Step1: Setup tool environment

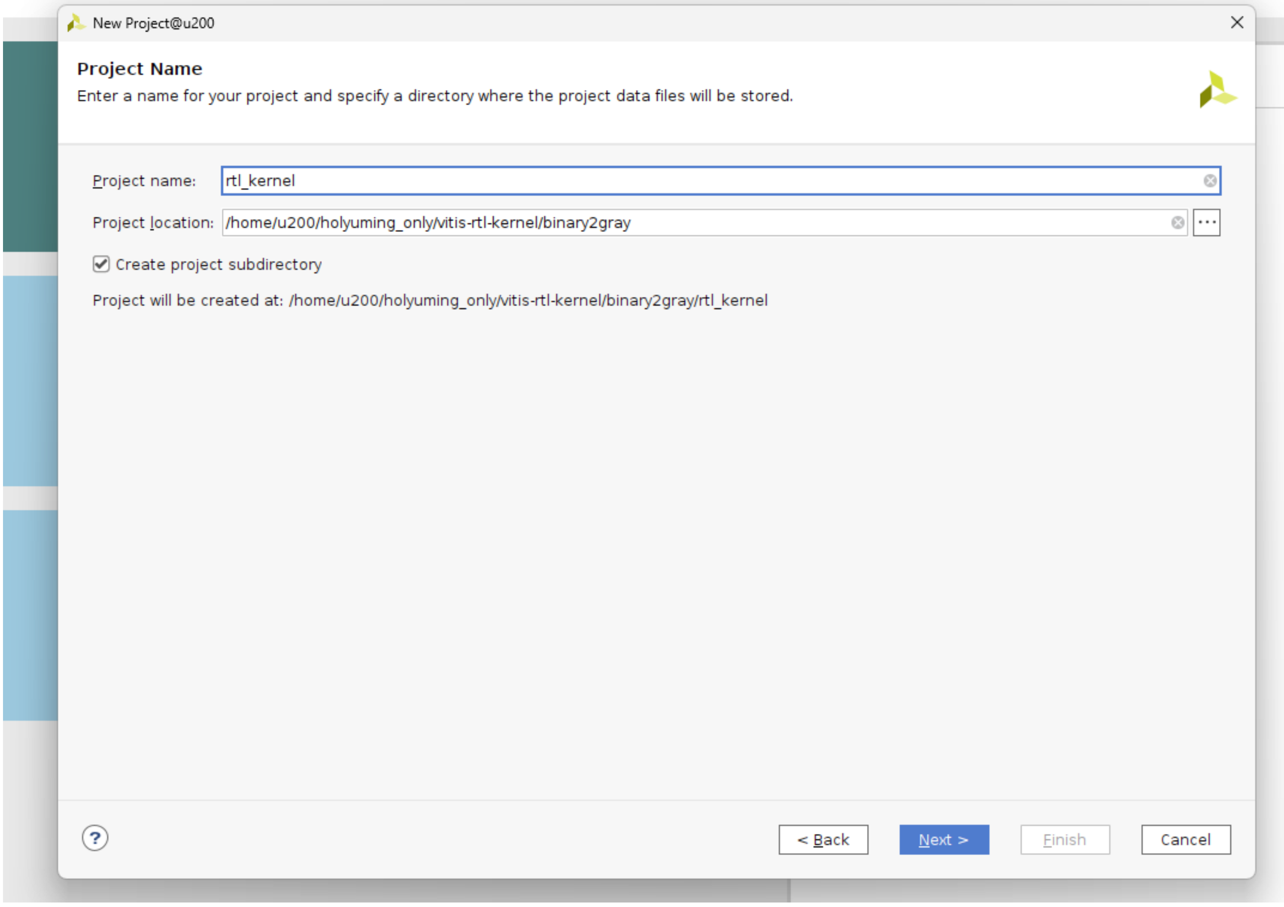
```
source /tools/Xilinx/Vitis/2022.1/settings64.sh
source /tools/Xilinx/Vitis_HLS/2022.1/settings64.sh
source /tools/Xilinx/Vivado/2022.1/settings64.sh
source /opt/xilinx/xrt/setup.sh
```

Step2: Package your rtl kernel in vivado

Open your vivado design suite to package your RTL code as an IP

```
$ vivado &
```

- Click `Create Project`, `Next`
- Edit Project name to `rtl_kernel`, and click `Next`



New Project@u200

Project Name

Enter a name for your project and specify a directory where the project data files will be stored.

Project name: rtl_kernel

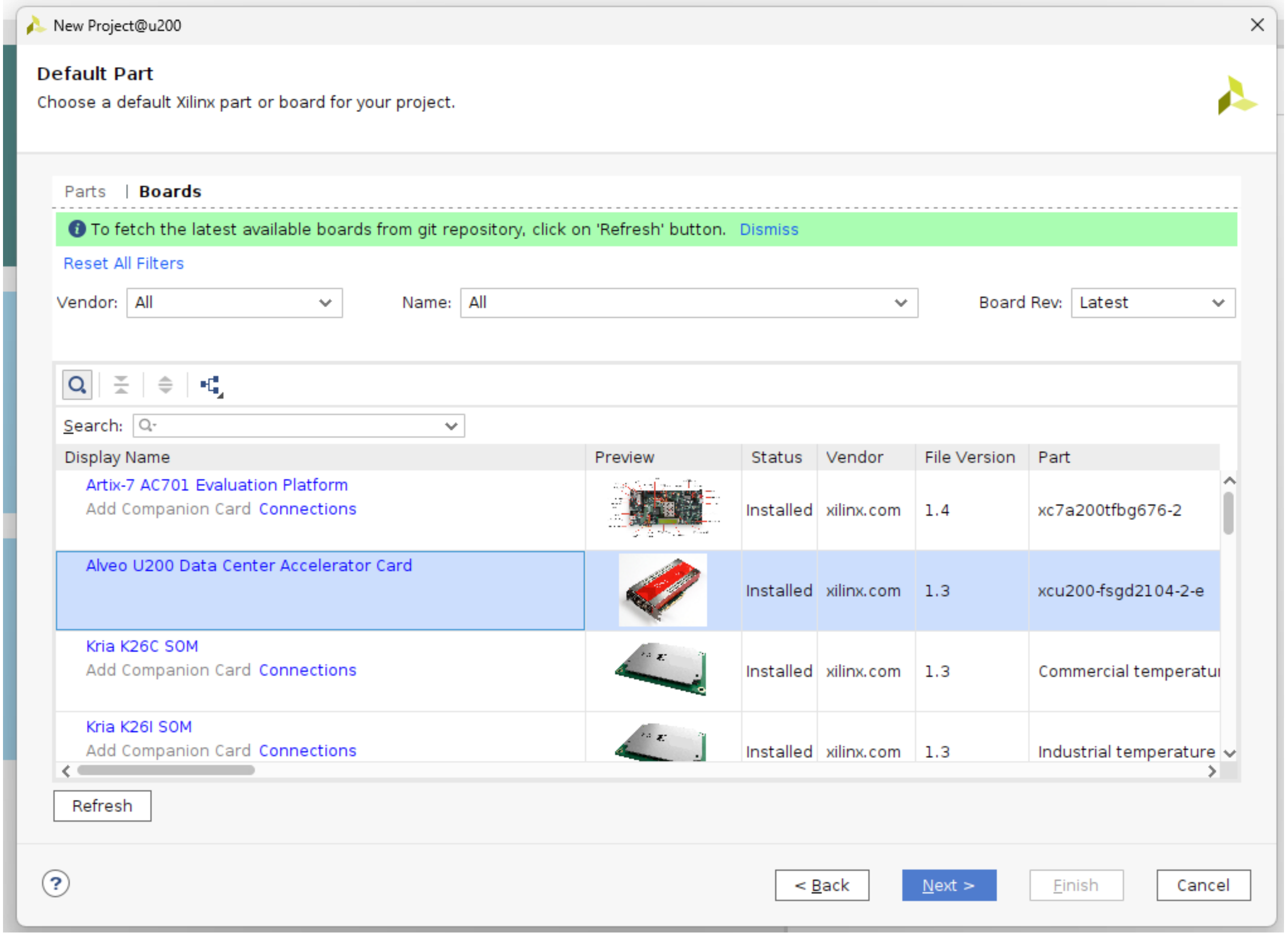
Project location: /home/u200/holyuming_only/vitis-rtl-kernel/binary2gray

☑ Create project subdirectory

Project will be created at: /home/u200/holyuming_only/vitis-rtl-kernel/binary2gray/rtl_kernel

- Enable `Do not specify sources at this time`
- Continue click `Next`

- Select `Alveo U200 Data Center Accelerator Card`, then click `Next`



New Project@u200

Default Part

Choose a default Xilinx part or board for your project.

Parts | **Boards**

To fetch the latest available boards from git repository, click on 'Refresh' button. Dismiss

Reset All Filters

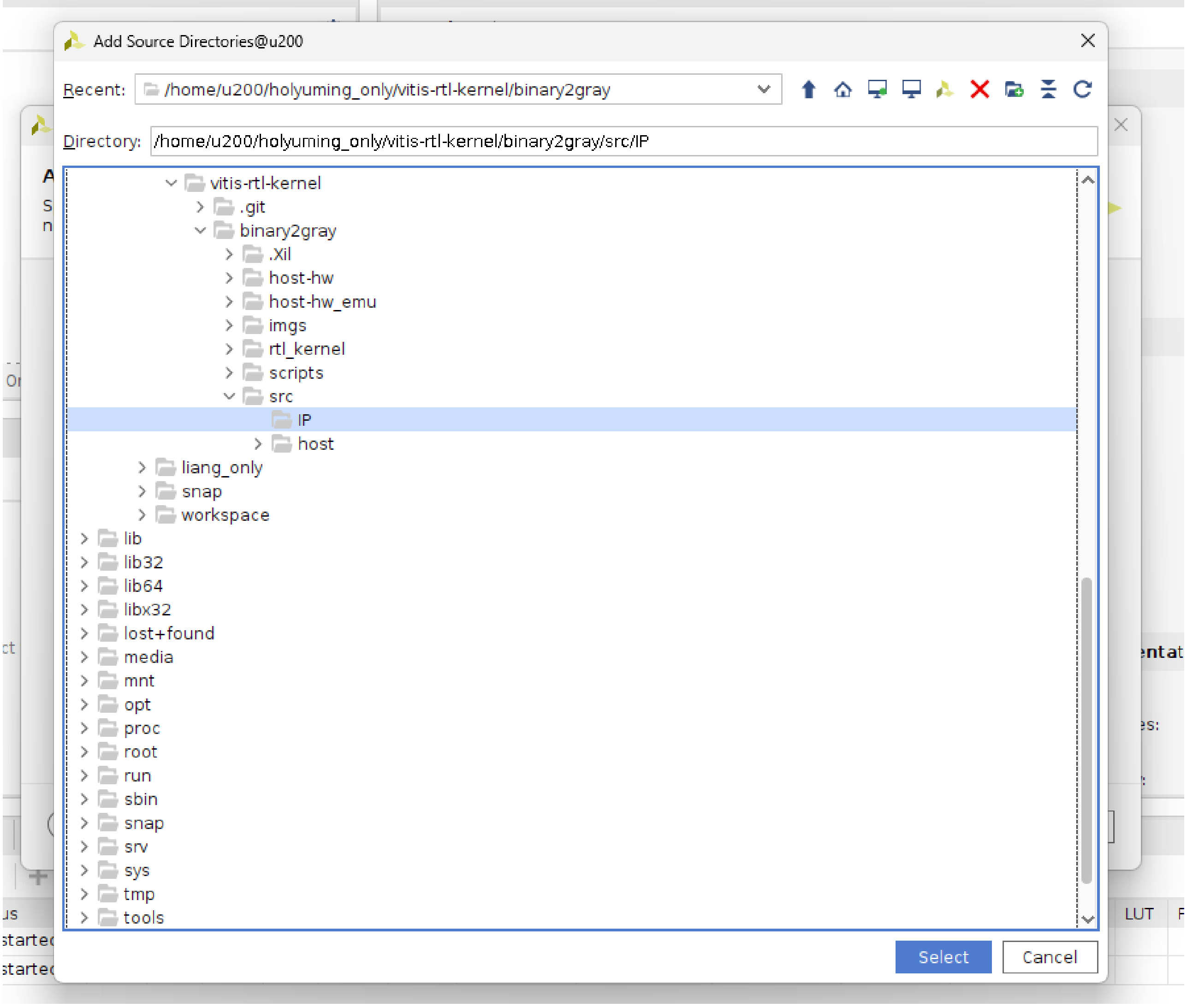
Vendor: All Name: All Board Rev: Latest

Search:

Display Name	Preview	Status	Vendor	File Version	Part
Artix-7 AC701 Evaluation Platform Add Companion Card Connections		Installed	xilinx.com	1.4	xc7a200tfbg676-2
Alveo U200 Data Center Accelerator Card		Installed	xilinx.com	1.3	xcu200-fsgd2104-2-e
Kria K26C SOM Add Companion Card Connections		Installed	xilinx.com	1.3	Commercial temperatu
Kria K26I SOM Add Companion Card Connections		Installed	xilinx.com	1.3	Industrial temperature

- Click `Finish`
- On the left, Click `Add sources` in Project Manager
- Select `Add or create design sources`, then click `Next`

- Click `Add Directories`, then click `Select`, and `Finish`

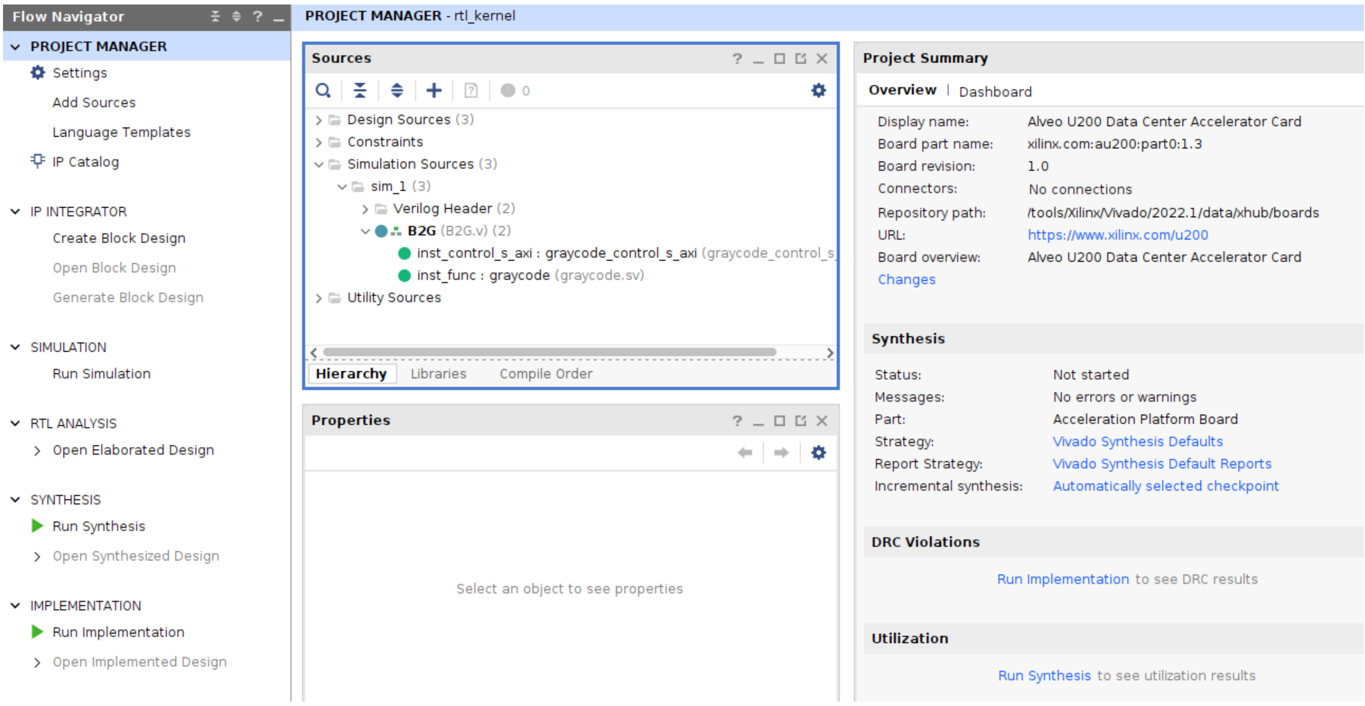


Add Source Directories@u200

Recent: /home/u200/holyuming_only/vitis-rtl-kernel/binary2gray

Directory: /home/u200/holyuming_only/vitis-rtl-kernel/binary2gray/src/IP

You can now see the design sources as following hierarchy:



Display name: Alveo U200 Data Center Accelerator Card
Board part name: xilinx.com:au200:part0:1.3
Board revision: 1.0
Connectors: No connections
Repository path: /tools/Xilinx/Vivado/2022.1/data/xhub/boards
URL: https://www.xilinx.com/u200
Board overview: Alveo U200 Data Center Accelerator Card
Changes

Synthesis

Status: Not started
Messages: No errors or warnings
Part: Acceleration Platform Board
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports
Incremental synthesis: Automatically selected checkpoint

DRC Violations

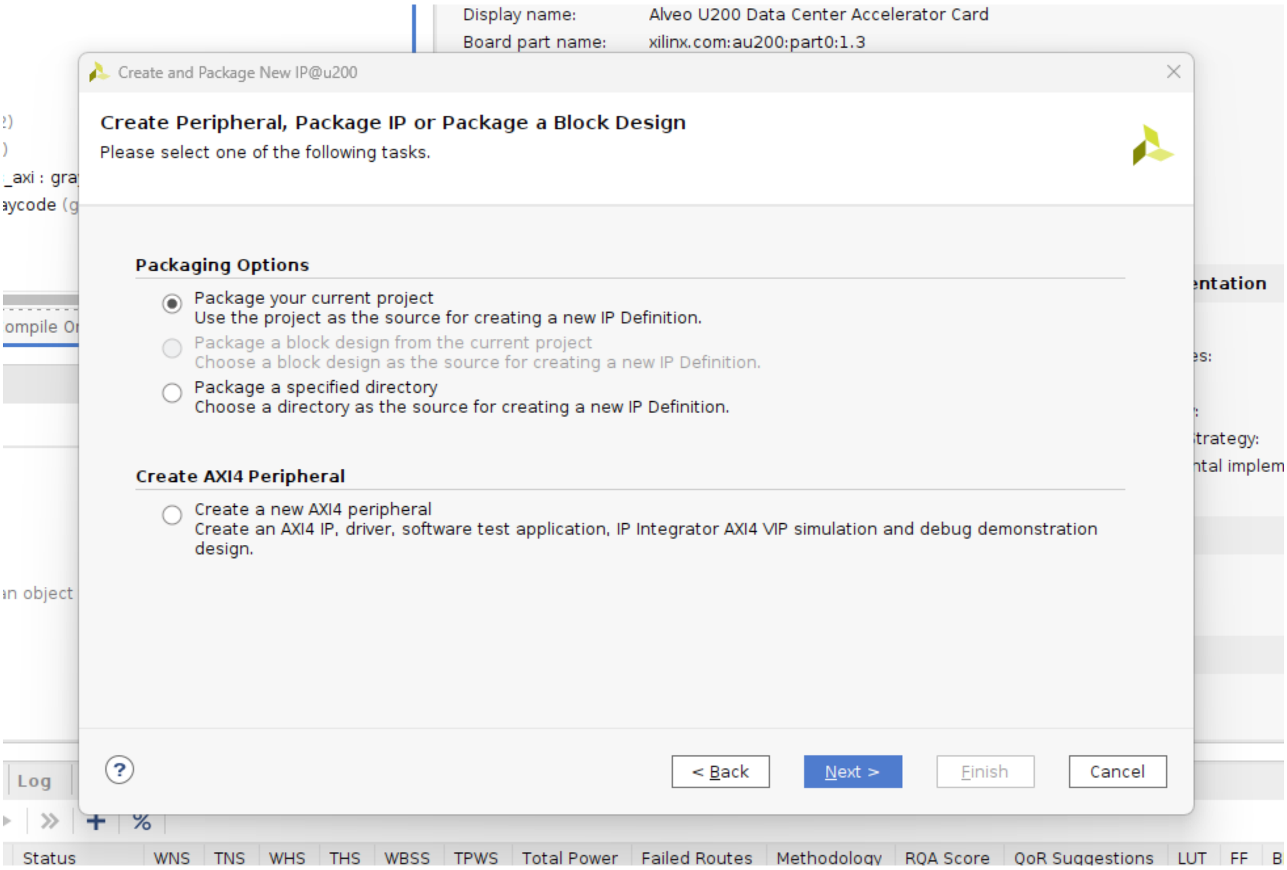
Run Implementation to see DRC results

Utilization

Run Synthesis to see utilization results

Then, we are going to pack our rtl code into IP (`.xo`)

- Click `Tools > Create and Package New IP...` on the tool bar on the top, then click `Next`
- Select `Package your current project`, then click `Next`



Display name: Alveo U200 Data Center Accelerator Card
Board part name: xilinx.com:au200:part0:1.3

Create and Package New IP@u200

Create Peripheral, Package IP or Package a Block Design

Please select one of the following tasks.

Packaging Options

● Package your current project
Use the project as the source for creating a new IP Definition.

○ Package a block design from the current project
Choose a block design as the source for creating a new IP Definition.

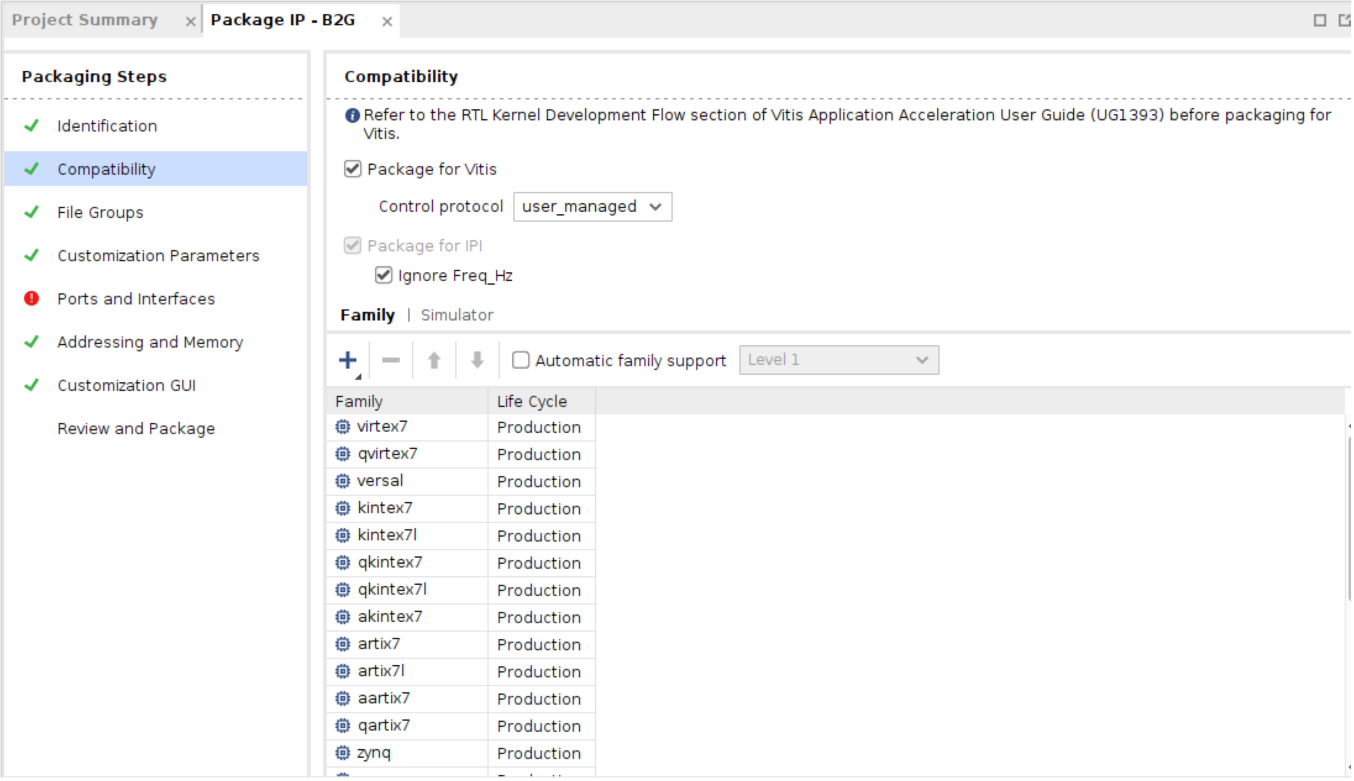
○ Package a specified directory
Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

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

- Click `Next` and `Finish`
- In `Compatibility`, enable `Package for Vitis`, `Package for IPI`, and `Ignore Freq_Hz`
- Change Control protocl to `user_managed`

You can now see the following window



Compatibility

Refer to the RTL Kernel Development Flow section of Vitis Application Acceleration User Guide (UG1393) before packaging for Vitis.

☑ Package for Vitis

Control protocol: user_managed

☑ Package for IPI

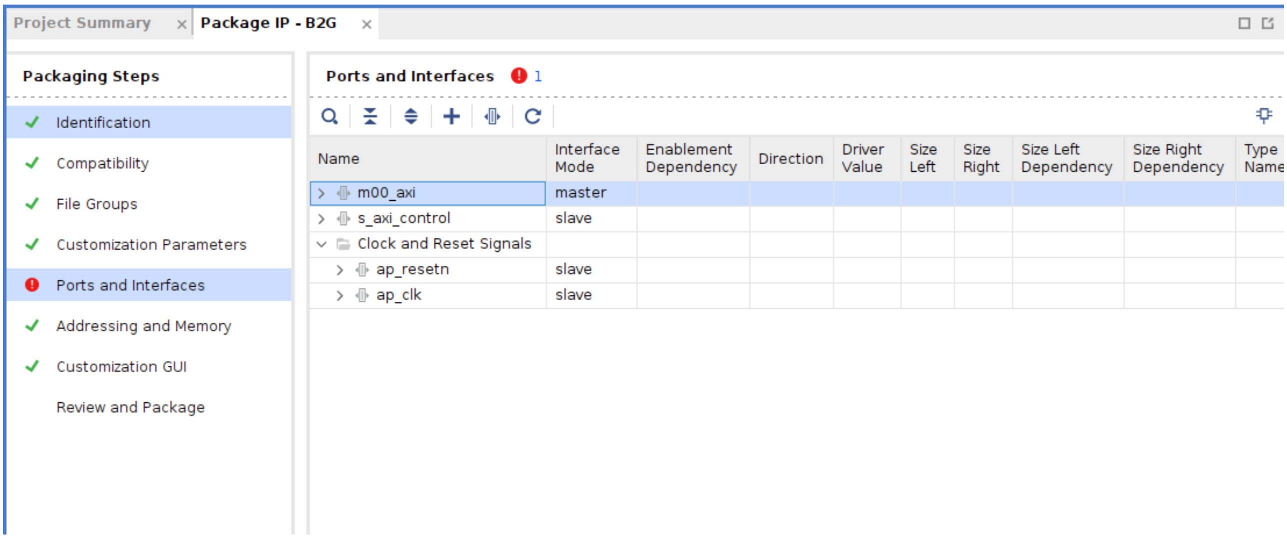
☑ Ignore Freq_Hz

Family | Simulator

☐ Automatic family support Level 1

Family	Life Cycle
virtex7	Production
qvirtex7	Production
versal	Production
kintex7	Production
kintex7l	Production
qkintex7	Production
qkintex7l	Production
akintex7	Production
artix7	Production
artix7l	Production
aartix7	Production
qartix7	Production
zynq	Production

- In `Ports and Interfaces`, add associated clock for `m00_axi`, `s_axi_control`, and `ap_resetn`



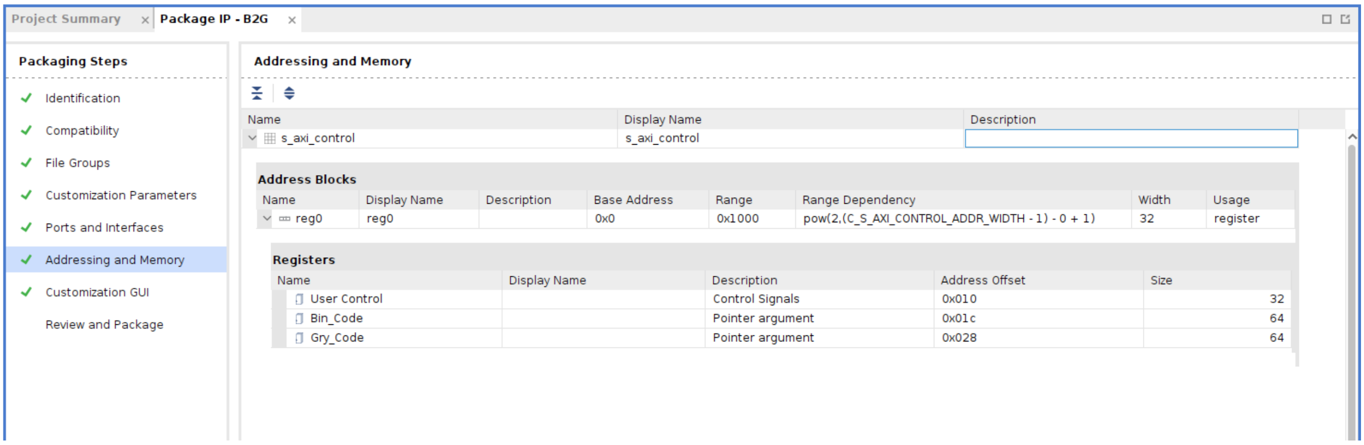
Name	Interface Mode	Enablement Dependency	Direction	Driver Value	Size Left	Size Right	Size Left Dependency	Size Right Dependency	Type Name
m00_axi	master								
s_axi_control	slave								
Clock and Reset Signals									
ap_resetn	slave								
ap_clk	slave								

- Right Click `m00_axi`, click `Associate Clocks...`, enable `ap_clk`, then click `OK`, repeat the above steps for `s_axi_control`, `ap_resetn`

- In `Addressing and Memory`, right click `reg0`, click `Add Register` for the follwing register.

Name	Description	Offset	Size (bits)	ASSOCIATED_BUSIF
User Control	Control Signals	0x010	32	
Bin_Code	Pointer argument	0x01c	64	m00_axi
Gry_Code	Pointer argument	0x028	64	m00_axi

Then you can see the window:



Name	Display Name	Description
s_axi_control	s_axi_control	

Address Blocks

Name	Display Name	Description	Base Address	Range	Range Dependency	Width	Usage
reg0	reg0		0x0	0x1000	pow(2,(C_S_AXI_CONTROL_ADDR_WIDTH - 1) - 0 + 1)	32	register

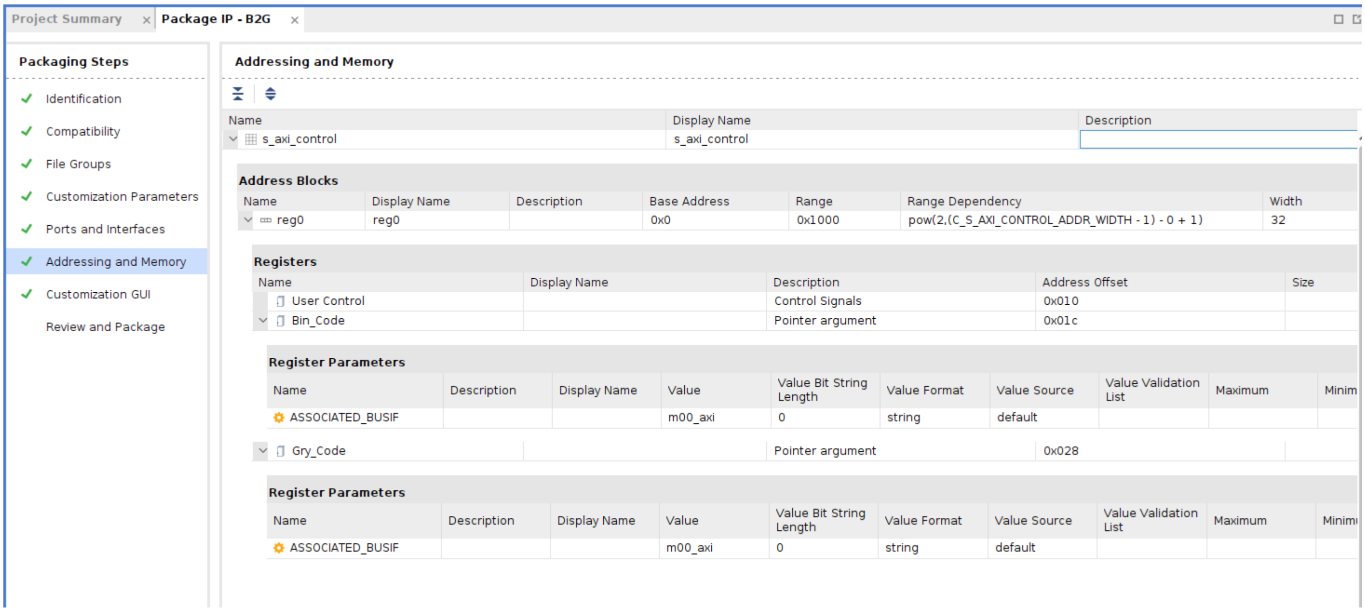
Registers

Name	Display Name	Description	Address Offset	Size
User Control		Control Signals	0x010	32
Bin_Code		Pointer argument	0x01c	64
Gry_Code		Pointer argument	0x028	64

Now, wee need to add associated bus interface for our Bin_Code and Gry_Code.

- Right Click `Bin_Code`, click `Add Register Parameter...`, enter `ASSOCIATED_BUSIF`, then click `OK`
- Right Click `Gry_Code`, click `Add Register Parameter...`, enter `ASSOCIATED_BUSIF`, then click `OK`
- Fill in the value `m00_axi`, in the `Value` for both ASSOCIATED_BUSIF

Now, you can see the following window:



Addressing and Memory

Name	Display Name	Description
s_axi_control	s_axi_control	

Address Blocks

Name	Display Name	Description	Base Address	Range	Range Dependency	Width
reg0	reg0		0x0	0x1000	pow(2,(C_S_AXI_CONTROL_ADDR_WIDTH - 1) - 0 + 1)	32

Registers

Name	Display Name	Description	Address Offset	Size
User Control		Control Signals	0x010	
Bin_Code		Pointer argument	0x01c	

Register Parameters

Name	Description	Display Name	Value	Value Bit String Length	Value Format	Value Source	Value Validation List	Maximum	Minim
ASSOCIATED_BUSIF			m00_axi	0	string	default			

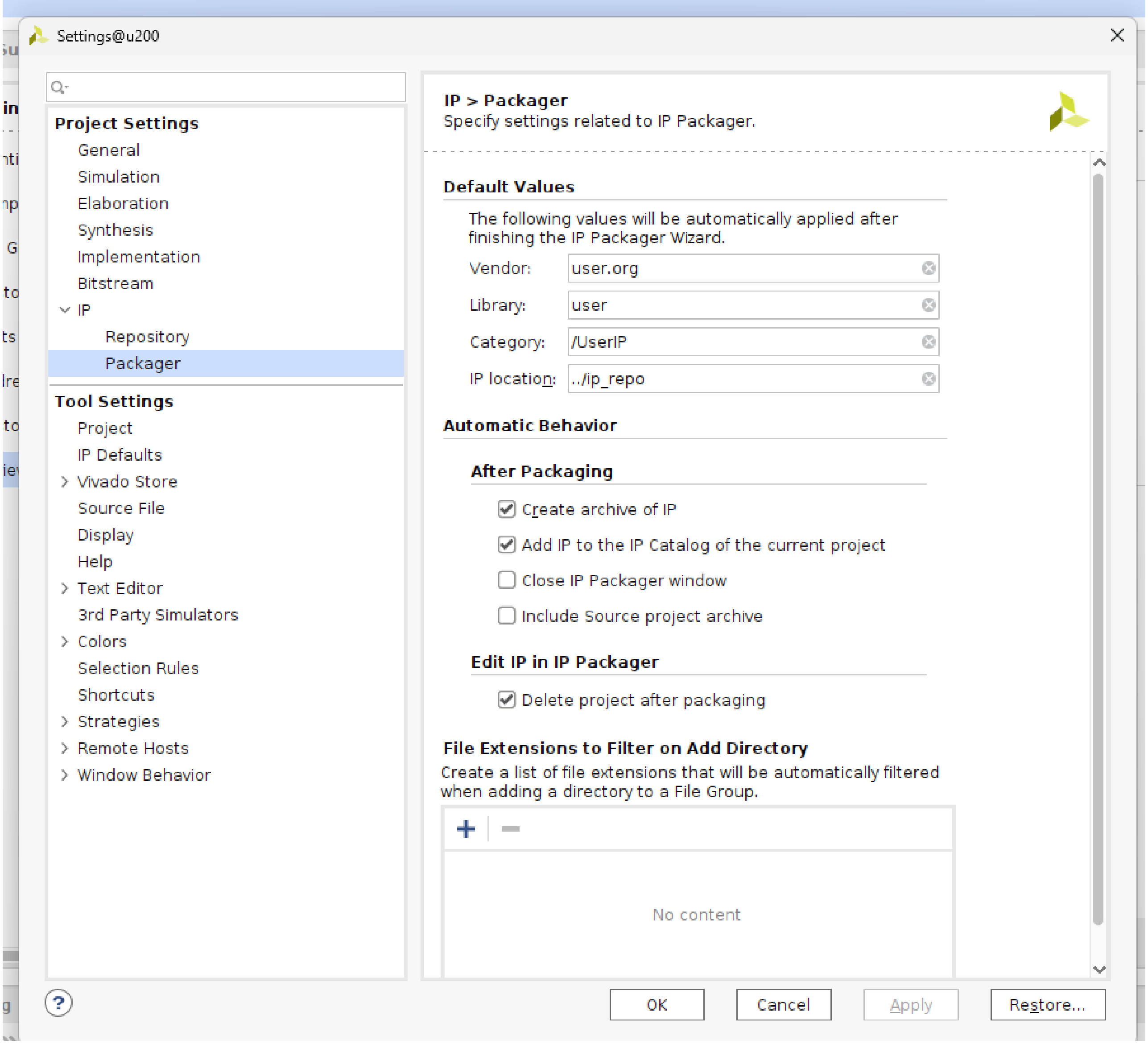
Name	Display Name	Description	Address Offset	Size
Gry_Code		Pointer argument	0x028	

Register Parameters

Name	Description	Display Name	Value	Value Bit String Length	Value Format	Value Source	Value Validation List	Maximum	Minim
ASSOCIATED_BUSIF			m00_axi	0	string	default			

Last,

- In `Review and Package`, click `Edit packaging settings`, make sure your `Create archive of IP` is enable like:



IP > Packager
Specify settings related to IP Packager.

Default Values

The following values will be automatically applied after finishing the IP Packager Wizard.

Vendor: user.org
Library: user
Category: /UserIP
IP location: ../ip_repo

Automatic Behavior

After Packaging

☑ Create archive of IP
☑ Add IP to the IP Catalog of the current project
☐ Close IP Packager window
☐ Include Source project archive

Edit IP in IP Packager

☑ Delete project after packaging

File Extensions to Filter on Add Directory

Create a list of file extensions that will be automatically filtered when adding a directory to a File Group.

No content

- Click `OK`, then `Package IP`, then you can see the info saying you package your ip sucessfully.
- Close your vivado.

Step3: Build up host program in vitis

Open your Vitis IDE

```
$ vitis -workspace work1
```

Then follow: https://github.com/Xilinx/Vitis-Tutorials/blob/2023.2/Hardware_Acceleration/Feature_Tutorials/01-rtl_kernel_workflow/using_the_rtl_kernel.md to finish the host program.

Differences:

Xilinx tutorial	Ours
xilinx_u250_gen3x16_xdma_4_1_202210_1	xilinx_u200_gen3x16_xdma_2_202110_1
Vadd_A_B.xo	B2G.xo