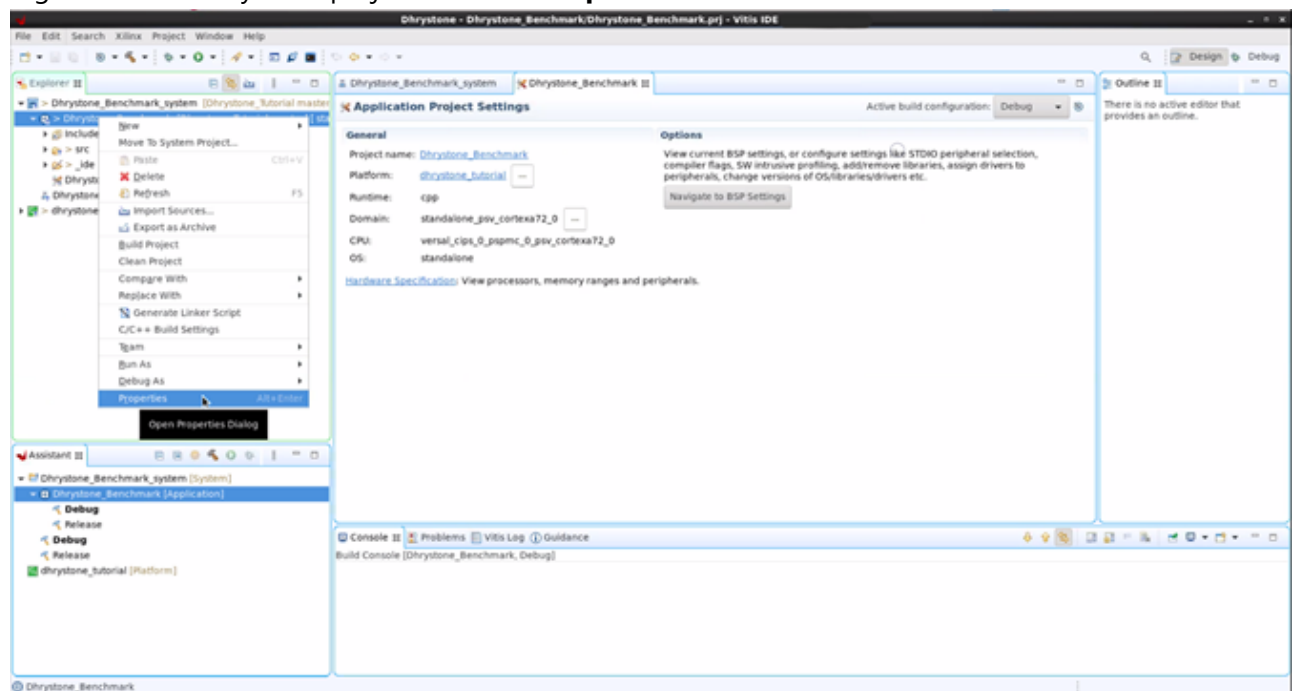


Build Dhrystone application

Optimization Level

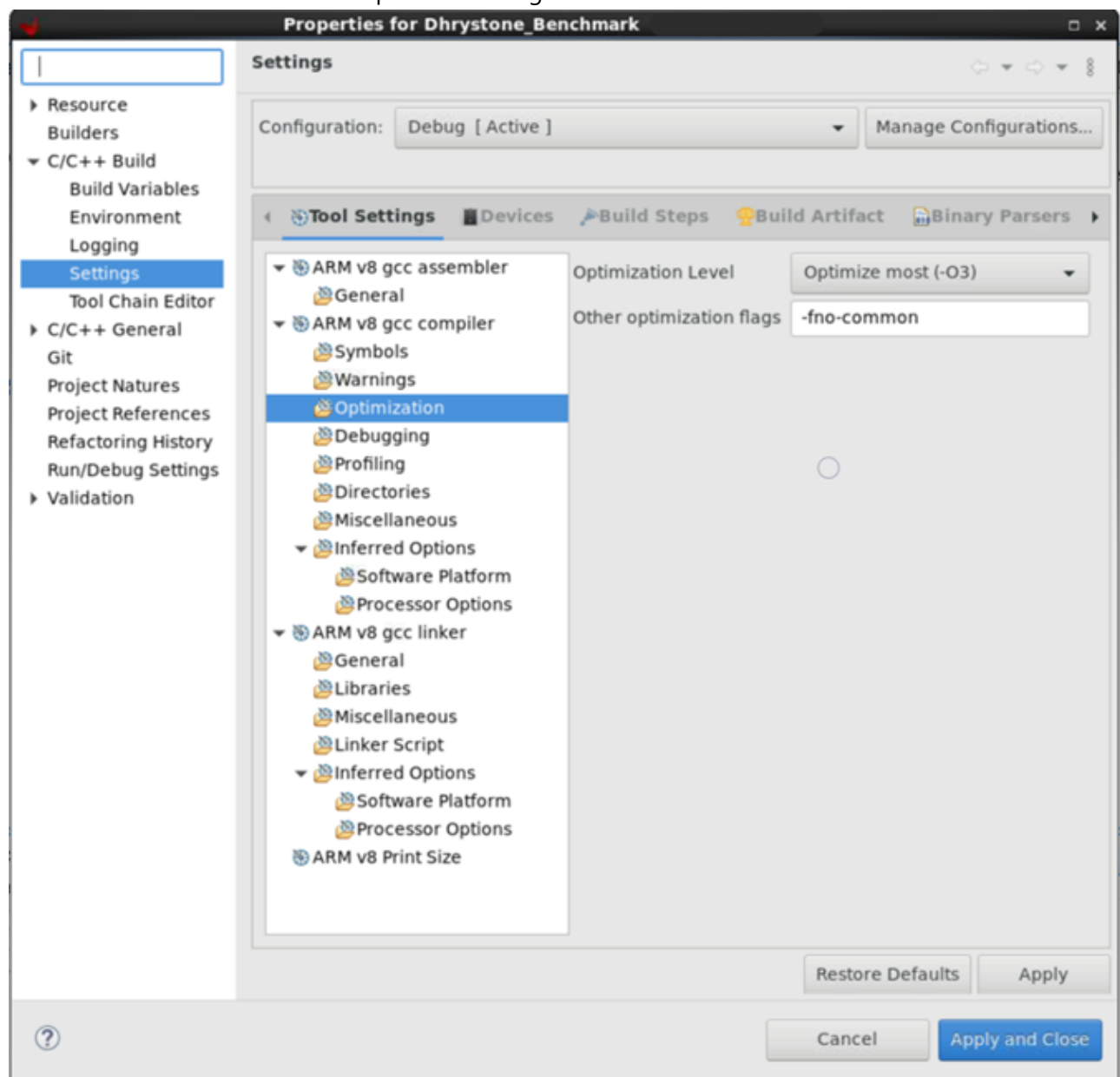
Note: Add the optimization Level **-O3** and **-fno-common** for better performance.

1. Right-click the Dhrystone project and click **Properties**.



2. Expand **C/C++ Build** and click **Settings**.
3. Click **Optimization**, and then select the Optimization Level **-O3**.

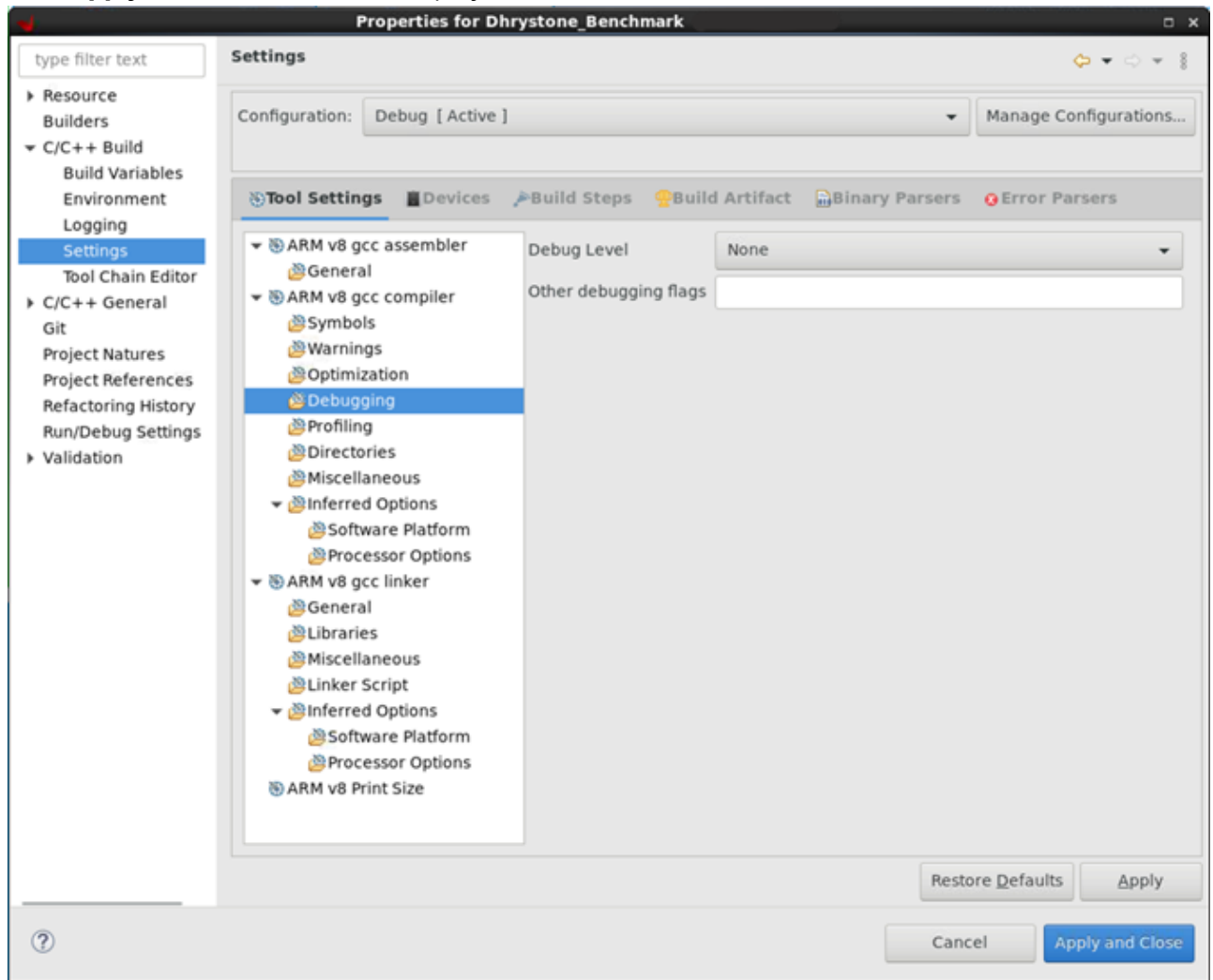
4. Add **-fno-common** in the Other optimization flags.



Debug Level

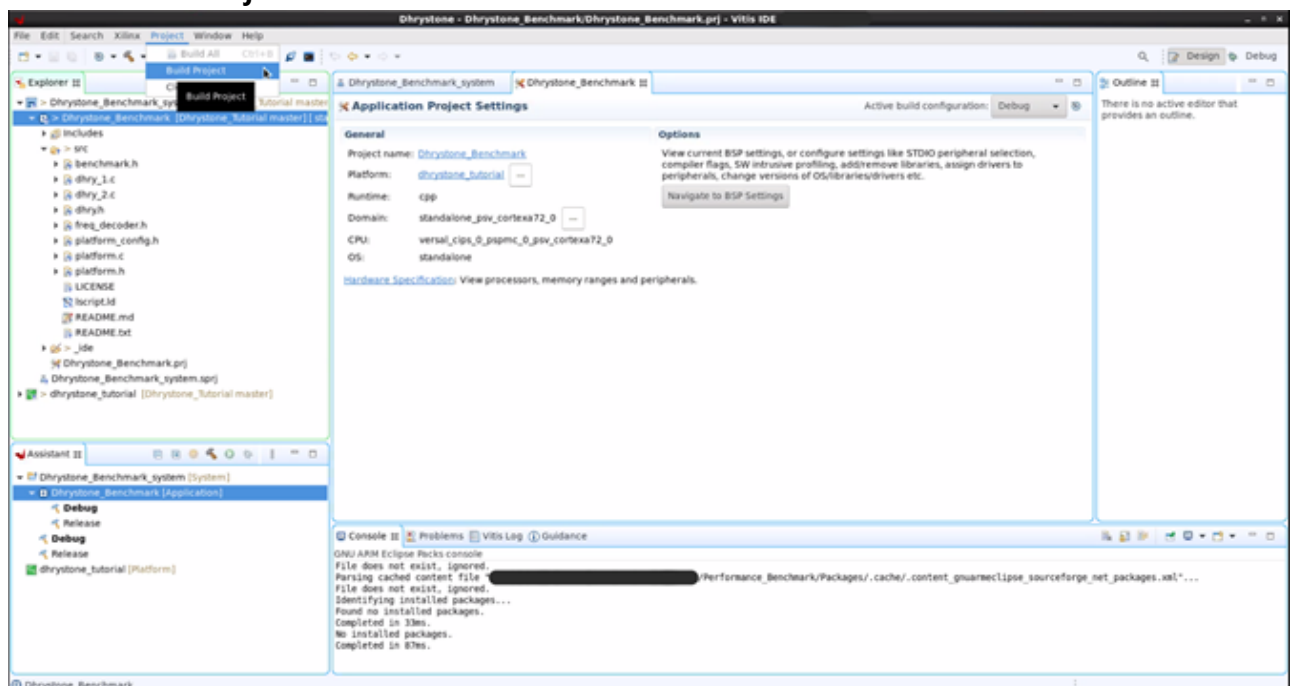
1. Navigate to **Properties->C/C++ Build->Settings->Debugging**.
2. Select **Debug Level** as **None**.

- Click **Apply** and **Close** and build the project.

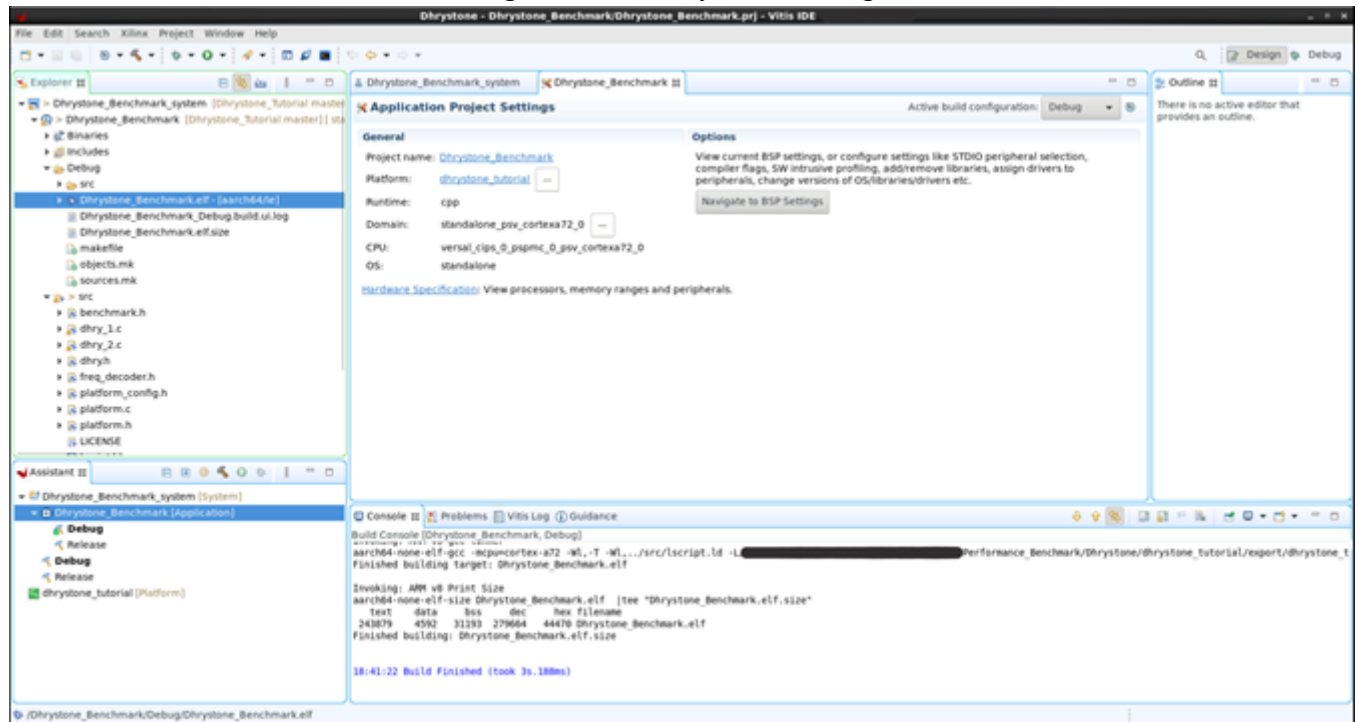


Build the Project

- Right-click the project name, and then click **Build Project**. Alternatively, navigate to the Project menu and click **Build Project**.



After the build is over, the executable is generated. Verify the build logs in the console.



For executable file path:

```
/<path for
workspace>/Performance_Benchmark/Dhrystone/Dhrystone_Benchmark/Debug/Dhrystone_Benchmark.elf
```

Run the Dhrystone Application

1. Insert the SD card with the system controller image into the System controller boot mode to SD1 (SW11 = 0111).
2. On the host machine, connect the USB Type-C cable into the VCK190 Board USB Type-C port (J207) and the other end into an open USB port.

3. Configure the board to boot in JTAG mode by setting switch SW1 = 0000 as shown in the following

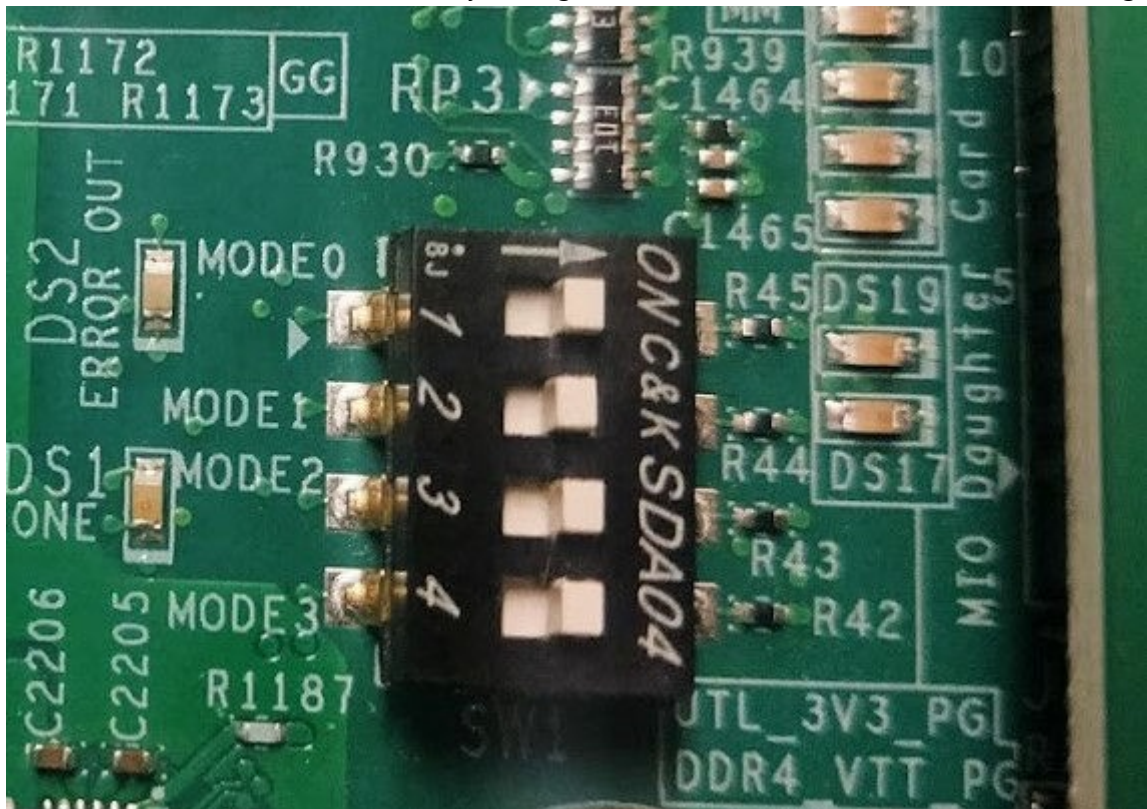
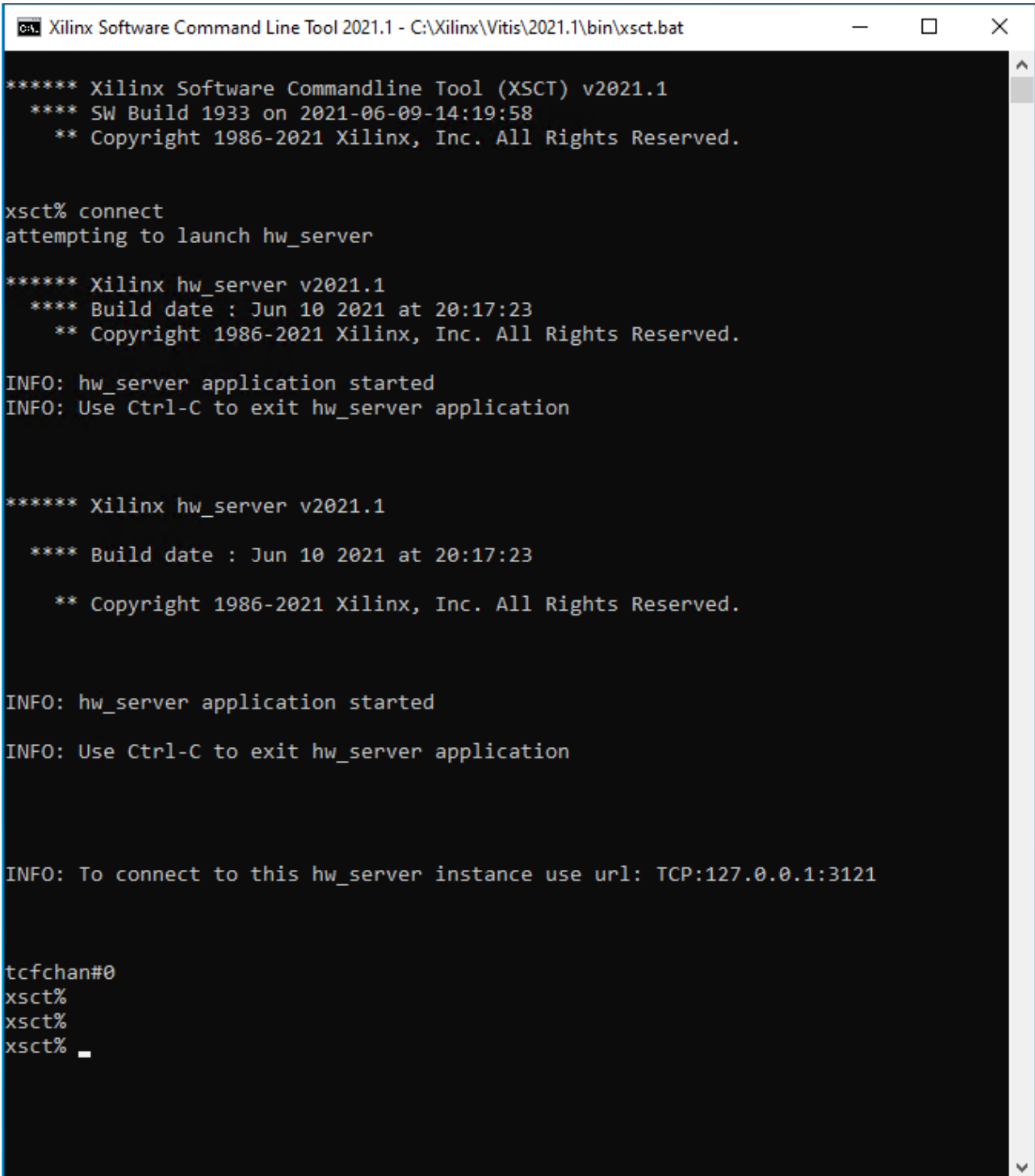


image.

4. Connect 180W(12V) power to the VCK190 6-Pin Molex connector(J16).
5. Power on the VCK190 board using the power switch (SW13).
6. Open the serial port in Tera Term/Putty and set baud rate(115200) for logs.
7. Go to Vitis command prompt, run `xsd` or `xsc` commands. **Note:** Refer Vivado/Vitis installation paths for this tools.

8. Run the `connect` command to launch `hw_server`.



```
OS% Xilinx Software Command Line Tool 2021.1 - C:\Xilinx\Vitis\2021.1\bin\xsct.bat

***** Xilinx Software Commandline Tool (XSCT) v2021.1
**** SW Build 1933 on 2021-06-09-14:19:58
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

xsct% connect
attempting to launch hw_server

***** Xilinx hw_server v2021.1
**** Build date : Jun 10 2021 at 20:17:23
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

INFO: hw_server application started
INFO: Use Ctrl-C to exit hw_server application

***** Xilinx hw_server v2021.1

**** Build date : Jun 10 2021 at 20:17:23

** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

INFO: hw_server application started
INFO: Use Ctrl-C to exit hw_server application

INFO: To connect to this hw_server instance use url: TCP:127.0.0.1:3121

tcfchan#0
xsct%
xsct%
xsct% _
```

9. List the targets by running the `targets` command.

```

Xilinx Software Command Line Tool 2021.1 - C:\Xilinx\Vitis\2021.1\bin\xsct.bat
xsct%
xsct% targets
 1 Versal xcvc1902
 2 RPU (PS POR is active)
 3 Cortex-R5 #0 (PS POR is active)
 4 Cortex-R5 #1 (PS POR is active)
 5 APU (FPD domain isolation)
 6 Cortex-A72 #0 (FPD domain isolation)
 7 Cortex-A72 #1 (FPD domain isolation)
 8 PPU
 9 MicroBlaze PPU (Sleeping after reset)
10 PSM
11 PMC
12 PL
13 DPC
xsct%
xsct%

```

10. Program the design.

```

xsct% device program /<path for
workspace>/Performance_Benchmark/Dhrystone/Design/runs/dperf_<*>/dhrystone_tutorial
1.runs/impl_1/dhrystone_perf_wrapper.pdi

```

```

Xilinx Software Command Line Tool 2021.1 - C:\Xilinx\Vitis\2021.1\bin\xsct.bat
xsct%
xsct% targets
 1 Versal xcvc1902
 2 RPU (PS POR is active)
 3 Cortex-R5 #0 (PS POR is active)
 4 Cortex-R5 #1 (PS POR is active)
 5 APU (FPD domain isolation)
 6 Cortex-A72 #0 (FPD domain isolation)
 7 Cortex-A72 #1 (FPD domain isolation)
 8 PPU
 9 MicroBlaze PPU (Sleeping after reset)
10 PSM
11 PMC
12 PL
13 DPC
xsct%
xsct% device program /Performance_Benchmark/Dhrystone/Design/runs/dperf_1624360968/dhrystone_tutorial.runs/impl_1/dhrystone_perf_wrapper.pdi
100% 1PB 1.0PB/s 00:01
xsct%
COMS - Tera Term VT
File Edit Setup Control Window Help
[4.036]PLM Initialization Time
[4.099]*****Boot PDI Load: Started*****
[4.177]Loading PDI from JTAG
[4.238]Monolithic Master Device
[4.356]10.146 ms PDI initialization time
[4.438]***Loading Image# 0x1, Name: lpd, Id: 0x04210002
[4.515]---Loading Partition# 0x1, Id: 0x0
[27.815]*****
[32.862]Xilinx Versal Platform Loader and Manager
[36.459]Release 2021.2 Jun 22 2021 - 11:50:05
[40.782]Platform Version: v2.0 PRC: v2.0, PS: v2.0
[45.189]BOOTMODE: 0x0, PDI1BOOT: 0x0
[48.495]*****
[52.774] 40.149 ms for Partition# 0x1, Size: 2320 Bytes
[57.649]---Loading Partition# 0x2, Id: 0x0
[62.985] 0.589 ms for Partition# 0x2, Size: 40 Bytes
[66.124]---Loading Partition# 0x3, Id: 0x0
[134.746] 64.772 ms for Partition# 0x3, Size: 59296 Bytes
[137.266]---Loading Partition# 0x4, Id: 0x0
[148.738] 2.579 ms for Partition# 0x4, Size: 5968 Bytes
[150.928]---Loading Partition# 0x5, Id: 0x0
[154.866] 0.987 ms for Partition# 0x5, Size: 80 Bytes
[159.635]***Loading Image# 0x2, Name: pl_sfi, Id: 0x18700000
[164.913]---Loading Partition# 0x6, Id: 0x3
[1827.238] 658.389 ms for Partition# 0x6, Size: 707056 Bytes
[1829.727]---Loading Partition# 0x7, Id: 0x5
[1255.040] 421.373 ms for Partition# 0x7, Size: 431200 Bytes
[1257.657]***Loading Image# 0x3, Name: fpd, Id: 0x0420C003
[1262.798]---Loading Partition# 0x8, Id: 0x8
[1267.225] 0.417 ms for Partition# 0x8, Size: 1024 Bytes
[1271.846]*****Boot PDI Load: Done*****
[1276.321]45821.087 ms: PGM Time
[1279.245]Total PGM Boot Time

```

11. Select the A72_0 target and reset.

```

xsct% rst -processor -skip-activate-subsystem

```

```

Xilinx Software Command Line Tool 2021.1 - C:\Xilinx\Vitis\2021.1\bin\xsct.bat
xsct%
xsct%
xsct% targets
 1 Versal xcvc1902
 2 RPU (Reset)
 3 Cortex-R5 #0 (RPU PGE Reset)
 4 Cortex-R5 #1 (RPU PGE Reset)
 5 APU
 6* Cortex-A72 #0 (Power On Reset)
 7 Cortex-A72 #1 (Power On Reset)
 8 PPU
 9 MicroBlaze PPU (Sleeping)
10 PSM
11 PMC
12 PL
13 DPC
xsct%
xsct% target -set -filter {name =~ "*A72*#0"}
xsct%
xsct% rst -processor -skip-activate-subsystem
Info: Cortex-A72 #0 (target 6) Stopped at 0xffff0000 (Reset Catch)
xsct%
xsct% targets
 1 Versal xcvc1902
 2 RPU (Reset)
 3 Cortex-R5 #0 (RPU PGE Reset)
 4 Cortex-R5 #1 (RPU PGE Reset)
 5 APU
 6* Cortex-A72 #0 (Reset Catch, EL3(S)/A64)
 7 Cortex-A72 #1 (Power On Reset)
 8 PPU
 9 MicroBlaze PPU (Sleeping)
10 PSM
11 PMC
12 PL
13 DPC
xsct%
xsct%

```

12. Download and run the Dhrystone benchmark application. Before executing Dhrystone Benchmark, see

4 Running Dhrystone section of the Dhrystone Benchmarking for ARM Cortex Processors -

<https://developer.arm.com/documentation/dai0273/a/>

```

xsct% dow -force /<path for
workspace>/Performance_Benchmark/Dhrystone/Dhrystone_Benchmark/Debug/Dhrystone_Ben
chmark.elf
xsct% con

```

```

Xilinx Software Command Line Tool 2021.1 - C:\Xilinx\Win2021\Fabric\asc.bat
asc% dow -force [redacted] /Performance_Benchmark/Dhrystone/Dhrystone_Benchmark/Debug/Dhrystone_Benchmark.elf
Downloading Program -- [redacted] /Performance_Benchmark/Dhrystone/Dhrystone_Benchmark/Debug/Dhrystone_Benchmark.elf
section, .text: 0x00000000 - 0x0001567f
section, .init: 0x00015680 - 0x000156b3
section, .fini: 0x000156c0 - 0x000156f3
section, .note.gnu.build-id: 0x000156f4 - 0x00015717
section, .rodata: 0x00015720 - 0x000167b6
section, .rodata1: 0x000167b7 - 0x000167bf
section, .sdata2: 0x000167c0 - 0x000167bf
section, .sbss2: 0x000167c8 - 0x000167bf
section, .data: 0x000167c0 - 0x0001796f
section, .data1: 0x00017968 - 0x0001797f
section, .ctors: 0x00017980 - 0x0001797f
section, .dtors: 0x00017980 - 0x0001797f
section, .eh_frame: 0x00017980 - 0x000179e3
section, .mmu_tbi0: 0x00018000 - 0x000180ff
section, .mmu_tbi1: 0x00019000 - 0x00019fff
section, .mmu_tbi2: 0x00019000 - 0x00019fff
section, .preinit_array: 0x0001e000 - 0x0001dfff
section, .init_array: 0x0001e000 - 0x0001e007
section, .fini_array: 0x0001e008 - 0x0001e047
section, .sdata: 0x0001e048 - 0x0001e07f
section, .sbss: 0x0001e080 - 0x0001e07f
section, .tdata: 0x0001e080 - 0x0001e07f
section, .tbss: 0x0001e080 - 0x0001e07f
section, .bss: 0x0001e080 - 0x000429ff
section, .heap: 0x00042a00 - 0x000429ff
section, .stack: 0x00042a00 - 0x000459ff
100% 0MB 0.2MB/s 00:01
Setting PC to Program Start Address 0x00000000
Successfully downloaded [redacted] /Performance_Benchmark/Dhrystone/Dhrystone_Benchmark/Debug/Dhrystone_Benchmark.elf
asc% con
Info: Cortex-A72 #0 (target 6) Running
asc%
asc%
asc%
asc%
asc%
asc%

```

```

C:\Xilinx\Win2021\Fabric\asc.bat
File Edit Setup Control Window Help
(1284.180)Total FLM Boot Time
Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Please give the number of runs through the benchmark:
Execution starts, 4000000 runs through Dhrystone
Execution ends
Final values of the variables used in the benchmark:
Int_Glob: should be: 5
Bool_Glob: should be: 1
Ch_1_Glob: should be: 1
Ch_2_Glob: should be: 8
Arr_1_Glob: should be: 7
Arr_2_Glob: should be: 7
Ptr_Glob: should be: Number_Of_Runs + 10
Ptr_Comp: 37748736
Discr: should be: (implementation-dependent)
Even_Comp: should be: 2
Int_Comp: should be: 17
Str_Comp: should be: DHRYSTONE PROGRAM, SOME STRING
Next_Ptr_Glob: 37748736
Ptr_Comp: should be: (implementation-dependent), same as above
Discr: should be: 0
Even_Comp: should be: 1
Int_Comp: should be: 18
Str_Comp: should be: DHRYSTONE PROGRAM, SOME STRING
Int_1_Loc: should be: 5
Int_2_Loc: should be: 13
Int_3_Loc: should be: 13
Even_Loc: should be: 1
Str_1_Loc: should be: DHRYSTONE PROGRAM, 1ST STRING
Str_2_Loc: should be: DHRYSTONE PROGRAM, 2ND STRING
Str_3_Loc: should be: DHRYSTONE PROGRAM, 3RD STRING
Microseconds for one run through Dhrystone: 70000.0
Dhrystones per Second: 14285565.0

```

For performance number calculation, use the Dhrystones per second value from the last UART log print (Highlighted using a blue rectangle in the previous picture).

Performance Calculation

Calculate DMIPS (Dhrystone MIPS) number by using the following formula:

$$\begin{aligned}
 \text{DMIPS} &= \text{Dhrystones per second} / 1757 \\
 &= 14285565 / 1757 \\
 &= 8130.6573
 \end{aligned}$$

A more commonly reported figure is DMIPS / MHz, where MHz is CPU Frequency
i.e $8130.6573 / 1400 = 5.80$

Note:

1. For more details on the formula, see **5 Measurement characteristics** of Dhrystone Benchmarking for ARM Cortex Processors - <https://developer.arm.com/documentation/dai0273/a/>
2. For CPU Frequency configured in design, see the **APU clock configuration** section.

Support

GitHub issues will be used for tracking requests and bugs. For questions go to forums.xilinx.com.

License

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

XD0xx | © Copyright 2021 Xilinx, Inc.