MYD-3EG4EV5EV Tutorial

Version V1.1

Revision History

Version	Description	Date
V1.0	Initial release	2020/12/03
V1.1	Add support for CZU5EV	2021/2/5

Table of Contents

1 Overview

This tutorial will guide you to use Vivado 2020.1 to generate a hardware platform for the MYD-3EG4EV5EV development board and run sample application on the development board.

2 Objectives

At the end of this tutorial，you will be able to:

•Using Vivado 2020.1 to create a hardware platform for MYD-3EG4EV5EV Development Suite

•Export hardware platform to Vitis 2020.1

•Create an "hello" application in Vitis 2020.1 and run it on MYD-3EG4EV5EV

•Start MYD-3EG4EV5EV from MicroSD Card

3Tutorial Requirements

This tutorial will require the following software and hardware setups.

3.1 Software

The software requirements for this reference design are:

•Xilinx Vivado 2020.1

•Xilinx Vitis 2020.1

•USB Serial Port Driver

•Putty terminal program

3.2 Hardware

The hardware setup for this reference design is:

•Computers with 4 GB RAM and 1 GB virtual memory (recommended)

•MYD-3EG4EV5EV Development Board

•12V power supply

•Mini_USB Cable

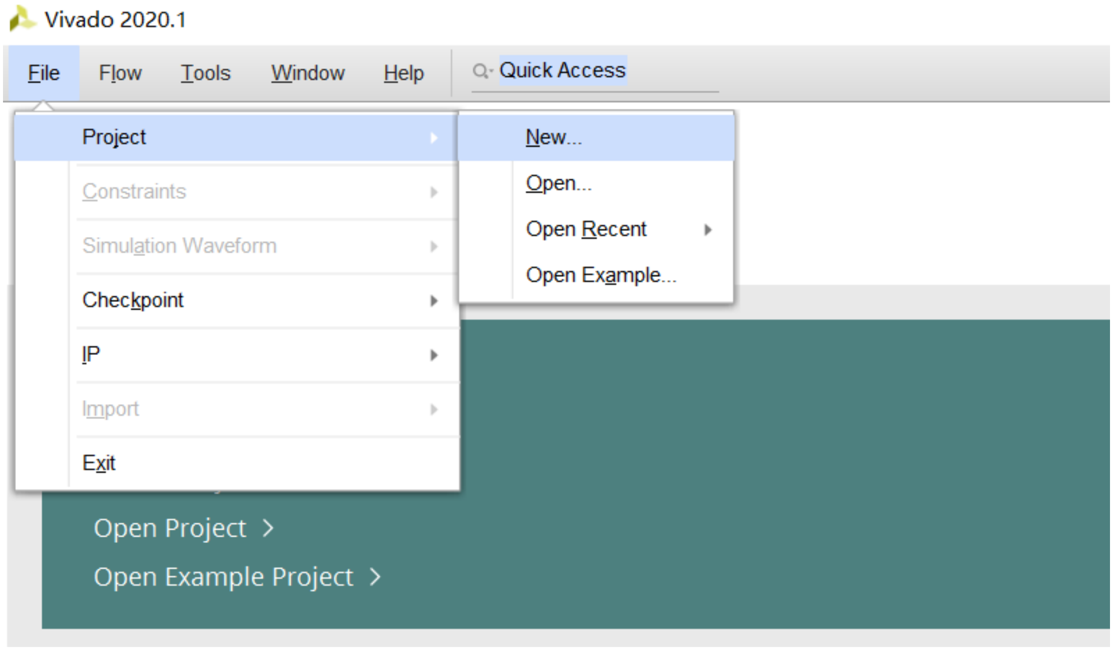
4 Hardware Platform for Generating MYD-3EG4EV Development Board

Follow these steps to generate the hardware platform using Vivado 2020.1 tools.

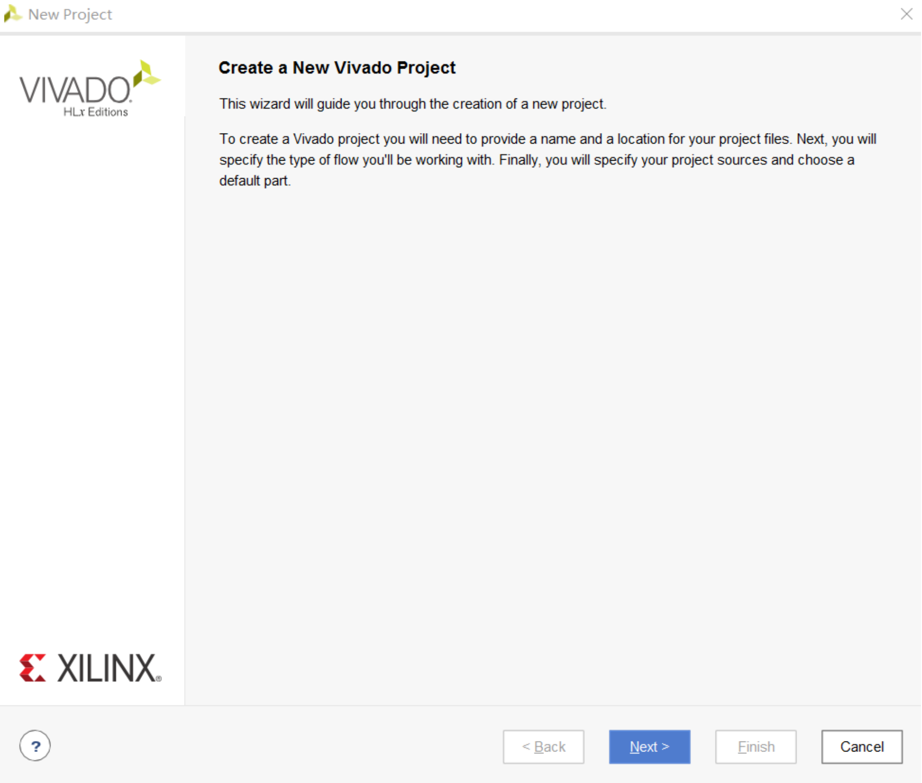
- Start the Vivado tool via Start > All Programs > Xilinx Design Tools > Vivado 2020.1 > Vivado 2020.1

Start vivado

- New project



- Click Next



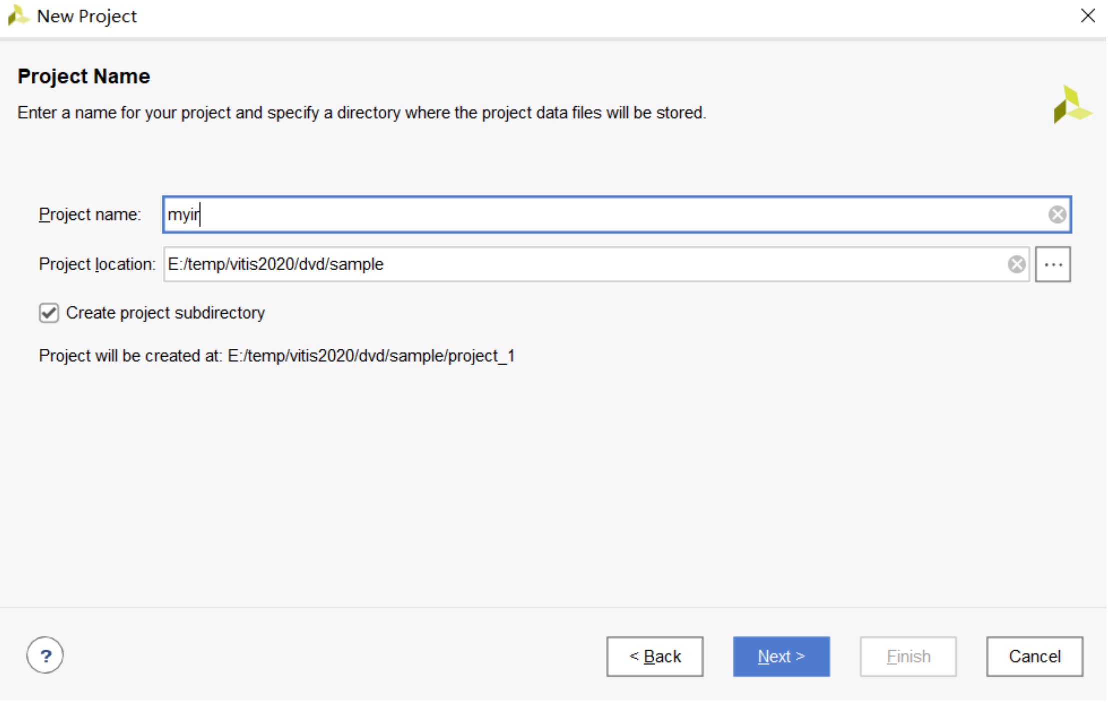
New Project

Create a New Vivado Project

This wizard will guide you through the creation of a new project.

To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.

•Fill in the name of the project and the save path of the project, and click Next.



New Project

Project Name

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

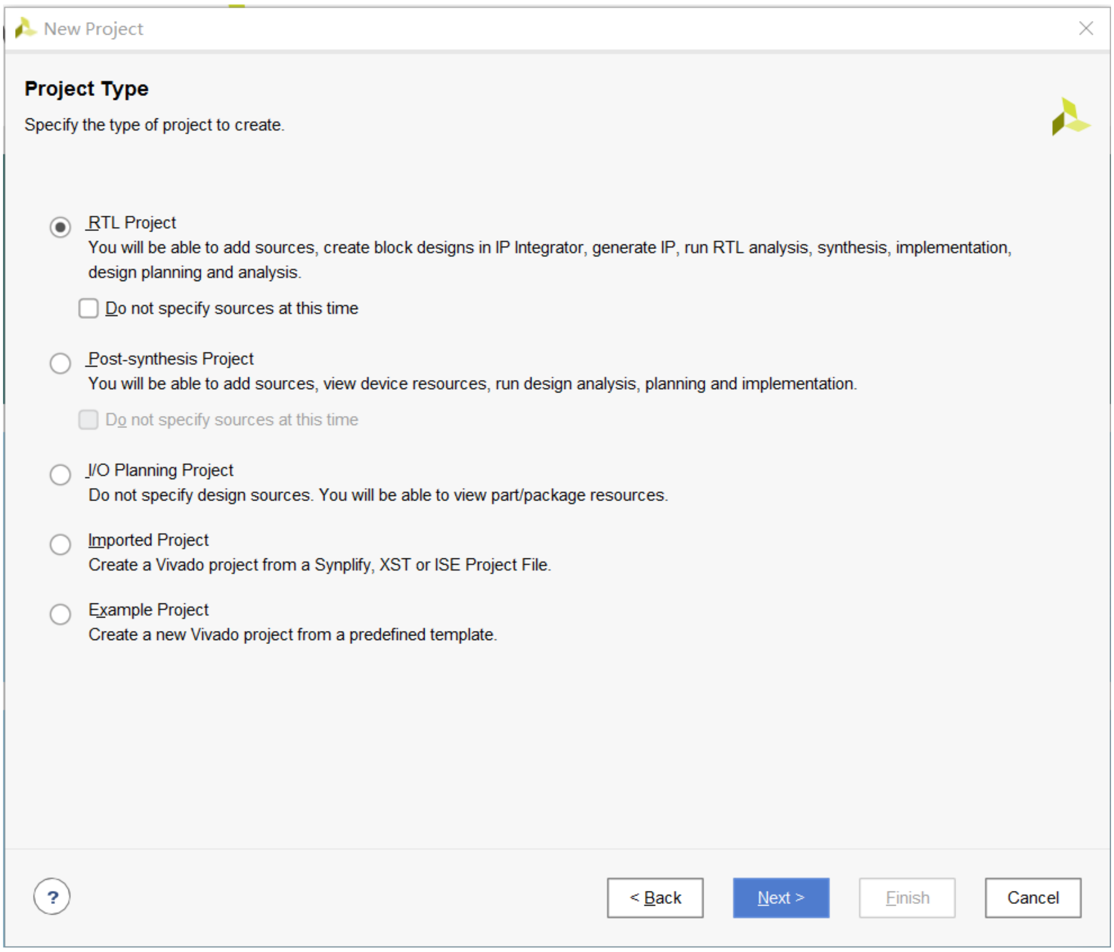
Project name: myir

Project location: E:/temp/vitis2020/dvd/sample

☑ Create project subdirectory

Project will be created at: E:/temp/vitis2020/dvd/sample/project_1

- Select to create an RTL_Project and click Next



New Project

Project Type

Specify the type of project to create.

● RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

□ Do not specify sources at this time

○ Post-synthesis Project
You will be able to add sources, view device resources, run design analysis, planning and implementation.

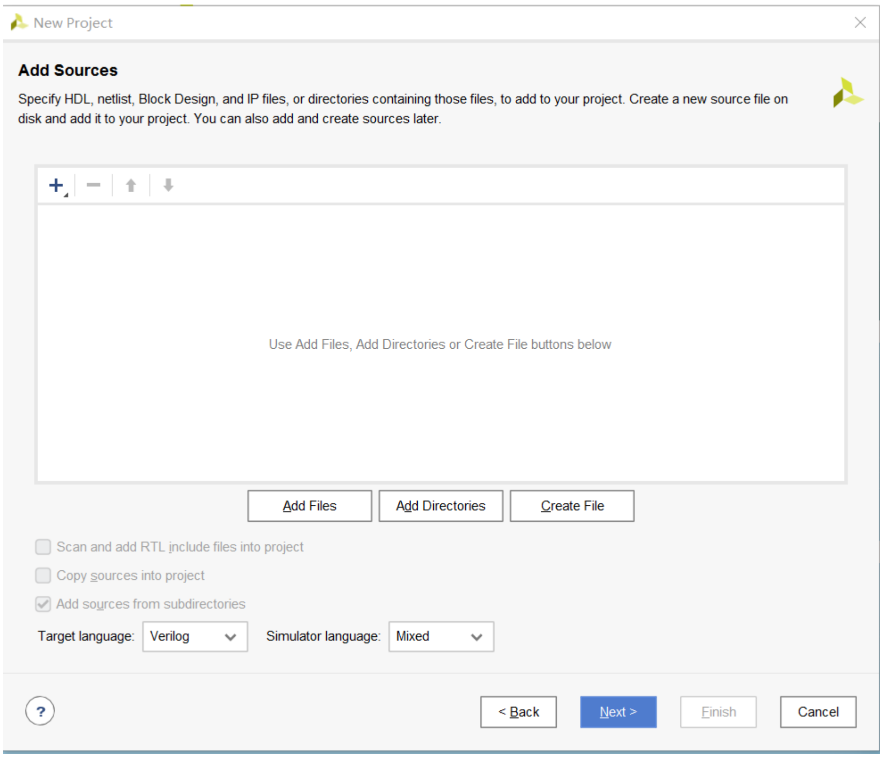
□ Do not specify sources at this time

○ I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.

○ Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

○ Example Project
Create a new Vivado project from a predefined template.

- Continue clicking Next



New Project

Add Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.

Use Add Files, Add Directories or Create File buttons below

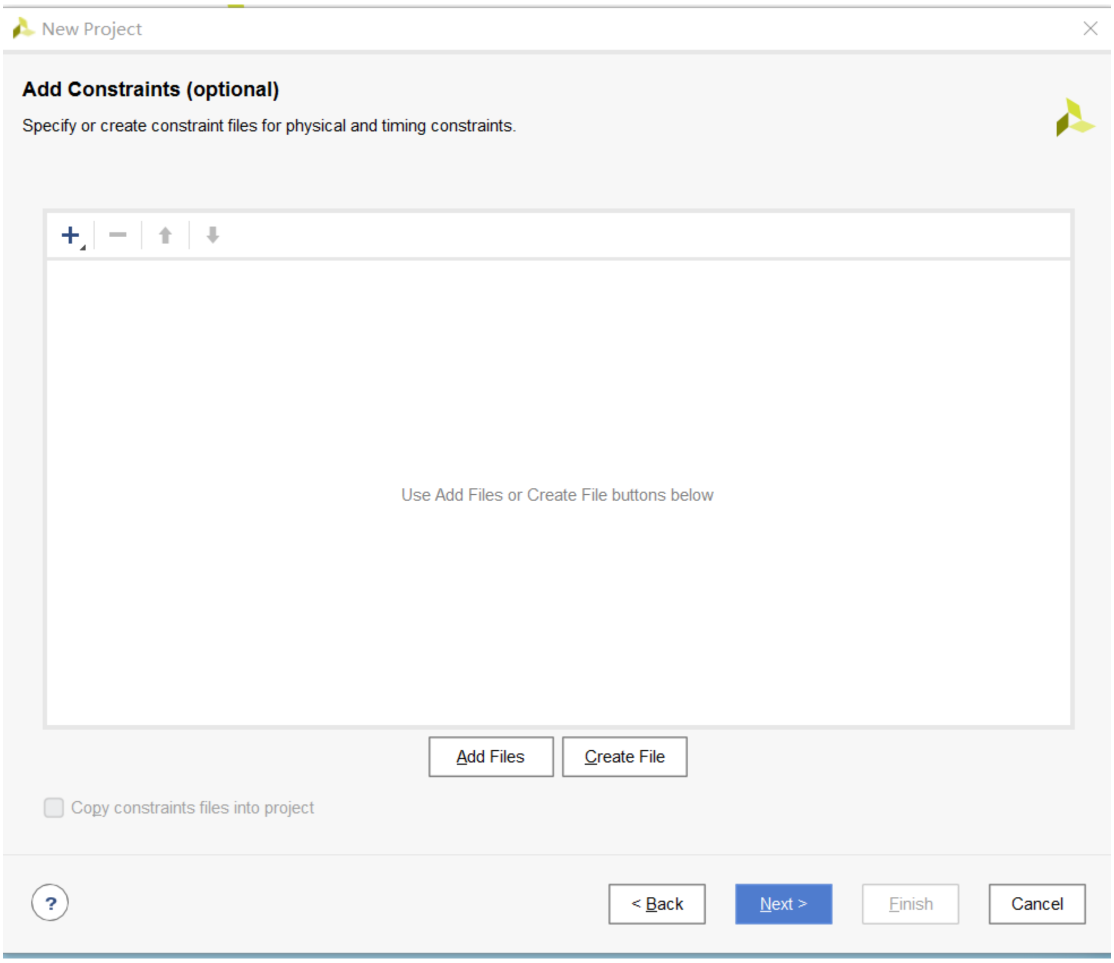
□ Scan and add RTL include files into project

□ Copy sources into project

☑ Add sources from subdirectories

Target language: Verilog   Simulator language: Mixed

- Continue clicking Next



Add Constraints (optional)

Specify or create constraint files for physical and timing constraints.

Use Add Files or Create File buttons below

□ Copy constraints files into project

- In the pop-up dialog box:

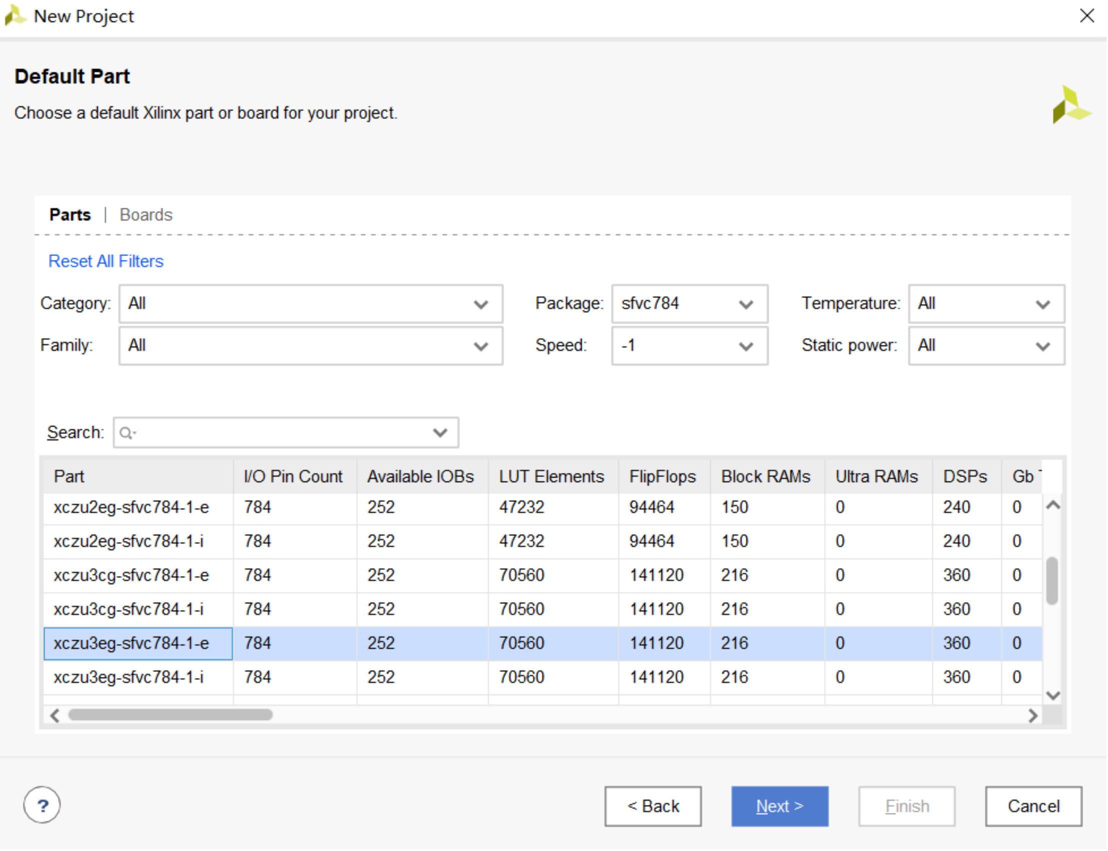
a. Click Select Parts.

b. Click the drop-down box to set speed grade to - 1.

c. The package is set to sfvc784.

d.Choose the development board chip as xczu3eg-sfvc784-1-e

e. Click Next.（If it is xczu3eg-sfvc784-1-e, follow this step to select the chip model）



Default Part

Choose a default Xilinx part or board for your project.

Category: All

Family: All

Package: sfvc784

Speed: -1

Temperature: All

Static power: All

Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb
xczu2eg-sfvc784-1-e	784	252	47232	94464	150	0	240	0
xczu2eg-sfvc784-1-i	784	252	47232	94464	150	0	240	0
xczu3cg-sfvc784-1-e	784	252	70560	141120	216	0	360	0
xczu3cg-sfvc784-1-i	784	252	70560	141120	216	0	360	0
xczu3eg-sfvc784-1-e	784	252	70560	141120	216	0	360	0
xczu3eg-sfvc784-1-i	784	252	70560	141120	216	0	360	0

- In the pop-up dialog box:

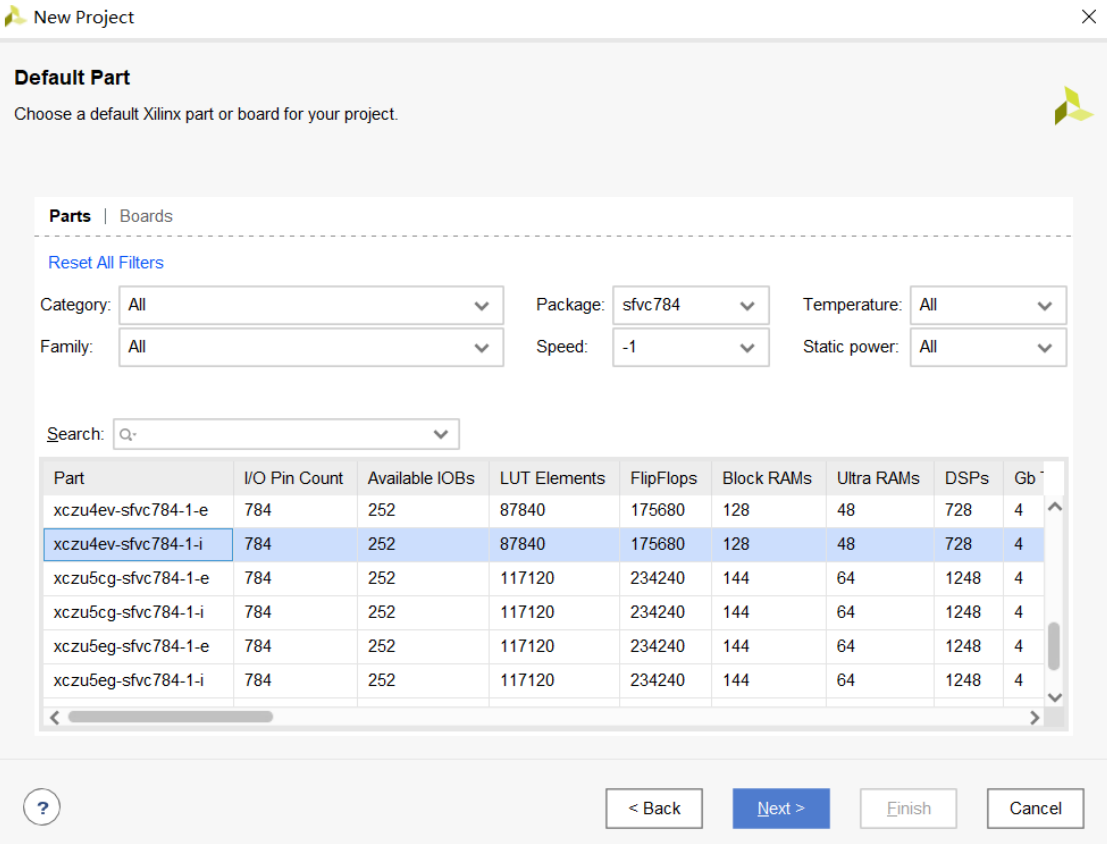
a. Click Select Parts.

b. Click the drop-down box to set speed grade to - 1.

c. The package is set to sfvc784.

d.Choose the development board chip as xczu4ev-sfvc784-1-i

e. Click Next.（If it is xczu4ev-sfvc784-1-i, follow this step to select the chip model）



Default Part

Choose a default Xilinx part or board for your project.

Category: All

Family: All

Package: sfvc784

Speed: -1

Temperature: All

Static power: All

Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs	Gb
xczu4ev-sfvc784-1-e	784	252	87840	175680	128	48	728	4
xczu4ev-sfvc784-1-i	784	252	87840	175680	128	48	728	4
xczu5cg-sfvc784-1-e	784	252	117120	234240	144	64	1248	4
xczu5cg-sfvc784-1-i	784	252	117120	234240	144	64	1248	4
xczu5eg-sfvc784-1-e	784	252	117120	234240	144	64	1248	4
xczu5eg-sfvc784-1-i	784	252	117120	234240	144	64	1248	4

- In the pop-up dialog box:

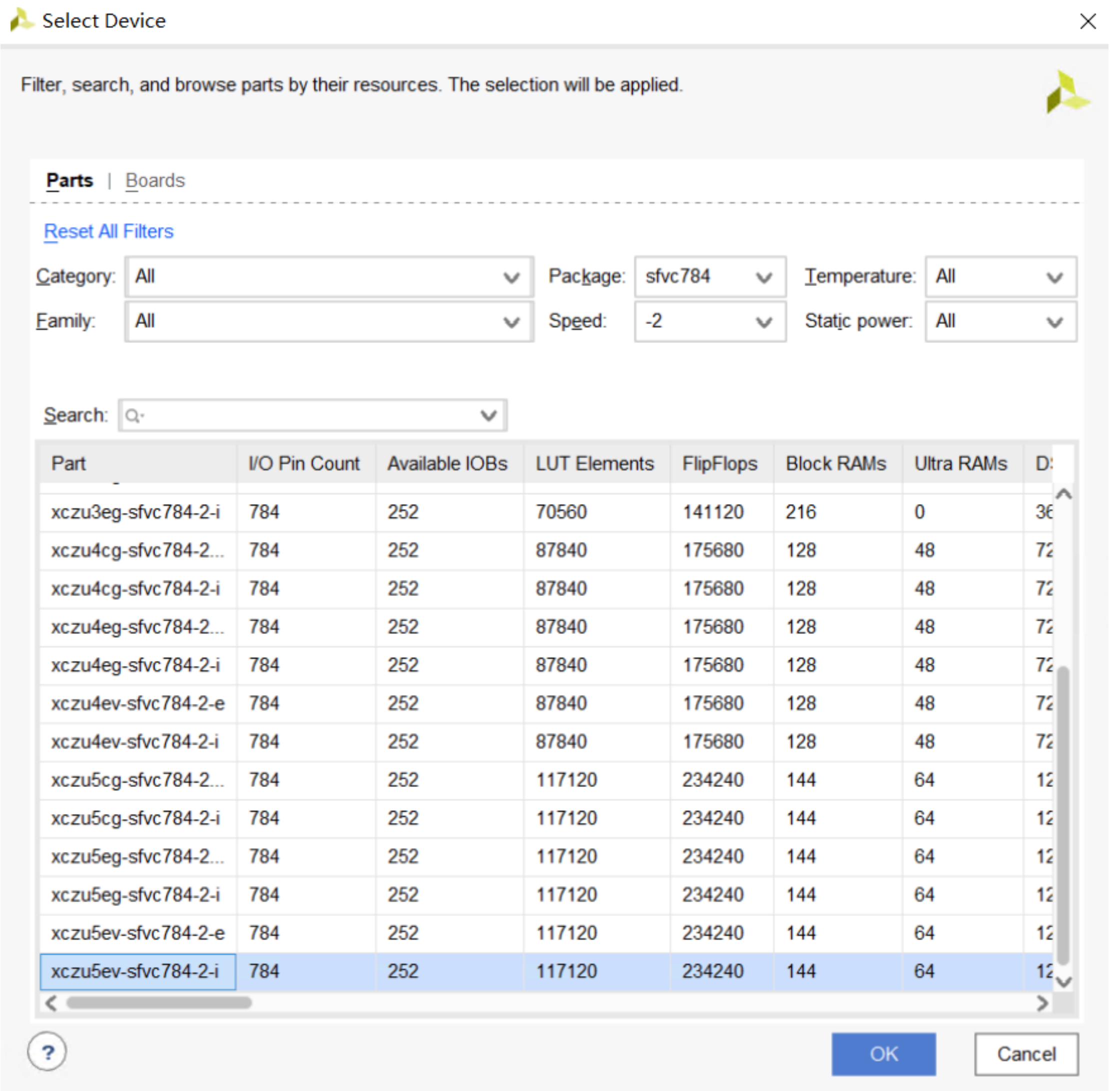
a. Click Select Parts.

b. Click the drop-down box to set speed grade to - 2.

c. The package is set to sfvc784.

d.Choose the development board chip as xczu5ev-sfvc784-2-i

e. Click Next.（If it is xczu5ev-sfvc784-2-i, follow this step to select the chip model）



Select Device

Filter, search, and browse parts by their resources. The selection will be applied.

Category: All

Family: All

Package: sfvc784

Speed: -2

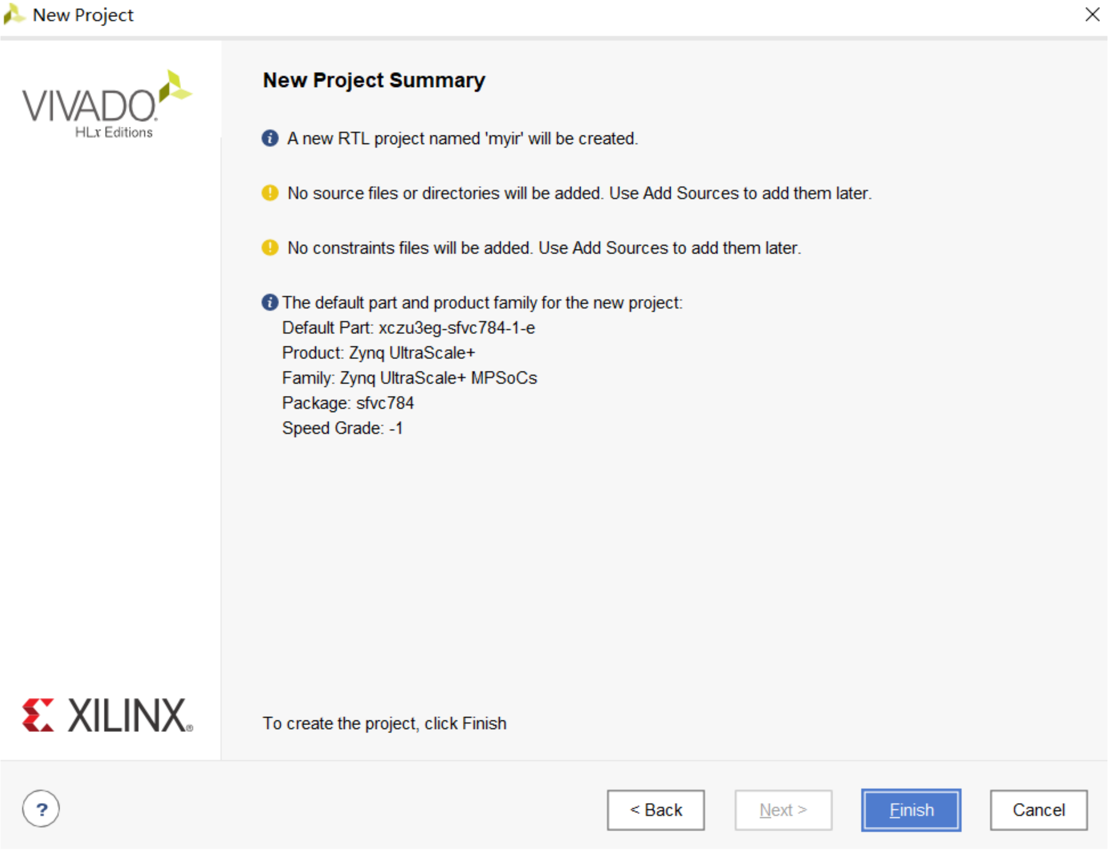
Temperature: All

Static power: All

Search:

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	D
xczu3eg-sfvc784-2-i	784	252	70560	141120	216	0	36
xczu4cg-sfvc784-2...	784	252	87840	175680	128	48	72
xczu4cg-sfvc784-2-i	784	252	87840	175680	128	48	72
xczu4eg-sfvc784-2...	784	252	87840	175680	128	48	72
xczu4eg-sfvc784-2-i	784	252	87840	175680	128	48	72
xczu4ev-sfvc784-2-e	784	252	87840	175680	128	48	72
xczu4ev-sfvc784-2-i	784	252	87840	175680	128	48	72
xczu5cg-sfvc784-2...	784	252	117120	234240	144	64	12
xczu5cg-sfvc784-2-i	784	252	117120	234240	144	64	12
xczu5eg-sfvc784-2...	784	252	117120	234240	144	64	12
xczu5eg-sfvc784-2-i	784	252	117120	234240	144	64	12
xczu5ev-sfvc784-2-e	784	252	117120	234240	144	64	12
xczu5ev-sfvc784-2-i	784	252	117120	234240	144	64	12

- In the New Project dialog box, click Finish



New Project Summary

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

No source files or directories will be added. Use Add Sources to add them later.

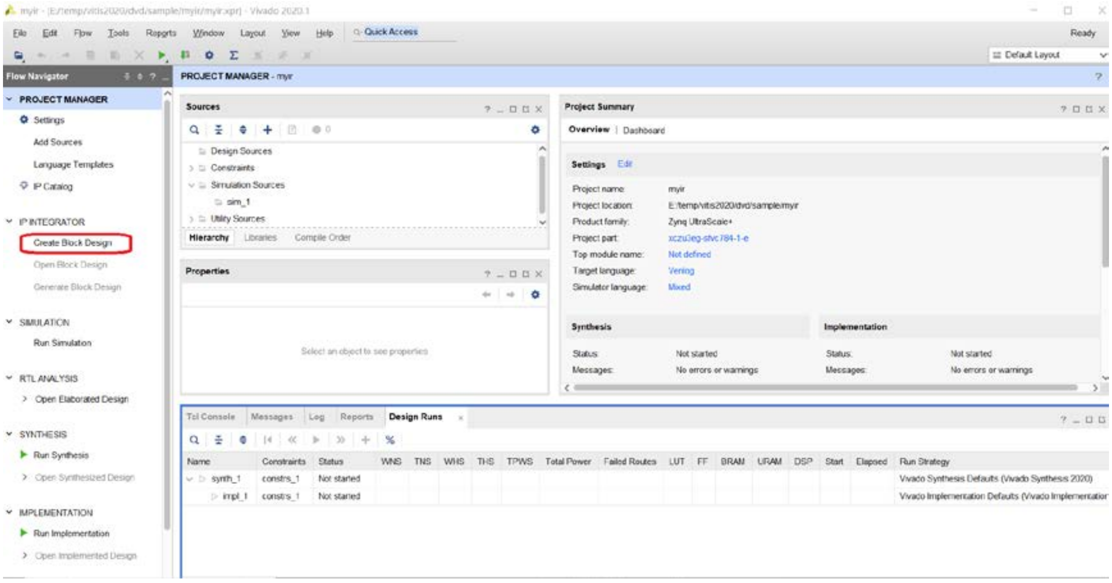
No constraints files will be added. Use Add Sources to add them later.

The default part and product family for the new project:
Default Part: xczu3eg-sfvc784-1-e
Product: Zynq UltraScale+
Family: Zynq UltraScale+ MPSoCs
Package: sfvc784
Speed Grade: -1

To create the project, click Finish

4. 1 Create a Block Design

•Click IP Integrator->Create Block Design



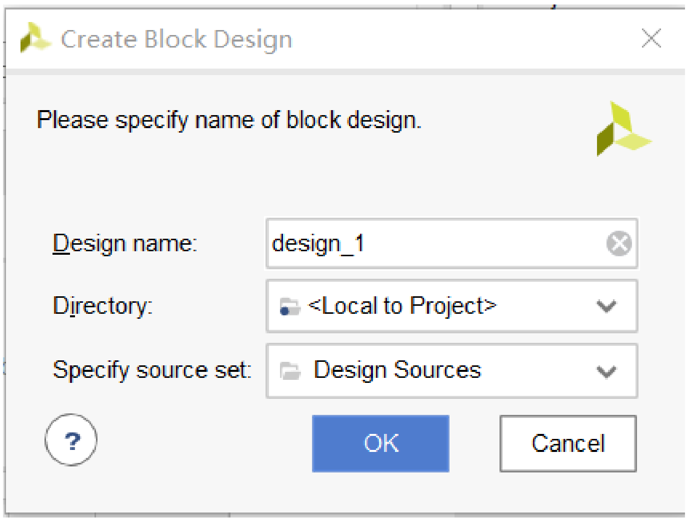
Project name: myir
Project location: E:/temp/vitis2020/dvd/sample/myir
Product family: Zynq UltraScale+
Project part: xczu3eg-sfvc784-1-e
Top module name: Not defined
Target language: Verilog
Simulator language: Mixed

Synthesis
Status: Not started
Messages: No errors or warnings

Implementation
Status: Not started
Messages: No errors or warnings

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Not started															Vivado Synthesis Defaults (Vivado Synthesis 2020)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado Implementatior

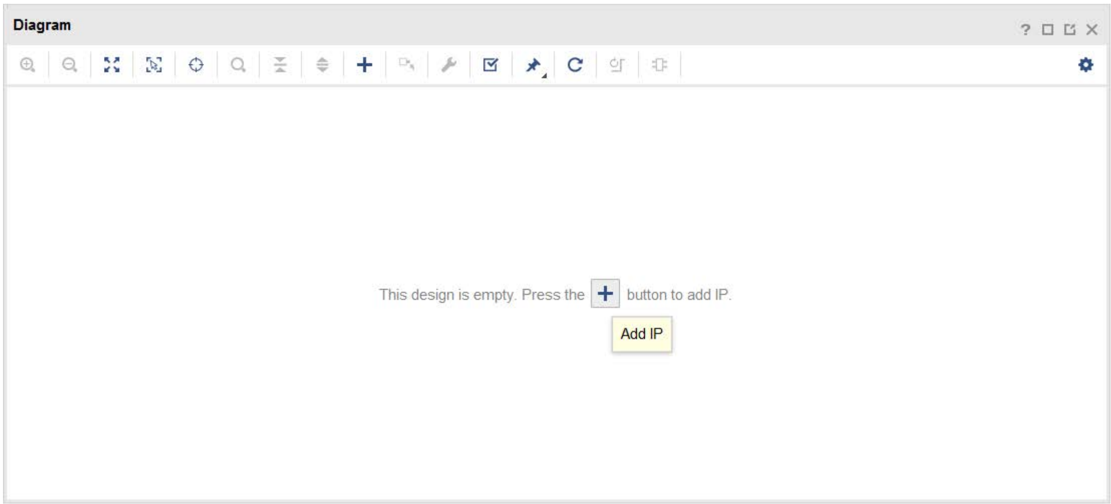
•You can click Design name to change design_1 to the name you want. Here, design_1 is the default name.



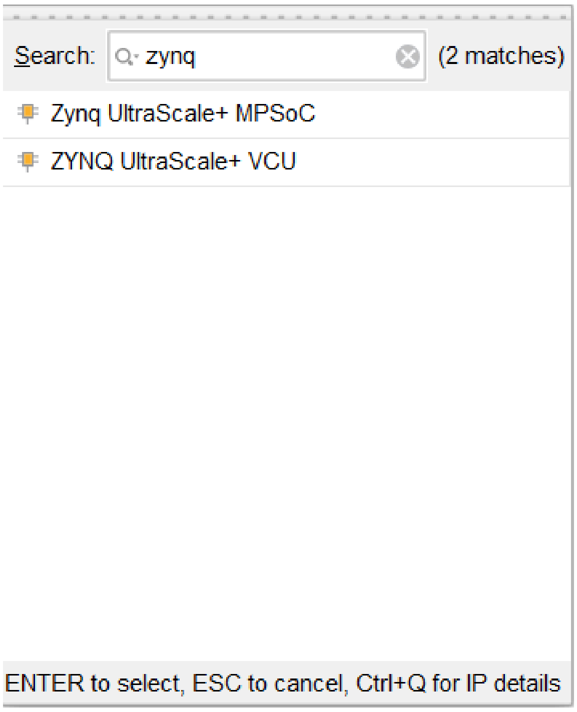
Please specify name of block design.

Design name: design_1
Directory: <Local to Project>
Specify source set: Design Sources

•Click Add IP in the figure below to add an IP core



•Click on the search bar to enter ZYNQ, and then double-click the searched Zynq UltraScale + MPSoC core

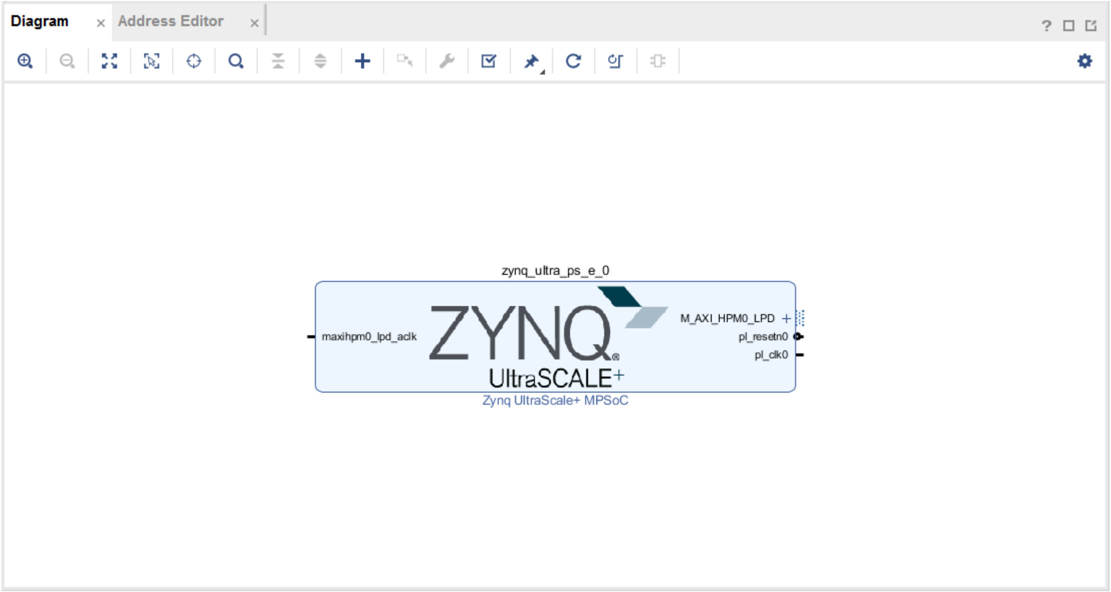

Search: zynq (2 matches)

Zynq UltraScale+ MPSoC

ZYNQ UltraScale+ VCU

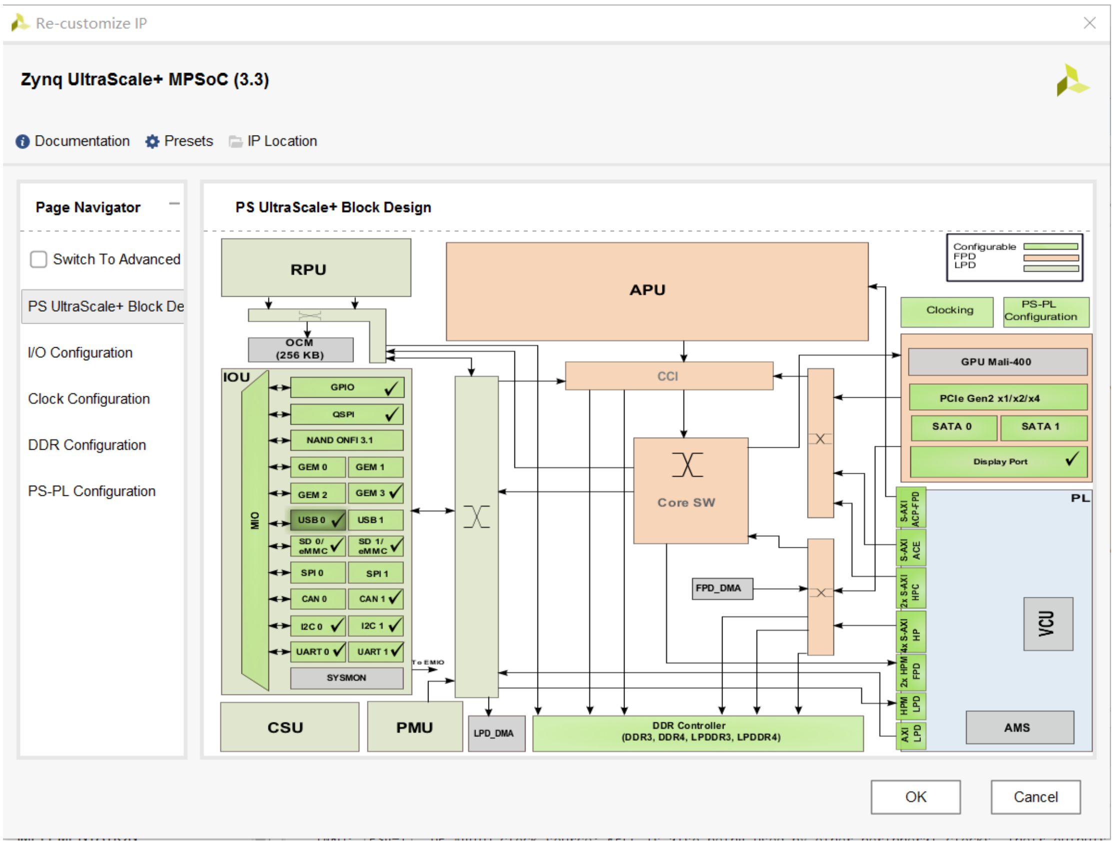
ENTER to select, ESC to cancel, Ctrl+Q for IP details

- The Zynq UltraScale + MPSoC core was added to the design, as shown in the following figure



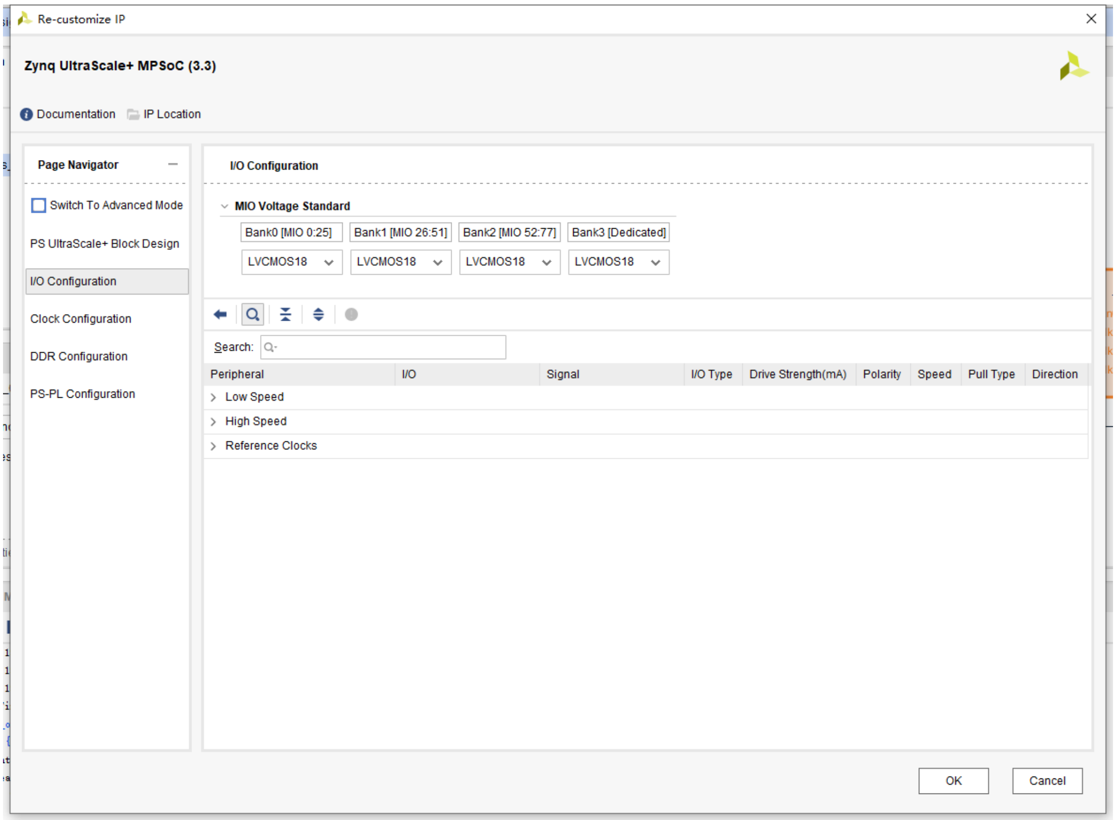
•Double-click the Zynq UltraScale + MPSoC core

Click the left item to enter editing. The functions of each window are introduced below.



1．Voltage configuration

In the I/O configuration window, configure the voltage of BANK0~BANK3 to “LVCMOS18”.



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

I/O Configuration

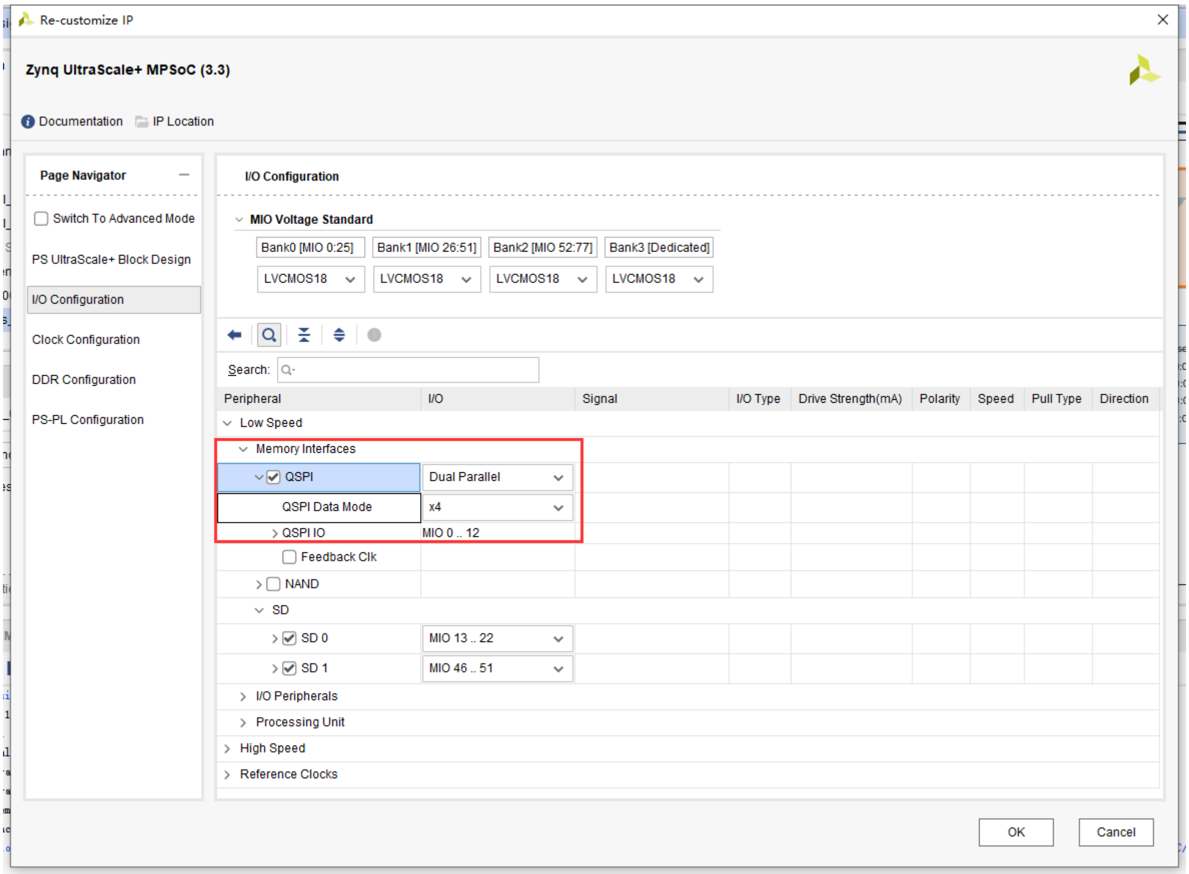
MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
Low Speed								
High Speed								
Reference Clocks								

2．Low speed configuration

Check QSPI and set it to "Dual Parallel" mode and "Data Mode" to "x4"



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation   IP Location

Page Navigator
- □ Switch To Advanced Mode
- PS UltraScale+ Block Design
- I/O Configuration
- Clock Configuration
- DDR Configuration
- PS-PL Configuration

I/O Configuration

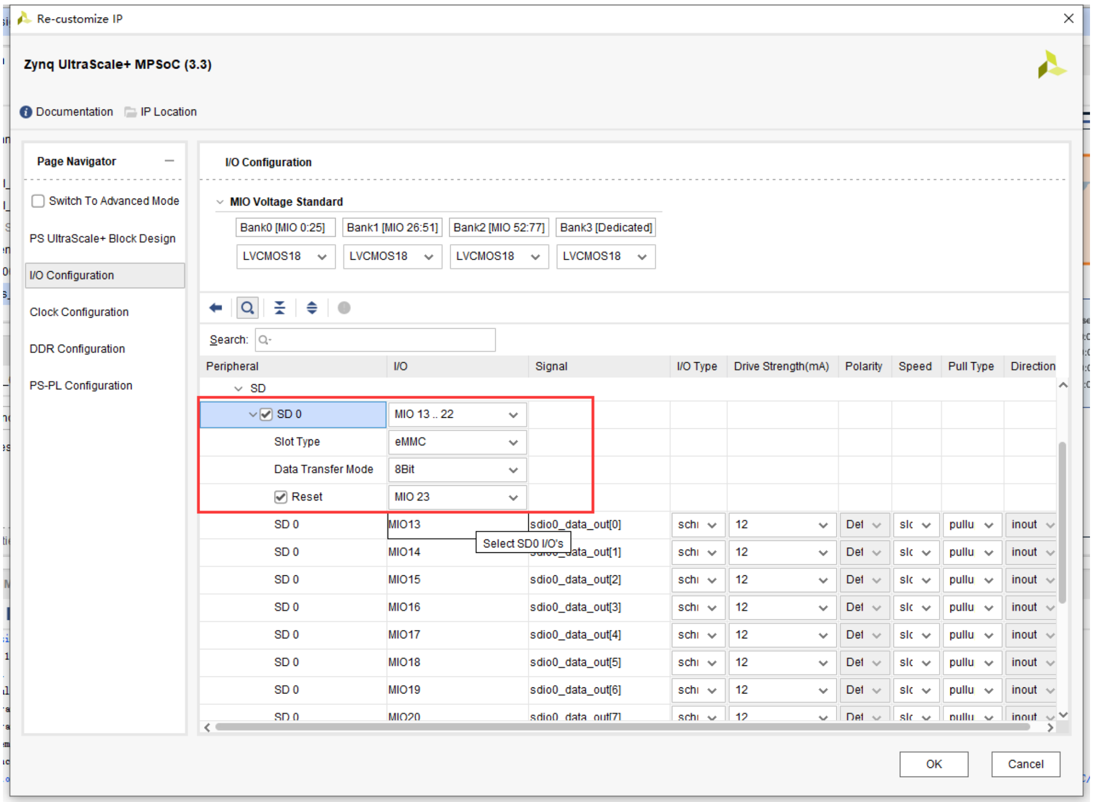
MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
Low Speed								
Memory Interfaces								
☑ QSPI	Dual Parallel							
QSPI Data Mode	x4							
QSPI IO	MIO 0 .. 12							
□ Feedback Clk								
□ NAND								
SD								
☑ SD 0	MIO 13 .. 22							
☑ SD 1	MIO 46 .. 51							
I/O Peripherals								
Processing Unit								
High Speed								
Reference Clocks								

Check SD 0，Select “MIO13..22”, Slot Type “eMMC”, Data Transfer Mode “8bit”, check Reset, and select “MIO23”



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation   IP Location

Page Navigator
- □ Switch To Advanced Mode
- PS UltraScale+ Block Design
- I/O Configuration
- Clock Configuration
- DDR Configuration
- PS-PL Configuration

I/O Configuration

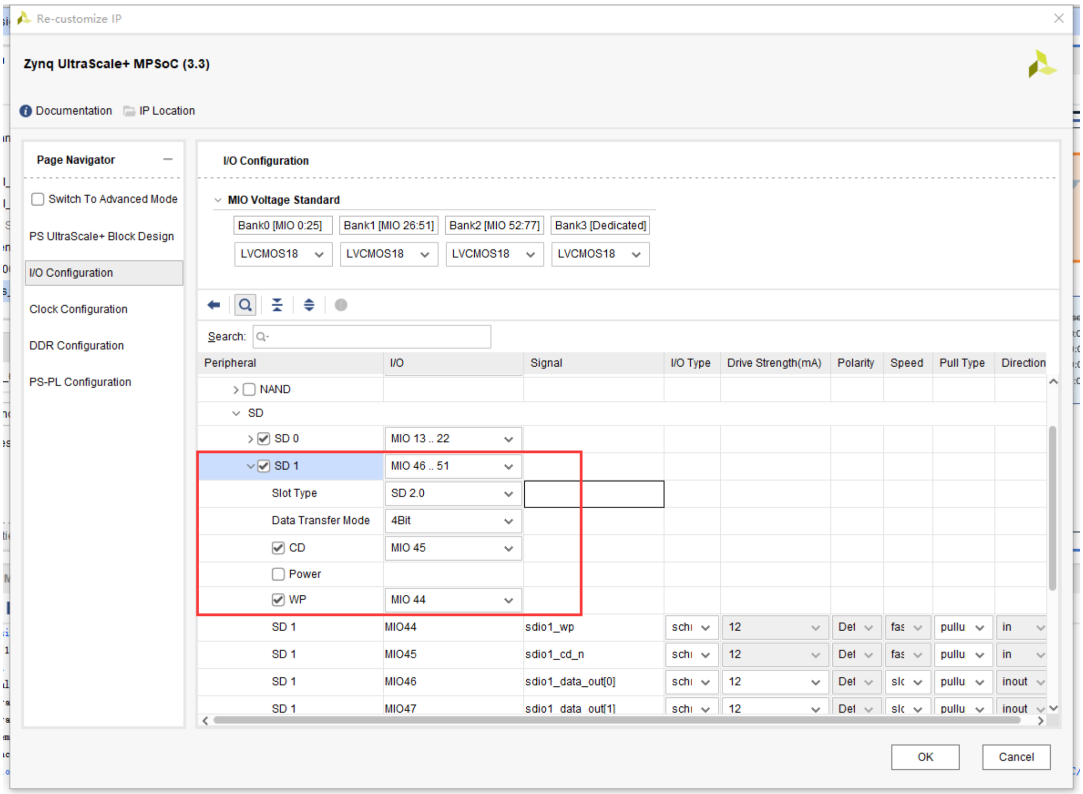
MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
SD								
☑ SD 0	MIO 13 .. 22							
Slot Type	eMMC							
Data Transfer Mode	8Bit							
☑ Reset	MIO 23							
SD 0	MIO13	sdio0_data_out[0]	schi	12	Def	slc	pullu	inout
SD 0	MIO14	sdio0_data_out[1]	schi	12	Def	slc	pullu	inout
SD 0	MIO15	sdio0_data_out[2]	schi	12	Def	slc	pullu	inout
SD 0	MIO16	sdio0_data_out[3]	schi	12	Def	slc	pullu	inout
SD 0	MIO17	sdio0_data_out[4]	schi	12	Def	slc	pullu	inout
SD 0	MIO18	sdio0_data_out[5]	schi	12	Def	slc	pullu	inout
SD 0	MIO19	sdio0_data_out[6]	schi	12	Def	slc	pullu	inout
SD 0	MIO20	sdio0_data_out[7]	schi	12	Def	slc	pullu	inout

Check SD 1. Select “MIO 46..51”, Slot Type “SD 2.0”, Data Transfer Mode select “4bit”, check CD to detect SD card insertion, select “MIO45”, check WP, select “MIO44” for SD card write protection



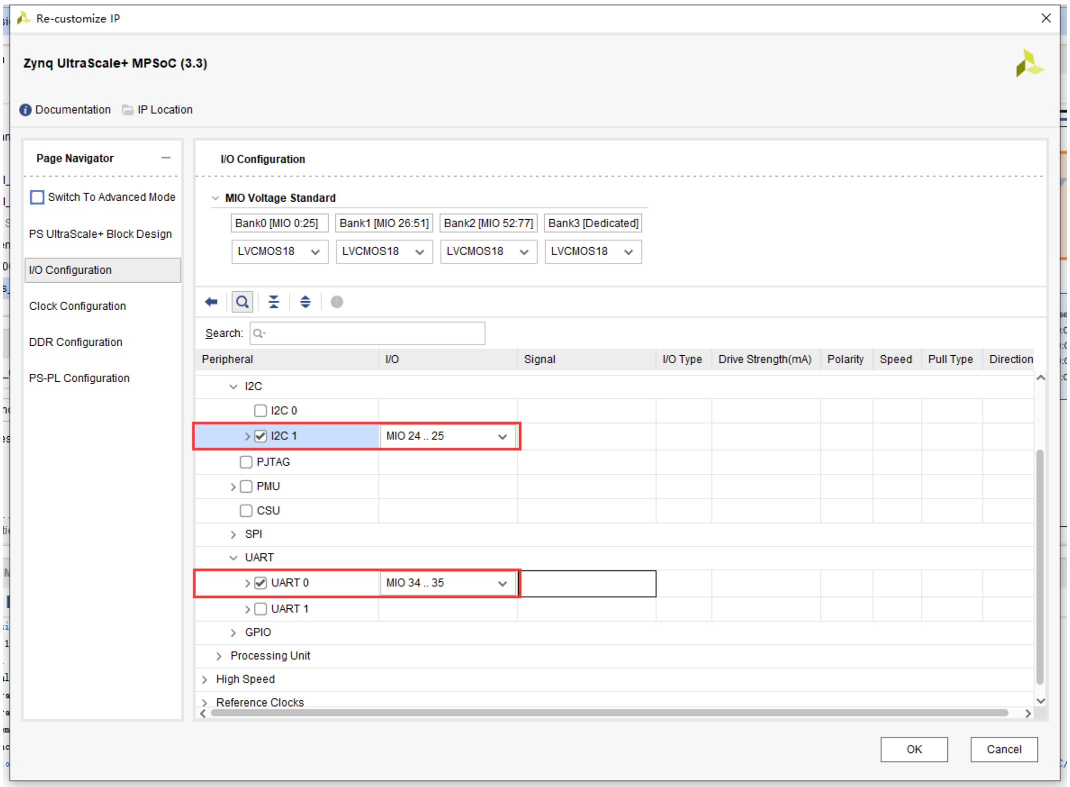
MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
□ NAND								
SD								
☑ SD 0	MIO 13 .. 22							
☑ SD 1	MIO 46 .. 51							
Slot Type	SD 2.0							
Data Transfer Mode	4Bit							
☑ CD	MIO 45							
□ Power								
☑ WP	MIO 44							
SD 1	MIO44	sdio1_wp	schı	12	Def	fas	pullu	in
SD 1	MIO45	sdio1_cd_n	schı	12	Def	fas	pullu	in
SD 1	MIO46	sdio1_data_out[0]	schı	12	Def	slc	pullu	inout
SD 1	MIO47	sdio1_data_out[1]	schı	12	Def	slc	pullu	inout

Check I2C 1， select “MIO24..25”

Check UART 0， select “MIO 34..35”



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation IP Location

Page Navigator

- □ Switch To Advanced Mode
- PS UltraScale+ Block Design
- I/O Configuration
- Clock Configuration
- DDR Configuration
- PS-PL Configuration

I/O Configuration

MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

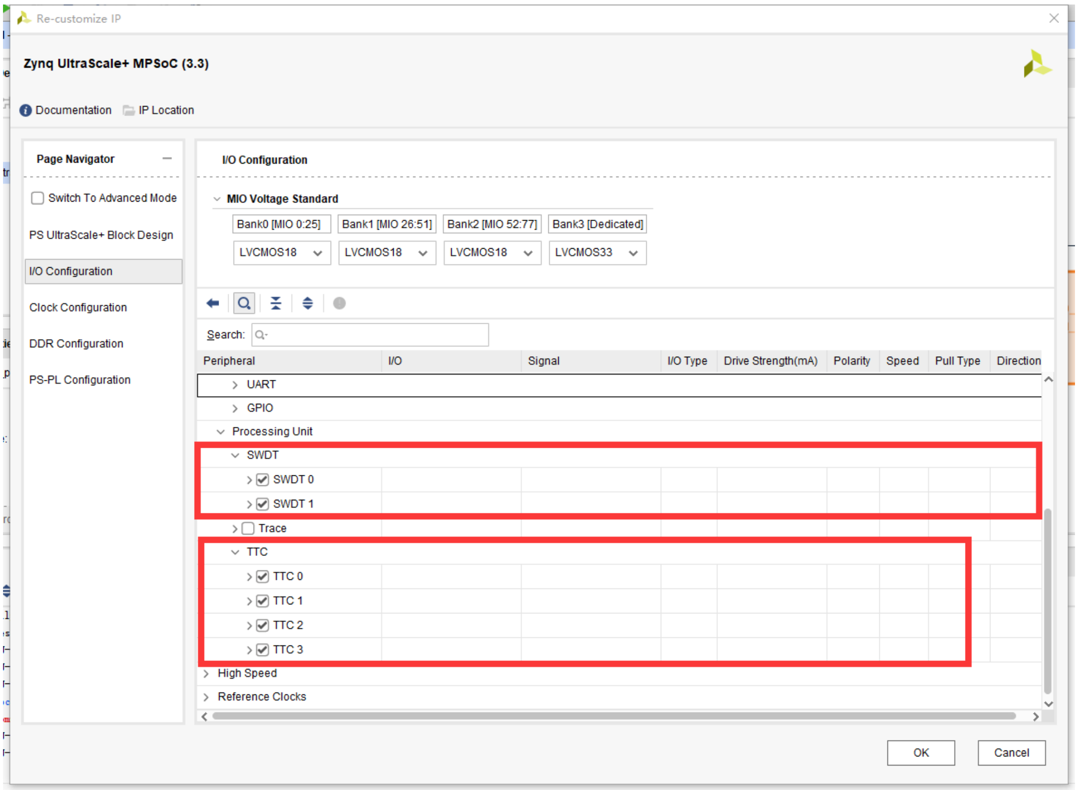
Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
I2C								
□ I2C 0								
☑ I2C 1	MIO 24 .. 25							
□ PJTAG								
□ PMU								
□ CSU								
SPI								
UART								
☑ UART 0	MIO 34 .. 35							
□ UART 1								
GPIO								
Processing Unit								
High Speed								
Reference Clocks								

OK Cancel

Check SWDT 0、SWDT 1

Check TTC 0~TTC 3



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation IP Location

Page Navigator

- □ Switch To Advanced Mode
- PS UltraScale+ Block Design
- I/O Configuration
- Clock Configuration
- DDR Configuration
- PS-PL Configuration

I/O Configuration

MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS33

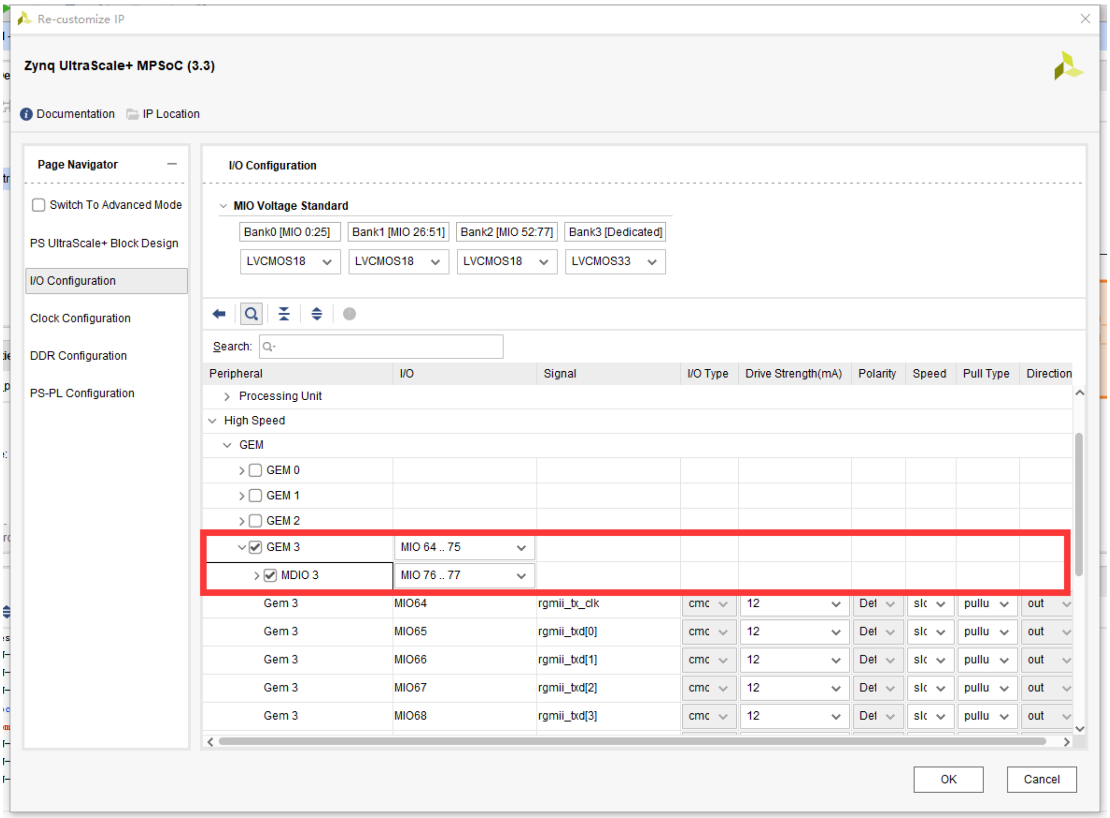
Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
UART								
GPIO								
Processing Unit								
SWDT								
☑ SWDT 0								
☑ SWDT 1								
□ Trace								
TTC								
☑ TTC 0								
☑ TTC 1								
☑ TTC 2								
☑ TTC 3								
High Speed								
Reference Clocks								

OK Cancel

3．High speed configuration

In the high speed part, firstly configure PS Ethernet, check GEM 3, select “MIO 64..75”, check MDIO3 and select “MIO 76..77”



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation IP Location

Page Navigator: □ Switch To Advanced Mode; PS UltraScale+ Block Design; I/O Configuration; Clock Configuration; DDR Configuration; PS-PL Configuration

I/O Configuration

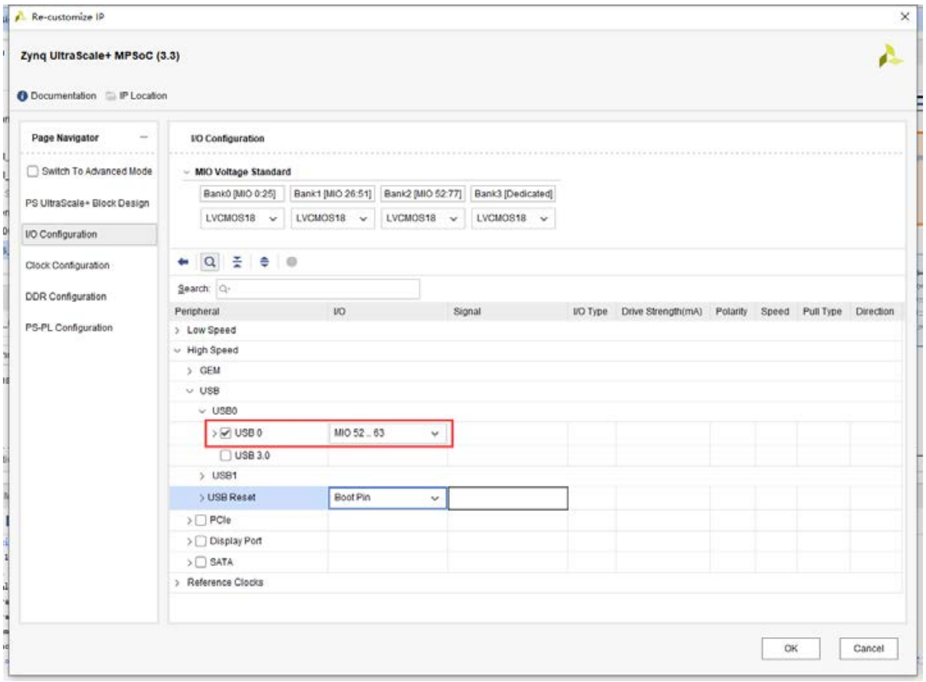
MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS33

Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
› Processing Unit								
⌄ High Speed								
⌄ GEM								
› □ GEM 0								
› □ GEM 1								
› □ GEM 2								
⌄ ☑ GEM 3	MIO 64 .. 75							
› ☑ MDIO 3	MIO 76 .. 77							
Gem 3	MIO64	rgmii_tx_clk	cmc	12	Def	slc	pullu	out
Gem 3	MIO65	rgmii_txd[0]	cmc	12	Def	slc	pullu	out
Gem 3	MIO66	rgmii_txd[1]	cmc	12	Def	slc	pullu	out
Gem 3	MIO67	rgmii_txd[2]	cmc	12	Def	slc	pullu	out
Gem 3	MIO68	rgmii_txd[3]	cmc	12	Def	slc	pullu	out

Check USB 0，select “MIO 52..63”，



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation IP Location

Page Navigator: □ Switch To Advanced Mode; PS UltraScale+ Block Design; I/O Configuration; Clock Configuration; DDR Configuration; PS-PL Configuration

I/O Configuration

MIO Voltage Standard

Bank0 [MIO 0:25]	Bank1 [MIO 26:51]	Bank2 [MIO 52:77]	Bank3 [Dedicated]
LVCMOS18	LVCMOS18	LVCMOS18	LVCMOS18

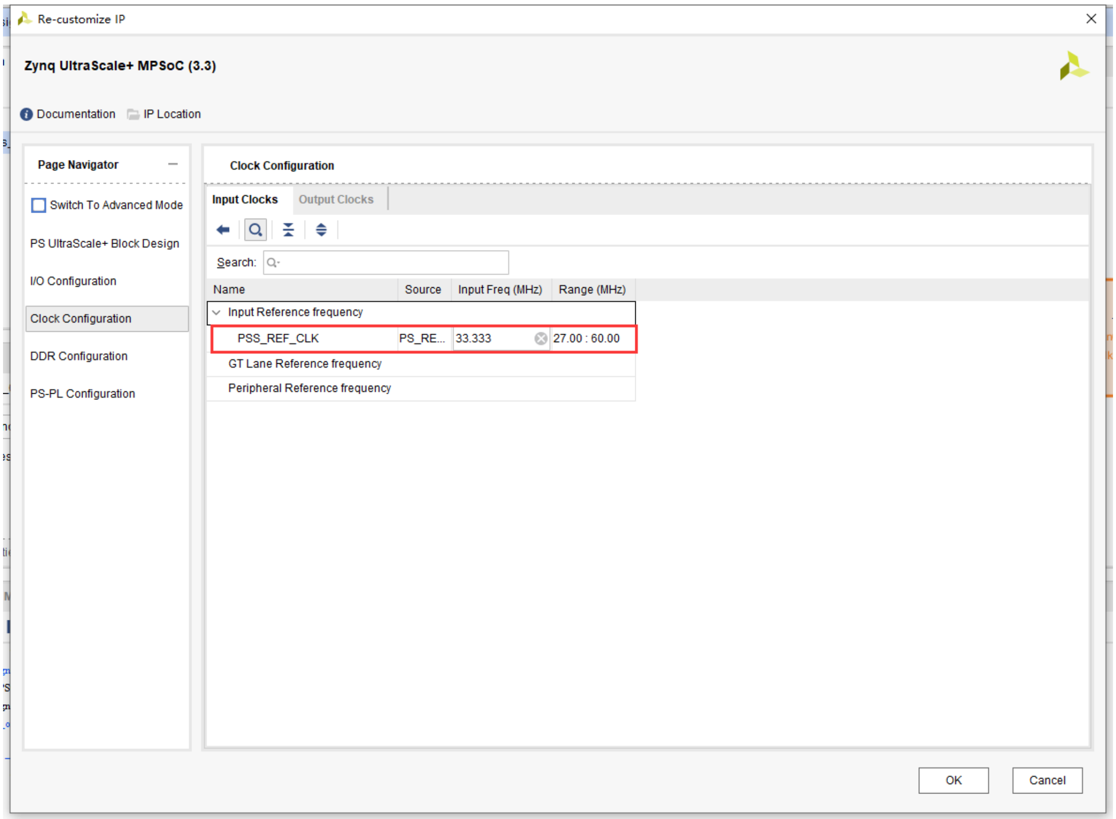
Search:

Peripheral	I/O	Signal	I/O Type	Drive Strength(mA)	Polarity	Speed	Pull Type	Direction
› Low Speed								
⌄ High Speed								
› GEM								
⌄ USB								
⌄ USB0								
› ☑ USB 0	MIO 52 .. 63							
□ USB 3.0								
› USB1								
› USB Reset	Boot Pin							
› □ PCIe								
› □ Display Port								
› □ SATA								
› Reference Clocks								

The IO part of the configuration is complete.

4．Clock configuration

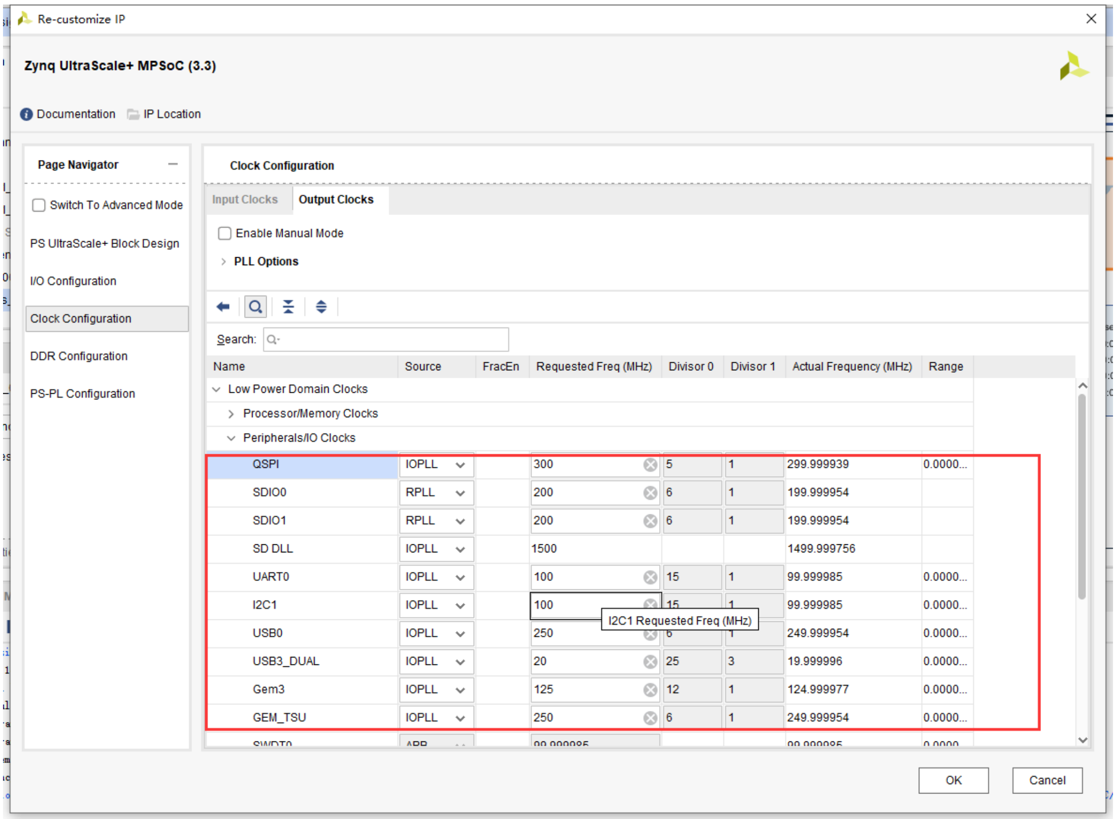
In the clock configuration window, configure the reference clock in the input clocks window, where PSS_ REF_ ClK is the reference clock of arm, which is 33.333MHz by default.



Clock Configuration

Name	Source	Input Freq (MHz)	Range (MHz)
Input Reference frequency			
PSS_REF_CLK	PS_RE...	33.333	27.00 : 60.00
GT Lane Reference frequency			
Peripheral Reference frequency			

In the Output Clocks window, the parameters are shown in the figure。



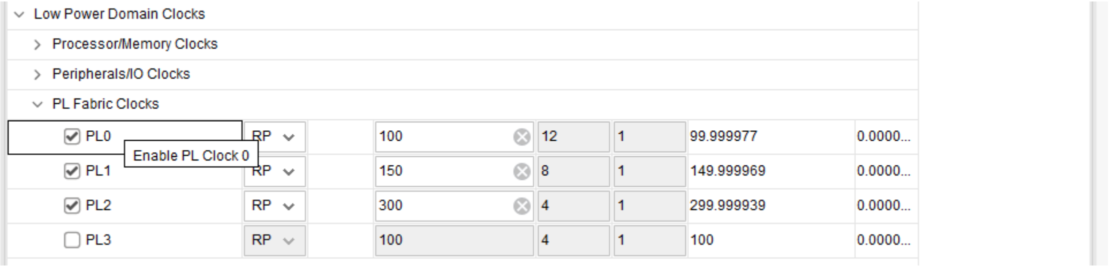
Clock Configuration

□ Enable Manual Mode

PLL Options

Name	Source	FracEn	Requested Freq (MHz)	Divisor 0	Divisor 1	Actual Frequency (MHz)	Range
Low Power Domain Clocks							
Processor/Memory Clocks							
Peripherals/IO Clocks							
QSPI	IOPLL		300	5	1	299.999939	0.0000...
SDIO0	RPLL		200	6	1	199.999954	
SDIO1	RPLL		200	6	1	199.999954	
SD DLL	IOPLL		1500			1499.999756	
UART0	IOPLL		100	15	1	99.999985	0.0000...
I2C1	IOPLL		100	15	1	99.999985	0.0000...
USB0	IOPLL		250	6	1	249.999954	0.0000...
USB3_DUAL	IOPLL		20	25	3	19.999996	0.0000...
Gem3	IOPLL		125	12	1	124.999977	0.0000...
GEM_TSU	IOPLL		250	6	1	249.999954	0.0000...
SWDT0	APB		99.999985			99.999985	0.0000

The clock configuration of PL is shown in the figure below.



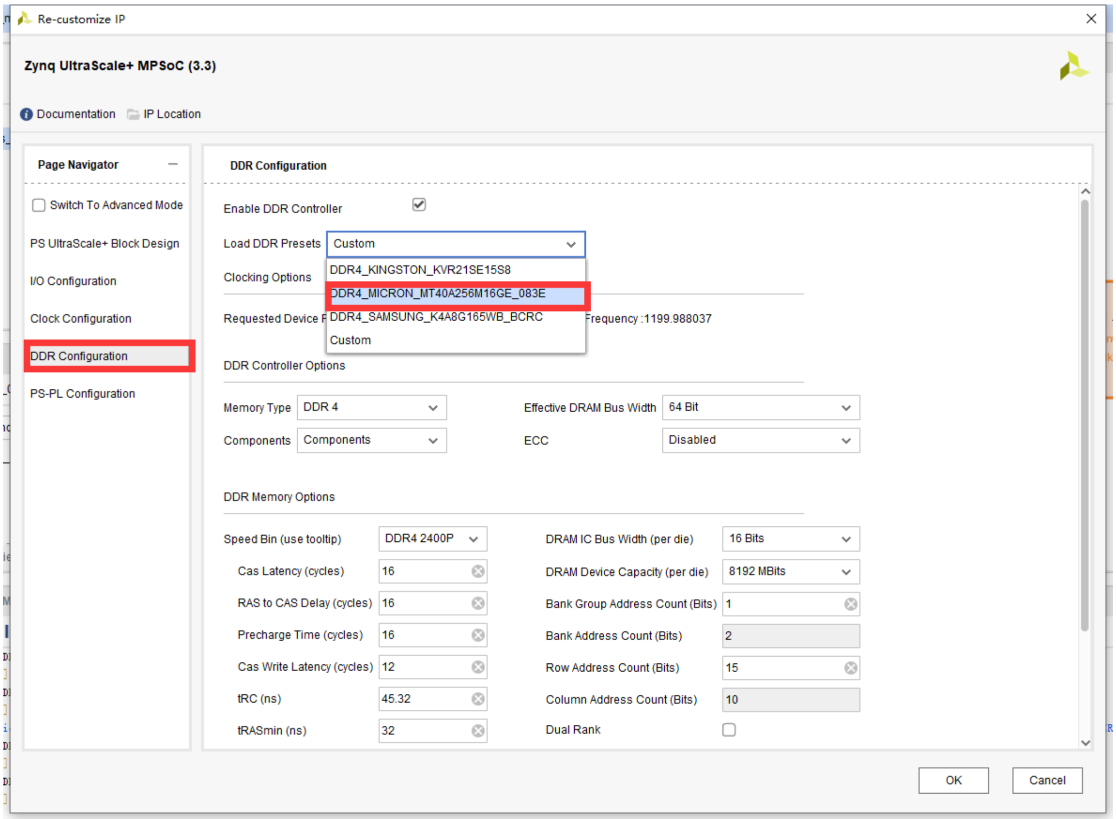
Low Power Domain Clocks
Processor/Memory Clocks
Peripherals/IO Clocks
PL Fabric Clocks

☑ PL0	RP	100	12	1	99.999977	0.0000...
☑ PL1	RP	150	8	1	149.999969	0.0000...
☑ PL2	RP	300	4	1	299.999939	0.0000...
□ PL3	RP	100	4	1	100	0.0000...

The other parts remain the default, so far, the clock part is configured

5. DDR configuration

In the DDR configuration window， select ”DDR4_MICRON_MT40A256M16GE_083E” for “Load DDR Presets”



DDR Configuration

Enable DDR Controller ☑
Load DDR Presets: Custom
- DDR4_KINGSTON_KVR21SE15S8
- DDR4_MICRON_MT40A256M16GE_083E
- DDR4_SAMSUNG_K4A8G165WB_BCRC
- Custom

Clocking Options
Requested Device F... Frequency :1199.988037

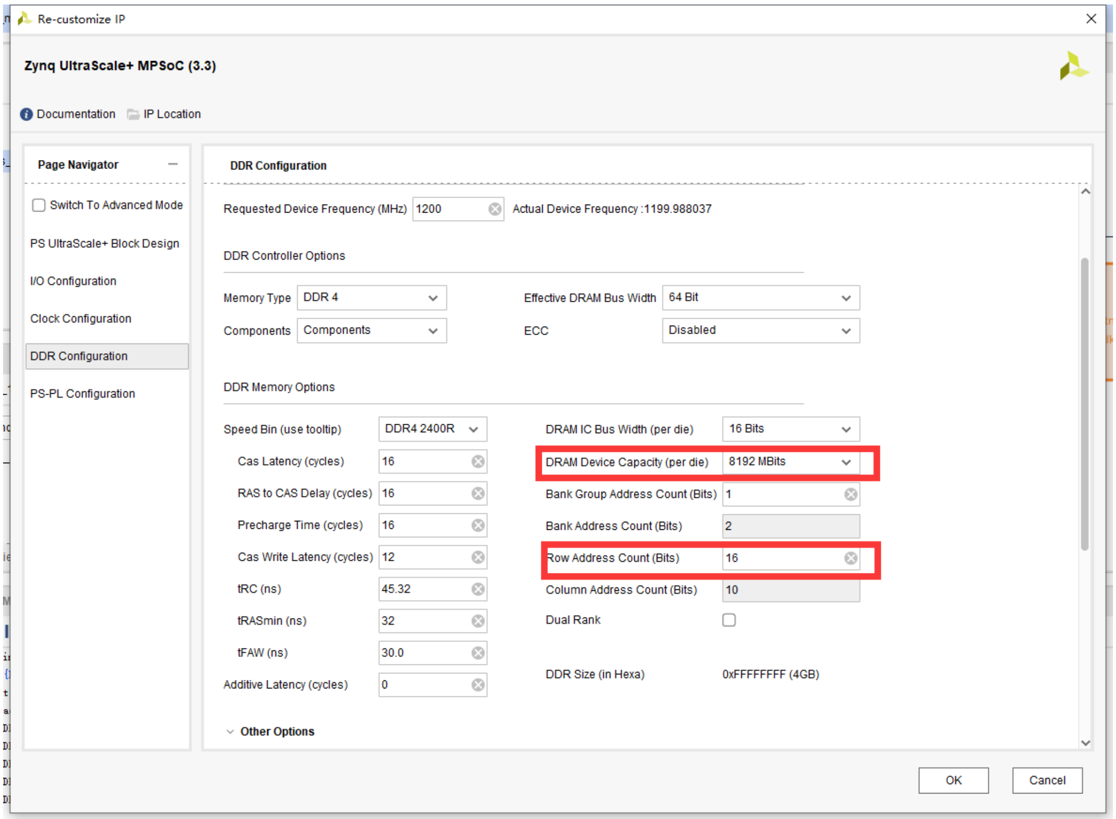
DDR Controller Options

Memory Type: DDR 4
Components: Components
Effective DRAM Bus Width: 64 Bit
ECC: Disabled

DDR Memory Options

Speed Bin (use tooltip): DDR4 2400P
Cas Latency (cycles): 16
RAS to CAS Delay (cycles): 16
Precharge Time (cycles): 16
Cas Write Latency (cycles): 12
tRC (ns): 45.32
tRASmin (ns): 32
DRAM IC Bus Width (per die): 16 Bits
DRAM Device Capacity (per die): 8192 MBits
Bank Group Address Count (Bits): 1
Bank Address Count (Bits): 2
Row Address Count (Bits): 15
Column Address Count (Bits): 10
Dual Rank □

The parameters are modified as follows，



Re-customize IP

Zynq UltraScale+ MPSoC (3.3)

Documentation IP Location

Page Navigator
- □ Switch To Advanced Mode
- PS UltraScale+ Block Design
- I/O Configuration
- Clock Configuration
- DDR Configuration
- PS-PL Configuration

DDR Configuration

Requested Device Frequency (MHz) 1200 Actual Device Frequency :1199.988037

DDR Controller Options

Memory Type: DDR 4 Effective DRAM Bus Width: 64 Bit
Components: Components ECC: Disabled

DDR Memory Options

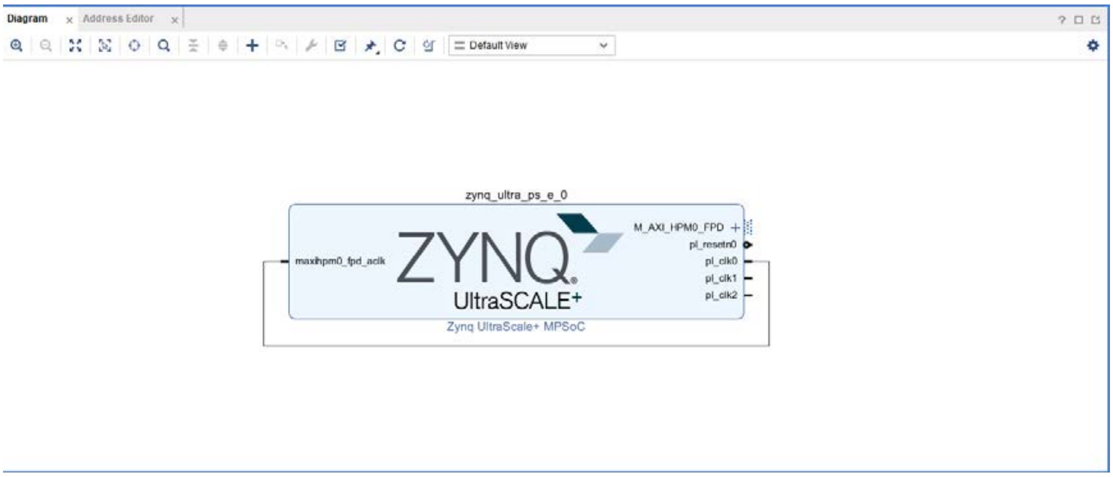
Speed Bin (use tooltip)	DDR4 2400R	DRAM IC Bus Width (per die)	16 Bits
Cas Latency (cycles)	16	DRAM Device Capacity (per die)	8192 MBits
RAS to CAS Delay (cycles)	16	Bank Group Address Count (Bits)	1
Precharge Time (cycles)	16	Bank Address Count (Bits)	2
Cas Write Latency (cycles)	12	Row Address Count (Bits)	16
tRC (ns)	45.32	Column Address Count (Bits)	10
tRASmin (ns)	32	Dual Rank	□
tFAW (ns)	30.0		
Additive Latency (cycles)	0	DDR Size (in Hexa)	0xFFFFFFFF (4GB)

Other Options

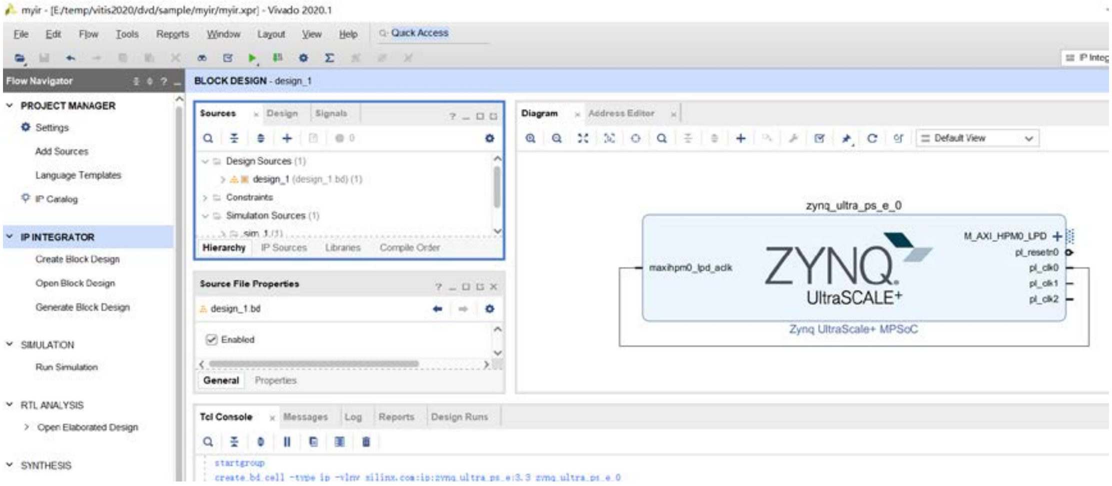
OK Cancel

6. PS-PL configuration

Keep the default, click OK, the configuration is complete, and connect to “pl_clk0” and “maximhpm0_ lpd_ aclk”, the final result is shown in the figure below.

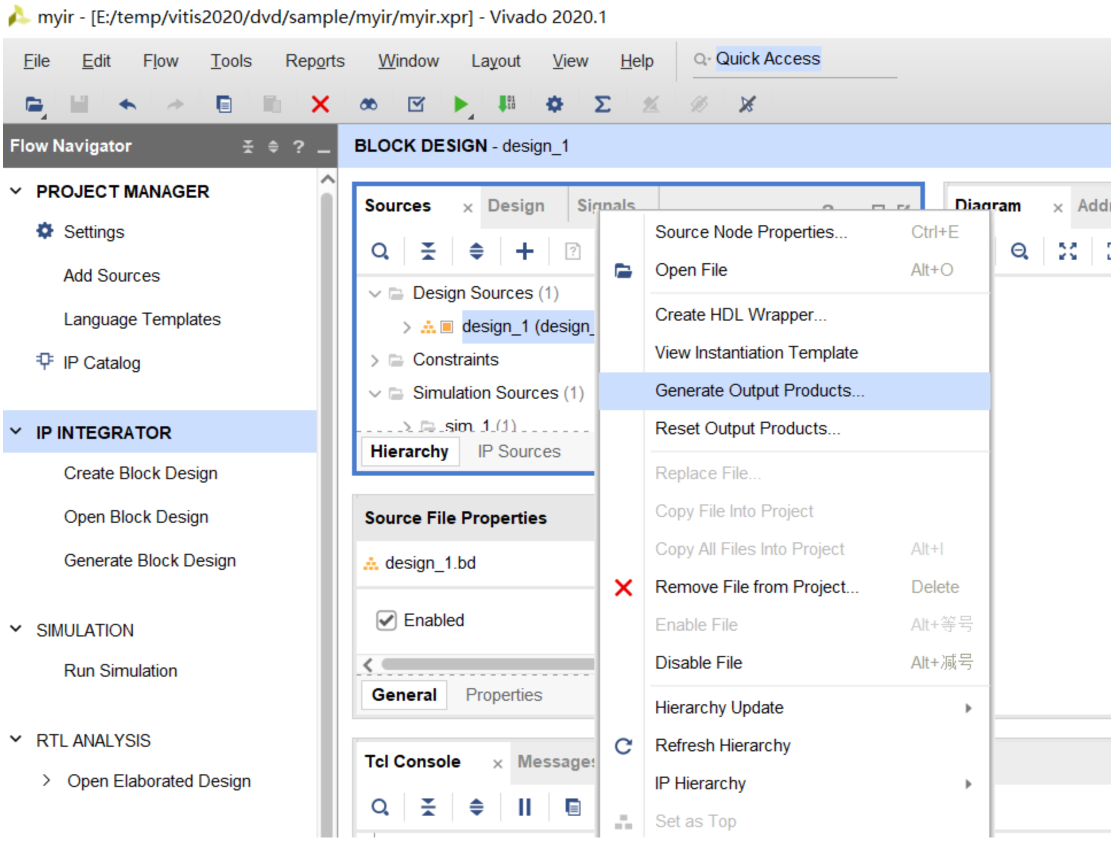


Click sources, as shown in the following figure

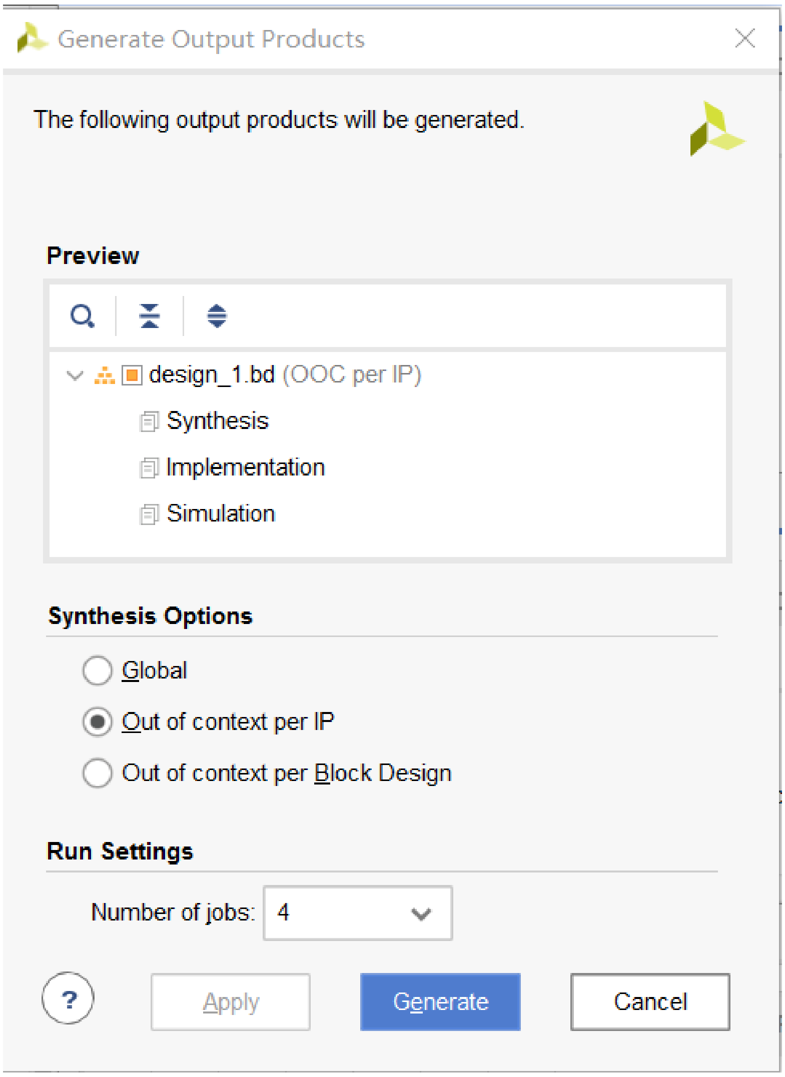


```
startgroup
create_bd_cell -type ip -vlnv xilinx.com:ip:zynq_ultra_ps_e:3.3 zynq_ultra_ps_e_0
```

Right click design_1-->Generate Output Products



Then click Generate



Generate Output Products

The following output products will be generated.

Preview

- design_1.bd (OOC per IP)
  - Synthesis
  - Implementation
  - Simulation

Synthesis Options

○ Global

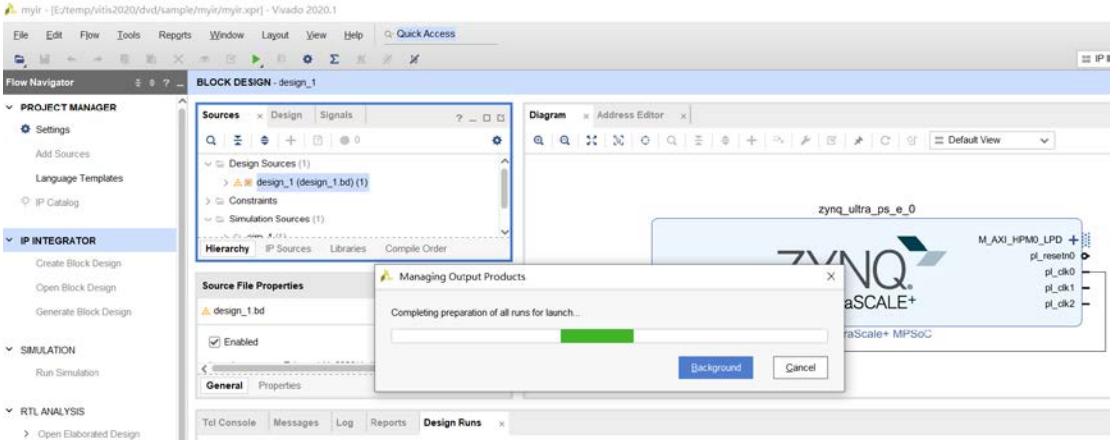
● Out of context per IP

○ Out of context per Block Design

Run Settings

Number of jobs: 4

Generating in progress

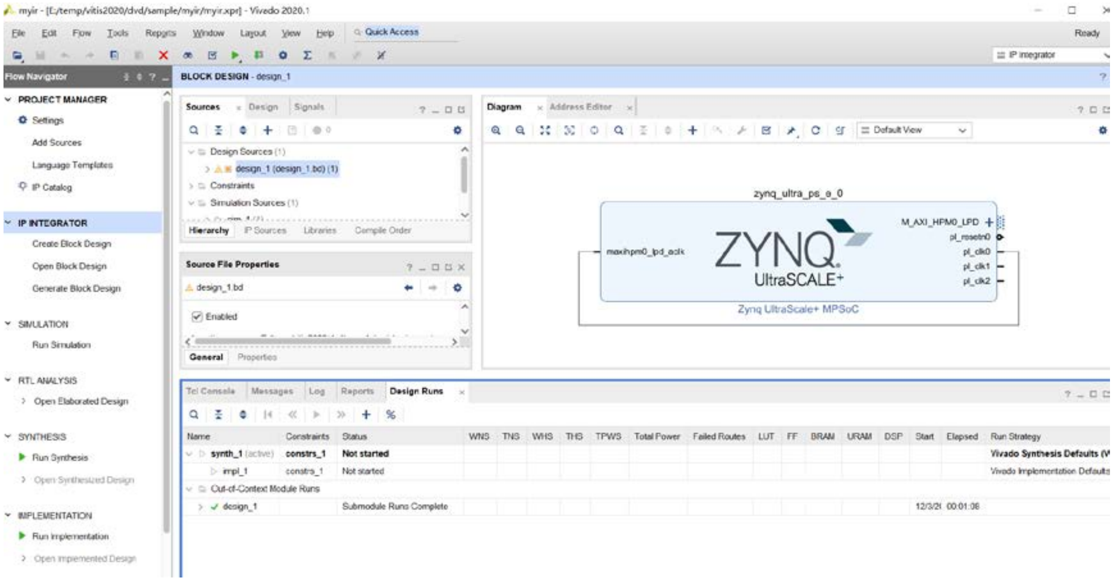


myir - [E:/temp/vitis2020/dvd/sample/myir/myir.xpr] - Vivado 2020.1

Managing Output Products

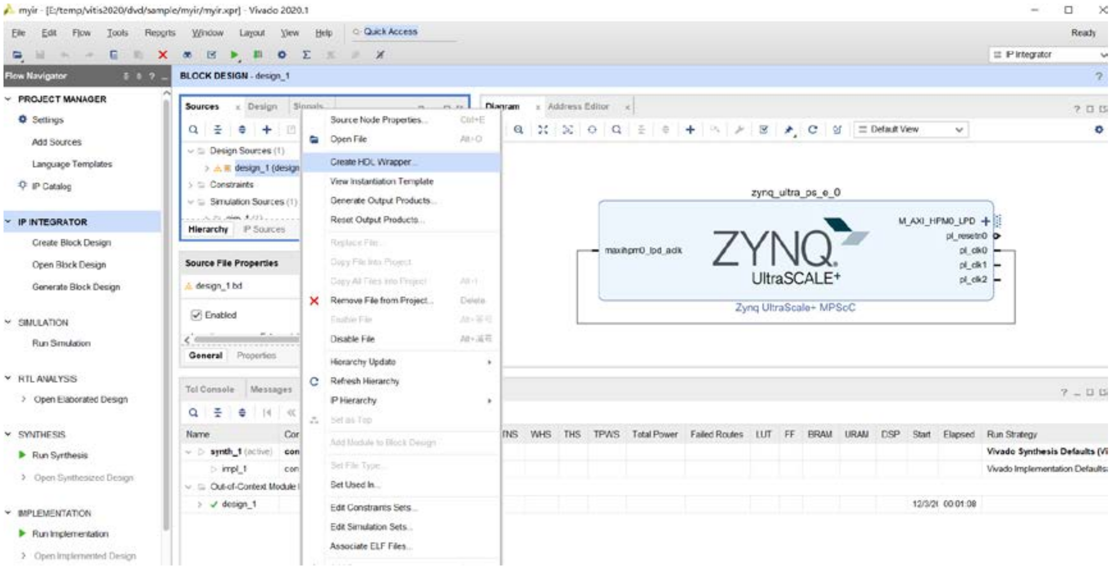
Completing preparation of all runs for launch...

After the synthesis is completed, the following figure is shown.

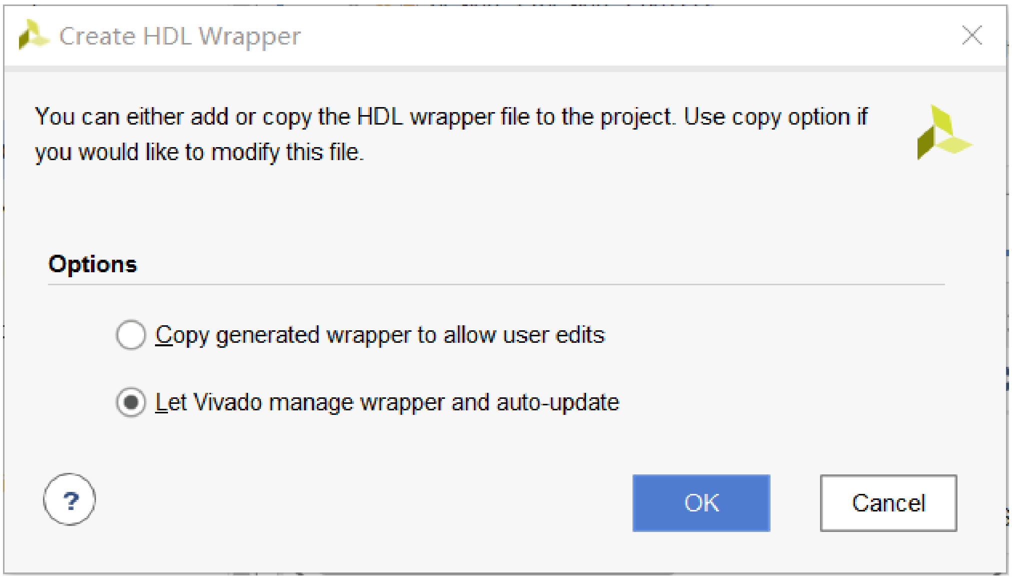


Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1 (active)	constrs_1	Not started															Vivado Synthesis Defaults (V
impl_1	constrs_1	Not started															Vivado Implementation Defaults
Out-of-Context Module Runs																	
design_1		Submodule Runs Complete													12/3/2(	00:01:08	

Right-click design_1 and select Create HDL Wrapper to generate the top-level file



Click OK



Create HDL Wrapper

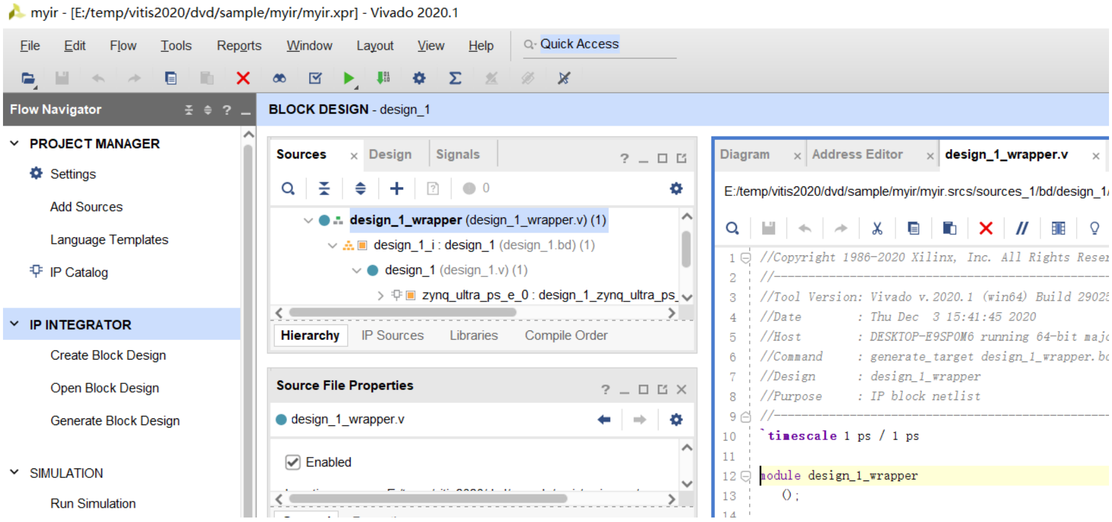
You can either add or copy the HDL wrapper file to the project. Use copy option if you would like to modify this file.

Options

○ Copy generated wrapper to allow user edits

● Let Vivado manage wrapper and auto-update

Automatic generation of top-level file design_1_warpper.v



myir - [E:/temp/vitis2020/dvd/sample/myir/myir.xpr] - Vivado 2020.1

design_1_wrapper (design_1_wrapper.v) (1)
design_1_i : design_1 (design_1.bd) (1)
design_1 (design_1.v) (1)
zynq_ultra_ps_e_0 : design_1_zynq_ultra_ps_

design_1_wrapper.v

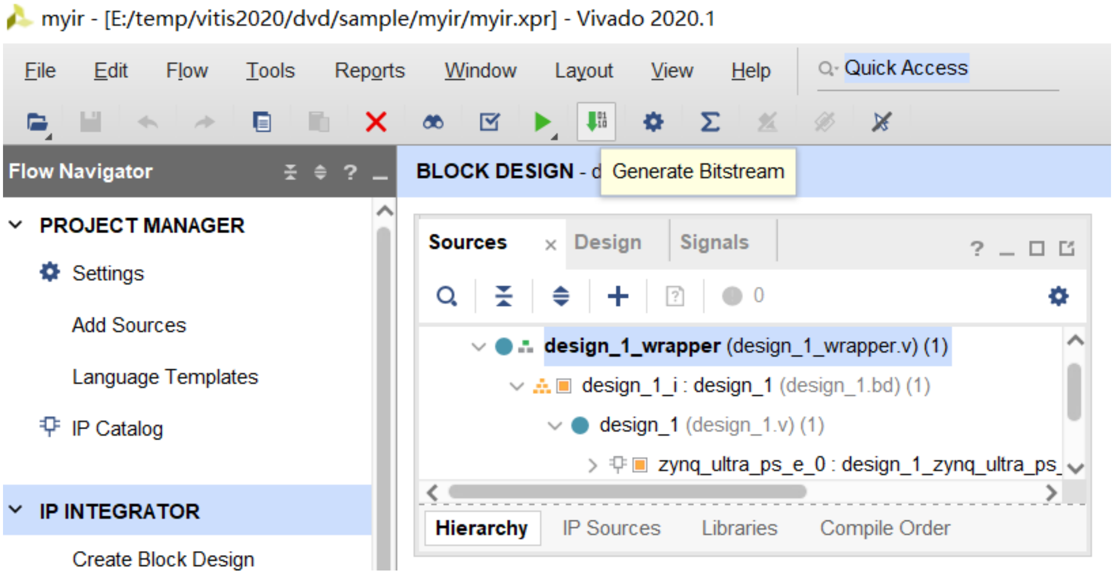
☑ Enabled

E:/temp/vitis2020/dvd/sample/myir/myir.srcs/sources_1/bd/design_1/

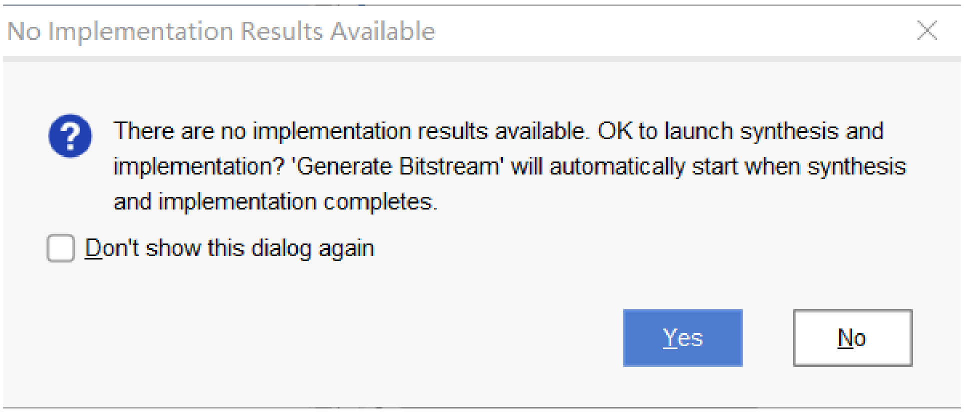
```
//Copyright 1986-2020 Xilinx, Inc. All Rights Reser
//---------------------------------------------------
//Tool Version: Vivado v.2020.1 (win64) Build 2902
//Date        : Thu Dec  3 15:41:45 2020
//Host        : DESKTOP-E9SP0M6 running 64-bit majc
//Command     : generate_target design_1_wrapper.bd
//Design      : design_1_wrapper
//Purpose     : IP block netlist
//---------------------------------------------------
`timescale 1 ps / 1 ps

module design_1_wrapper
   ();
```

Click Generating Bitstream to generate binary files



Click Yes in the dialog box that appears

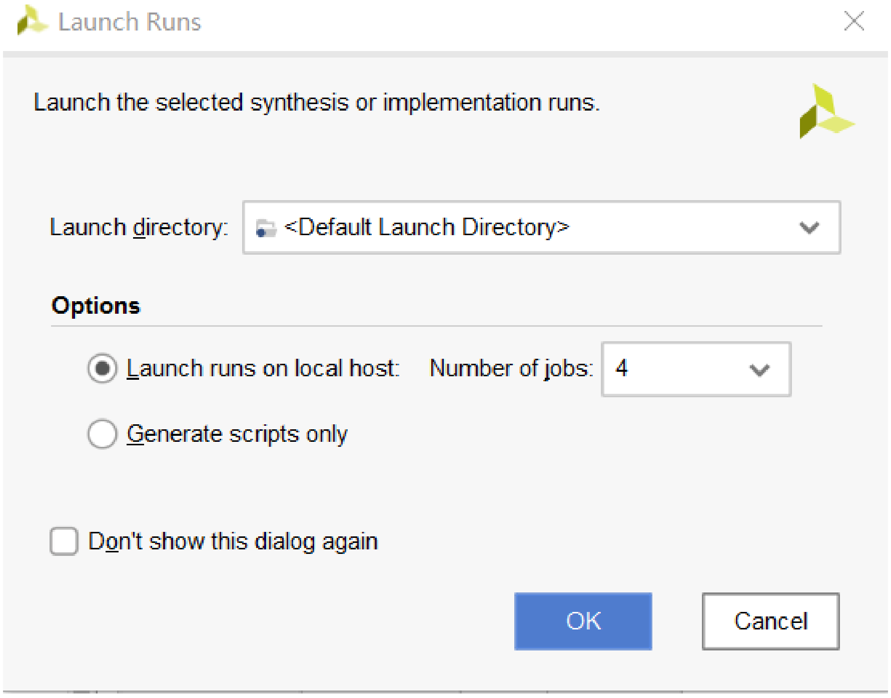


No Implementation Results Available

There are no implementation results available. OK to launch synthesis and implementation? 'Generate Bitstream' will automatically start when synthesis and implementation completes.

□ Don't show this dialog again

Click OK



Launch Runs

Launch the selected synthesis or implementation runs.

Launch directory: <Default Launch Directory>

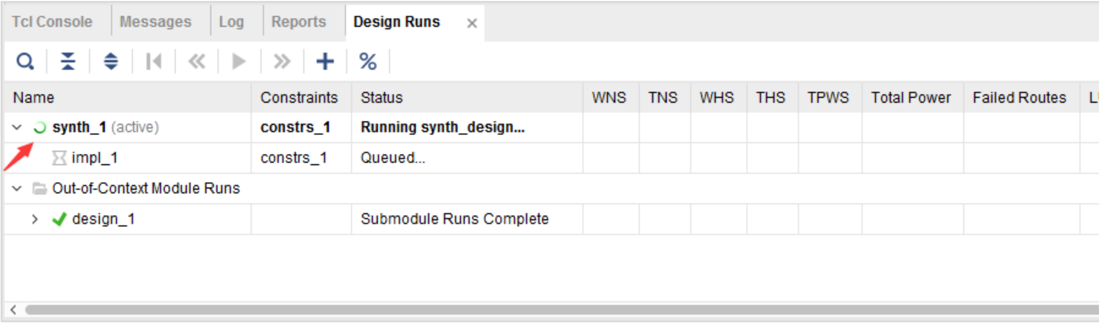
Options

● Launch runs on local host: Number of jobs: 4

○ Generate scripts only

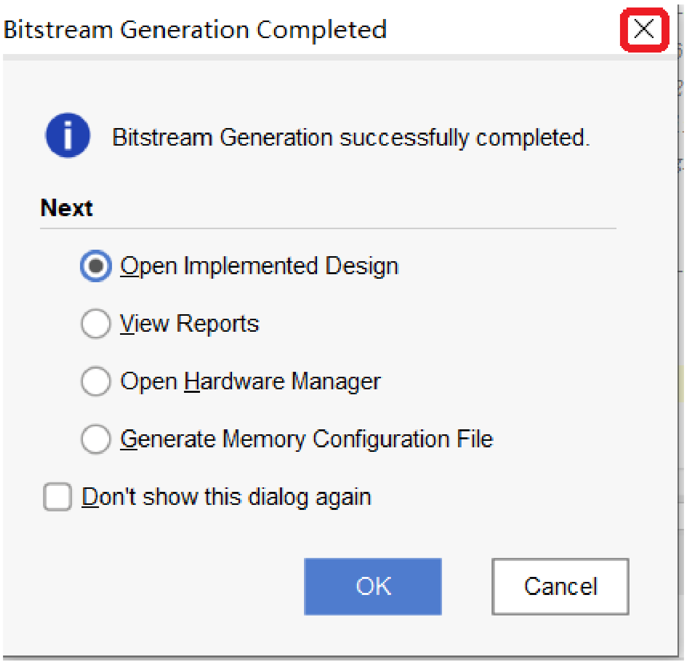
□ Don't show this dialog again

Binary file bit being generated



Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	L
synth_1 (active)	constrs_1	Running synth_design...								
impl_1	constrs_1	Queued...								
Out-of-Context Module Runs										
✓ design_1		Submodule Runs Complete								

Binary file bit generation completed, close the pop-up dialog box



Bitstream Generation Completed

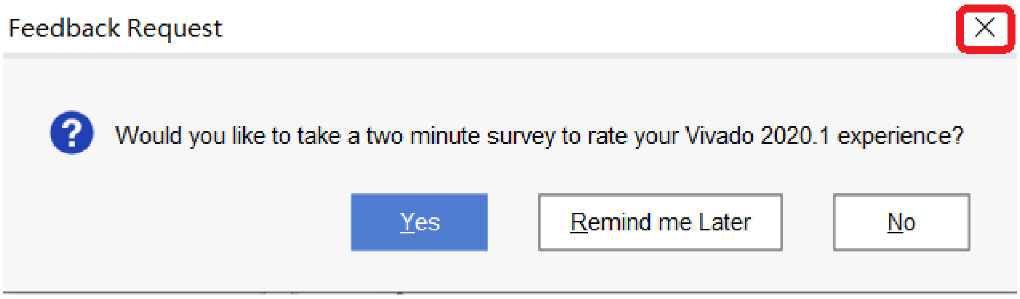
Bitstream Generation successfully completed.

Next

- ● Open Implemented Design
- ○ View Reports
- ○ Open Hardware Manager
- ○ Generate Memory Configuration File

□ Don't show this dialog again

Continue close the next pop-up dialog box

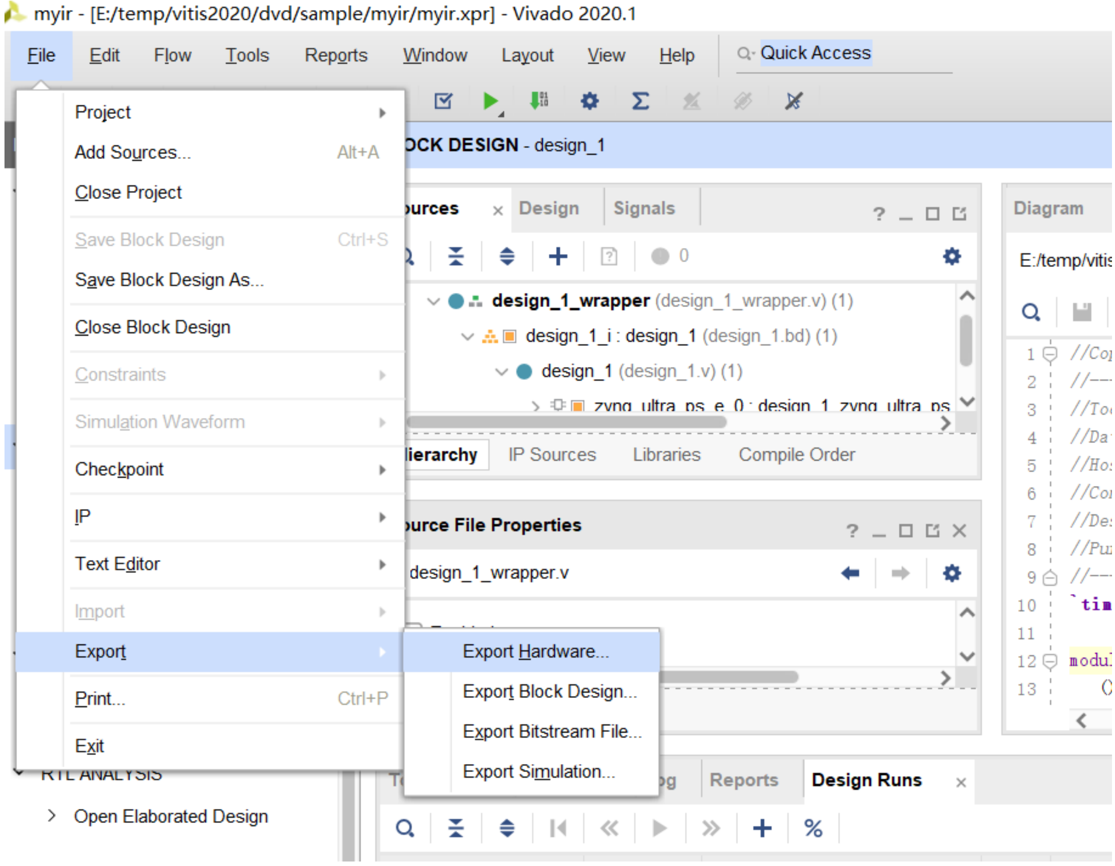


Feedback Request

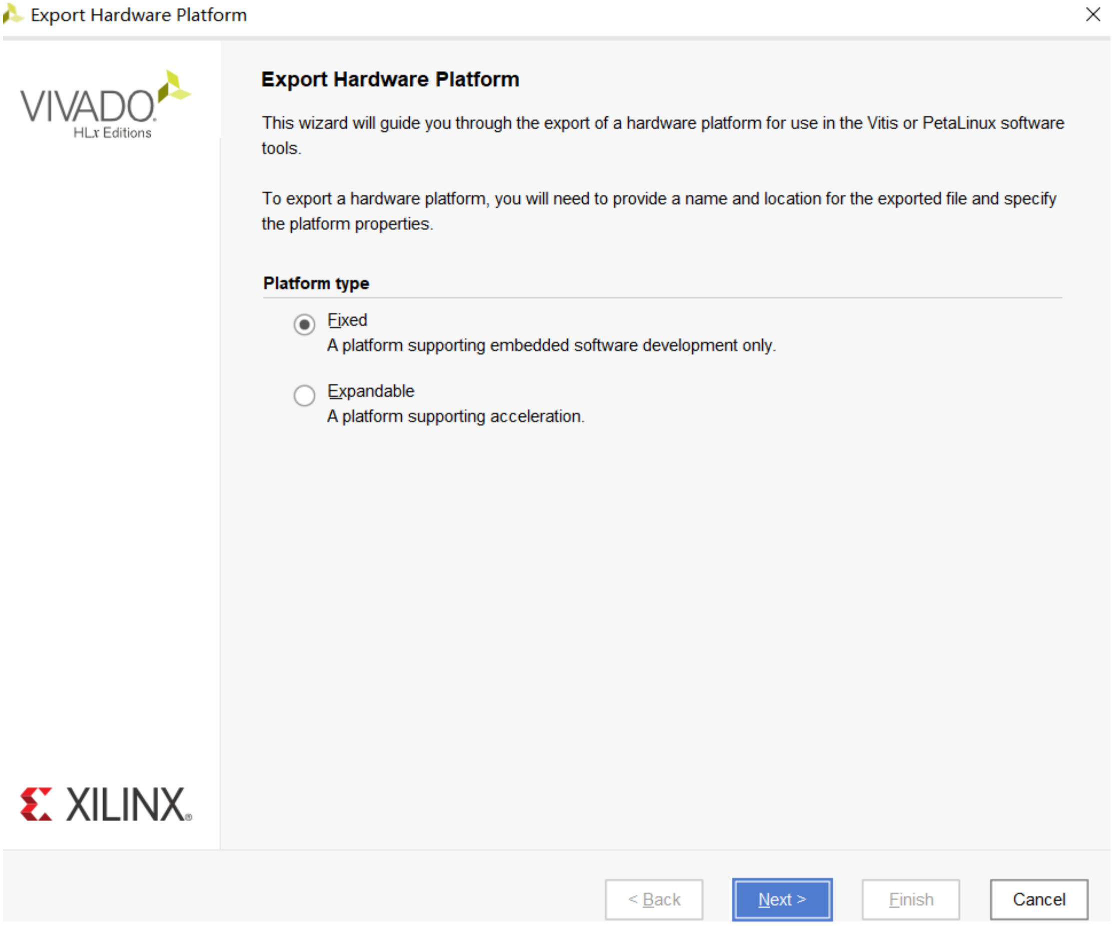
Would you like to take a two minute survey to rate your Vivado 2020.1 experience?

5 VIVADO export hardware platform to VITIS

SelectFile > Export > Export Hardware



Click Next



Export Hardware Platform

Export Hardware Platform

This wizard will guide you through the export of a hardware platform for use in the Vitis or PetaLinux software tools.

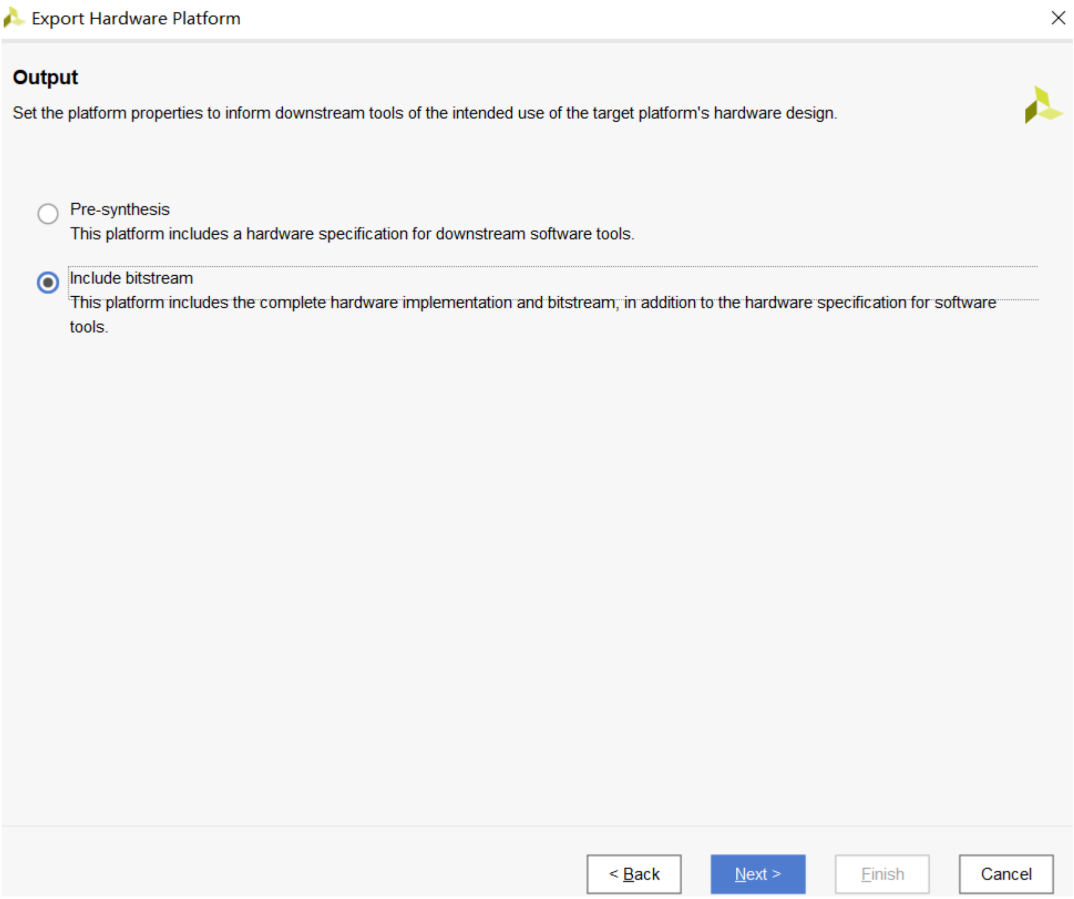
To export a hardware platform, you will need to provide a name and location for the exported file and specify the platform properties.

Platform type

● Fixed
A platform supporting embedded software development only.

○ Expandable
A platform supporting acceleration.

Check Include bitstream and click Next



Export Hardware Platform

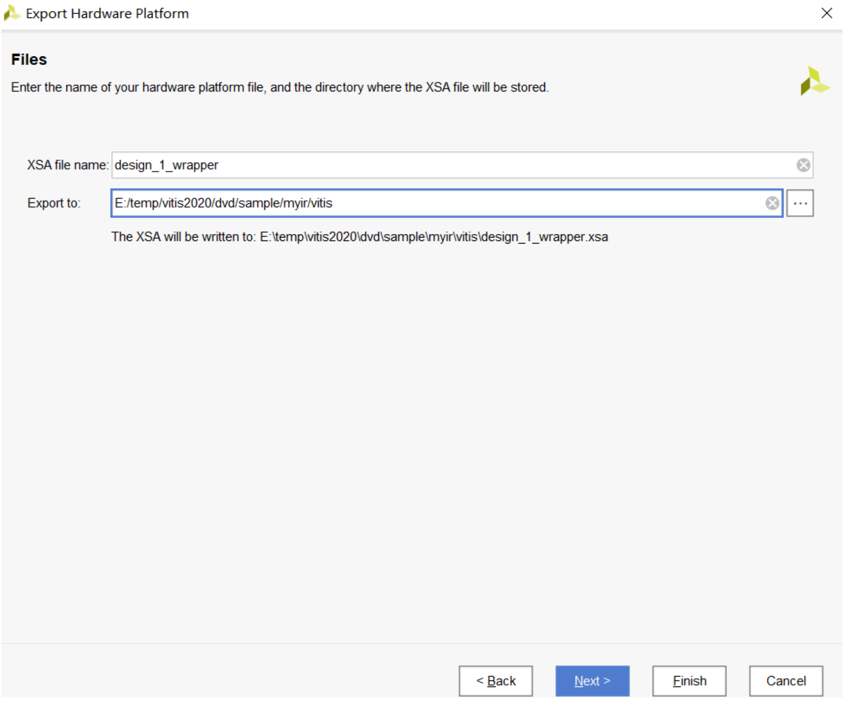
Output

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.

Then select the directory where the hardware platform files are stored and click Next



Export Hardware Platform

Files

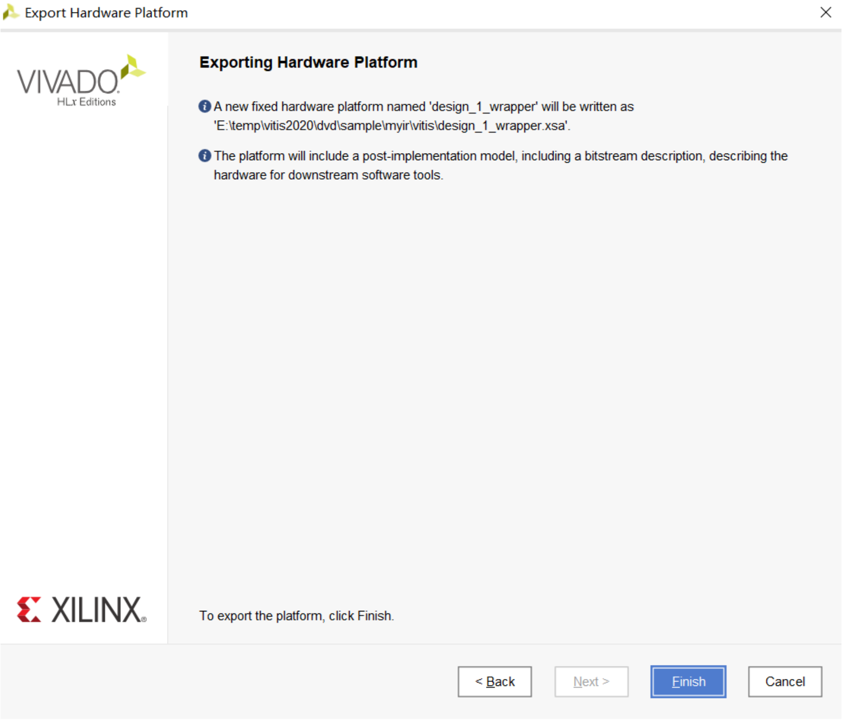
Enter the name of your hardware platform file, and the directory where the XSA file will be stored.

XSA file name: design_1_wrapper

Export to: E:/temp/vitis2020/dvd/sample/myir/vitis

The XSA will be written to: E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper.xsa

Then click Finish to generate the “design_1_wrapper.xsa” file



Exporting Hardware Platform

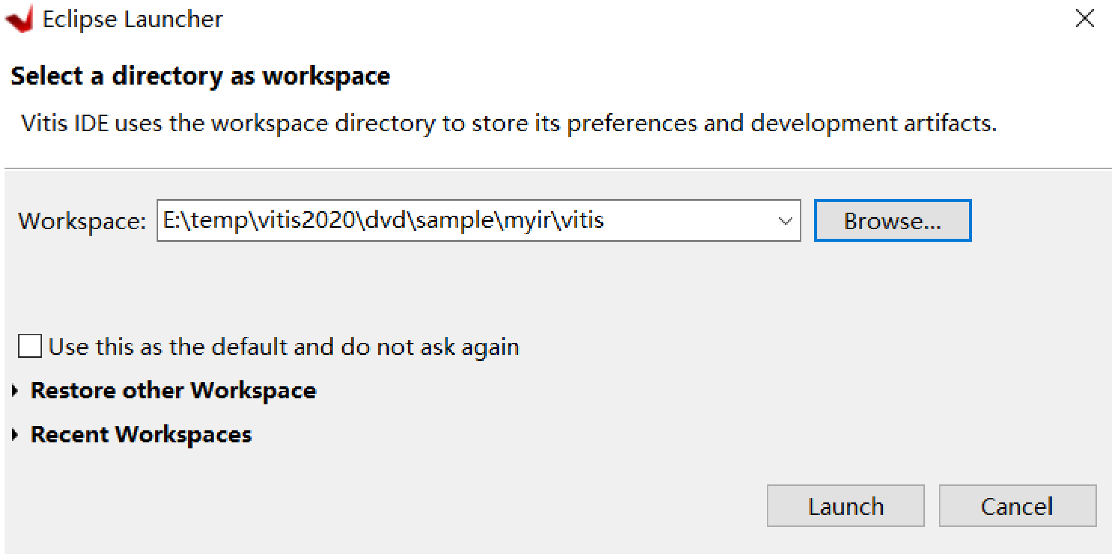
A new fixed hardware platform named 'design_1_wrapper' will be written as 'E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper.xsa'.

The platform will include a post-implementation model, including a bitstream description, describing the hardware for downstream software tools.

To export the platform, click Finish.

5. 1 VITIS create an hello project

Open Xilinx vitis2020.1, select the following directory to generate BSP of the platform, and click launch.



Eclipse Launcher

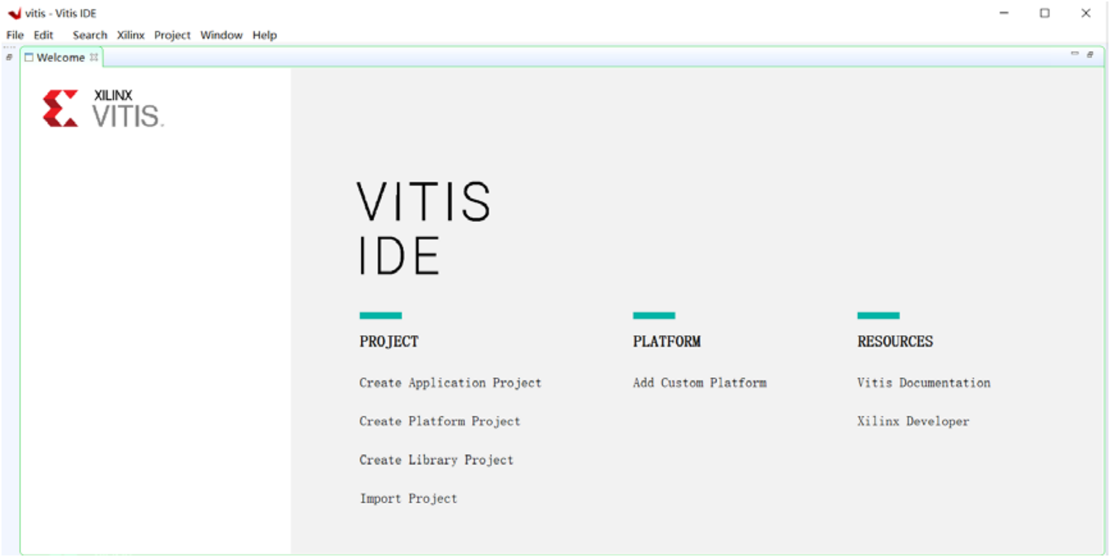
Select a directory as workspace

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

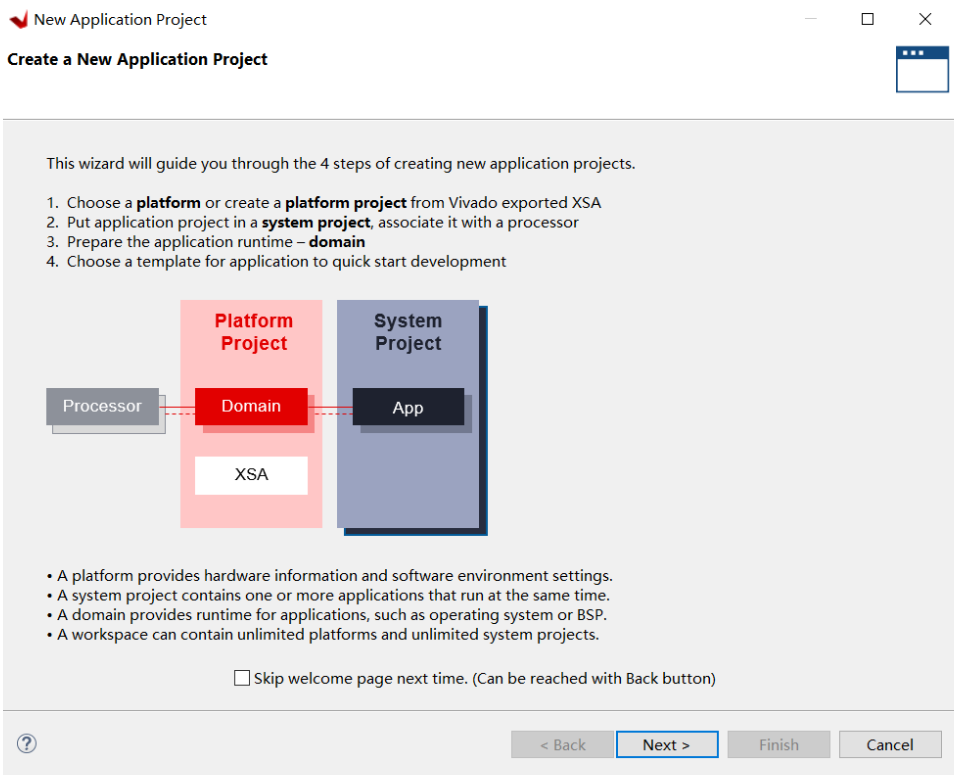
Workspace: E:\temp\vitis2020\dvd\sample\myir\vitis

□ Use this as the default and do not ask again

The following window appears, click Create Application Project



Click Next



New Application Project

Create a New Application Project

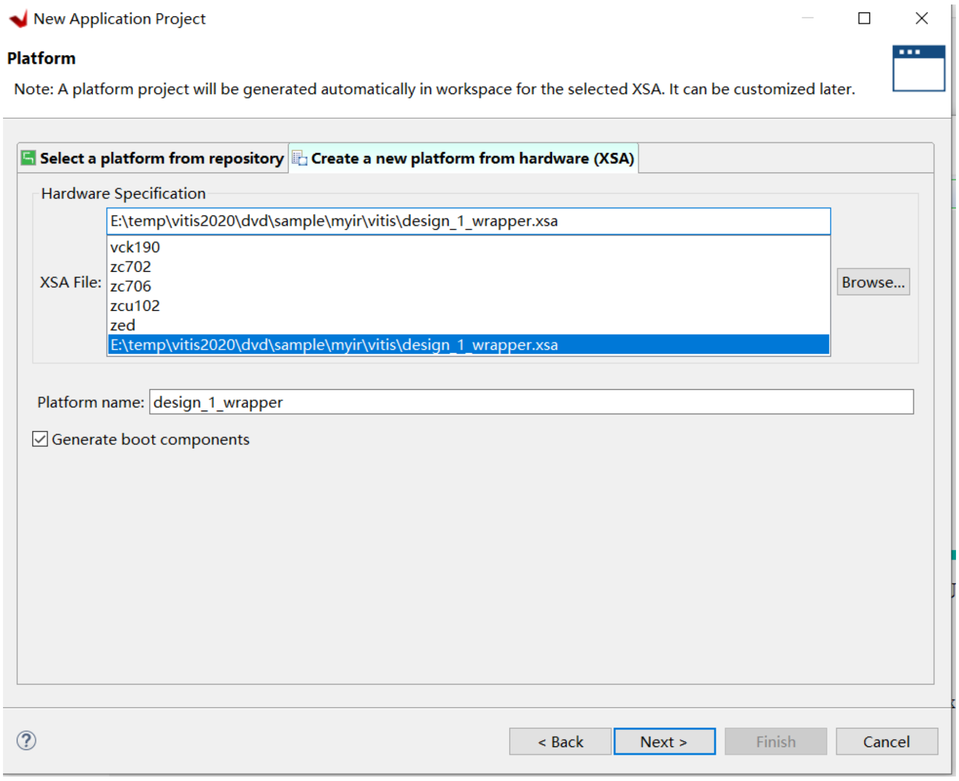
This wizard will guide you through the 4 steps of creating new application projects.

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.
- A system project contains one or more applications that run at the same time.
- A domain provides runtime for applications, such as operating system or BSP.
- A workspace can contain unlimited platforms and unlimited system projects.

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

Then select the just generated design_1_wrapper.xsa file，click Next。



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

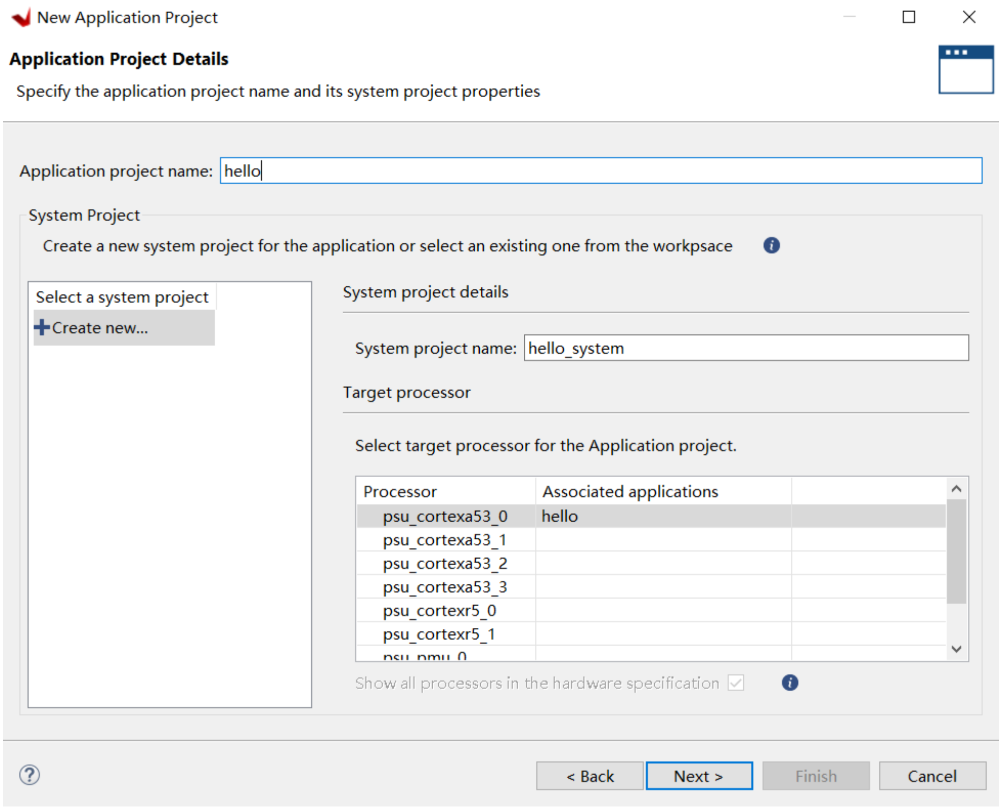
XSA File: E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper.xsa

vck190
zc702
zc706
zcu102
zed
E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper.xsa

Platform name: design_1_wrapper

☑ Generate boot components

Then entry “hello”，click Next



New Application Project

Application Project Details

Specify the application project name and its system project properties

Application project name: hello

System Project

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

Select a system project
+Create new...

System project details

System project name: hello_system

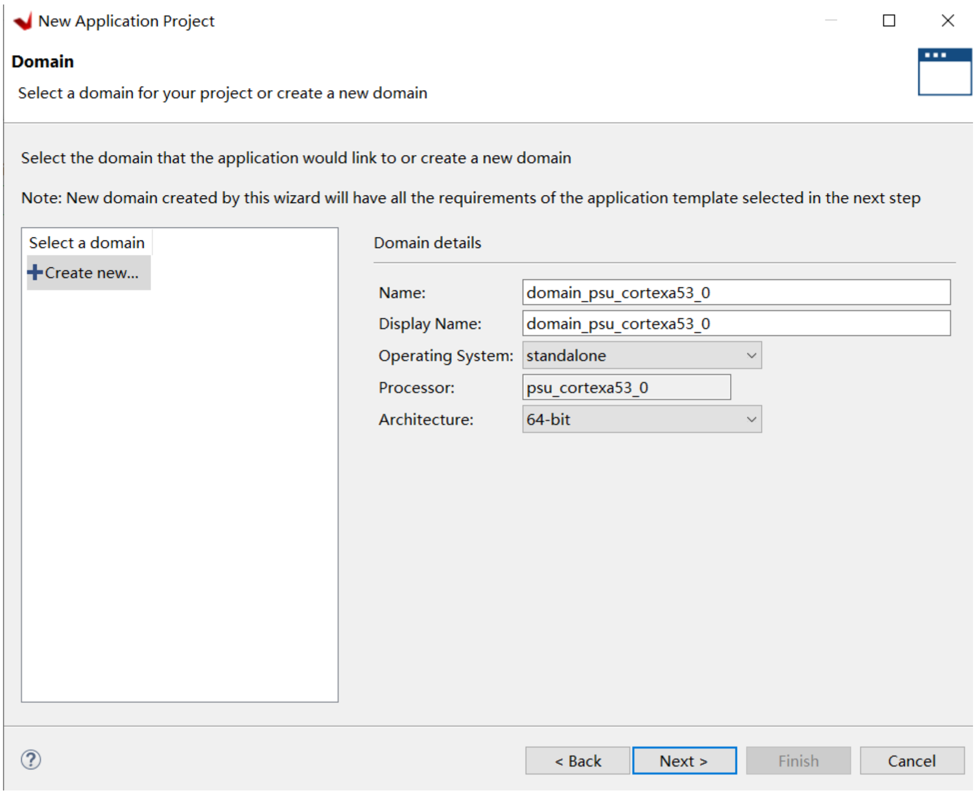
Target processor

Select target processor for the Application project.

Processor	Associated applications
psu_cortexa53_0	hello
psu_cortexa53_1	
psu_cortexa53_2	
psu_cortexa53_3	
psu_cortexr5_0	
psu_cortexr5_1	
psu_pmu_0	

Show all processors in the hardware specification ☑

Click Next



New Application Project

Domain

Select a domain for your project or create a new domain

Select the domain that the application would link to or create a new domain

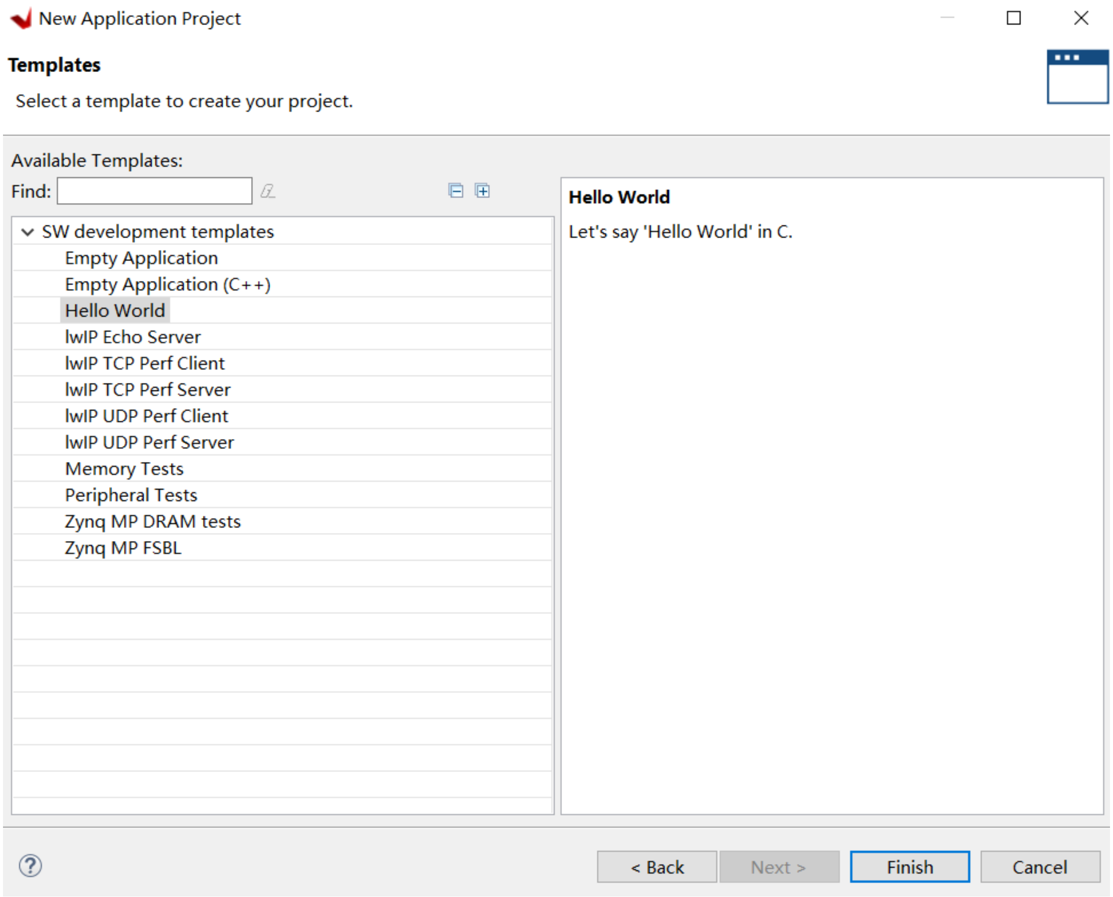
Note: New domain created by this wizard will have all the requirements of the application template selected in the next step

Select a domain
+Create new...

Domain details

Name: domain_psu_cortexa53_0
Display Name: domain_psu_cortexa53_0
Operating System: standalone
Processor: psu_cortexa53_0
Architecture: 64-bit

Click Finish，The vitis IDE interface is displayed.



New Application Project

Templates

Select a template to create your project.

Available Templates:

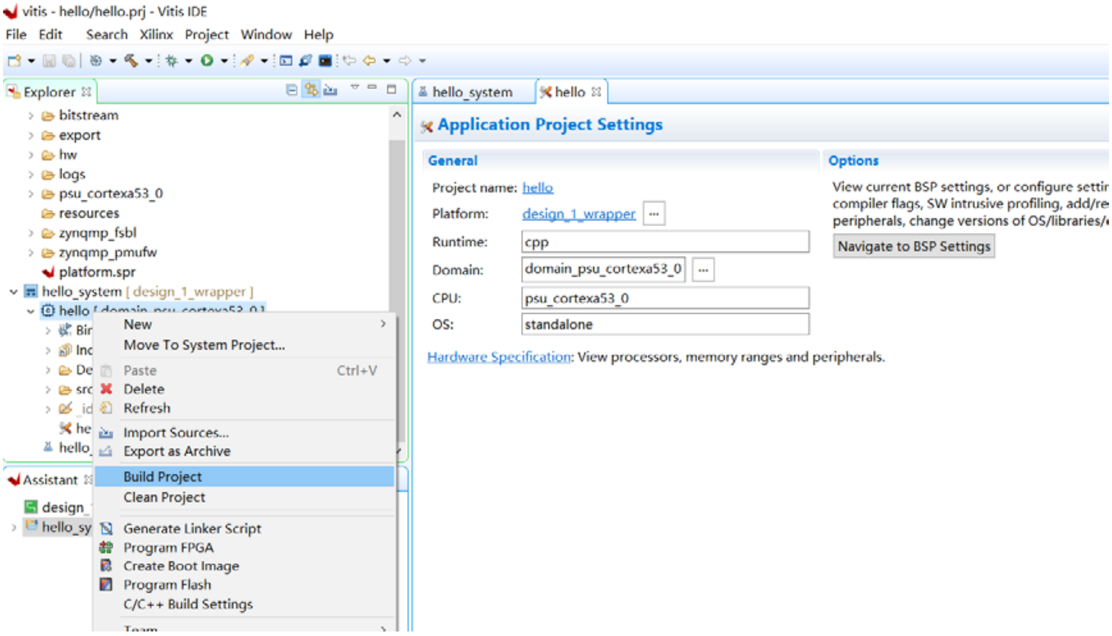
Find:

- SW development templates
  - Empty Application
  - Empty Application (C++)
  - Hello World
  - lwIP Echo Server
  - lwIP TCP Perf Client
  - lwIP TCP Perf Server
  - lwIP UDP Perf Client
  - lwIP UDP Perf Server
  - Memory Tests
  - Peripheral Tests
  - Zynq MP DRAM tests
  - Zynq MP FSBL

Hello World

Let's say 'Hello World' in C.

In vitis IDE，click “Build project” ，Compile the program and wait for the program to compile.



vitis - hello/hello.prj - Vitis IDE

Application Project Settings

General

Project name: hello

Platform: design_1_wrapper

Runtime: cpp

Domain: domain_psu_cortexa53_0

CPU: psu_cortexa53_0

OS: standalone

Hardware Specification: View processors, memory ranges and peripherals.

Options

View current BSP settings, or configure settir compiler flags, SW intrusive profiling, add/re peripherals, change versions of OS/libraries/

6Start-up Equipment

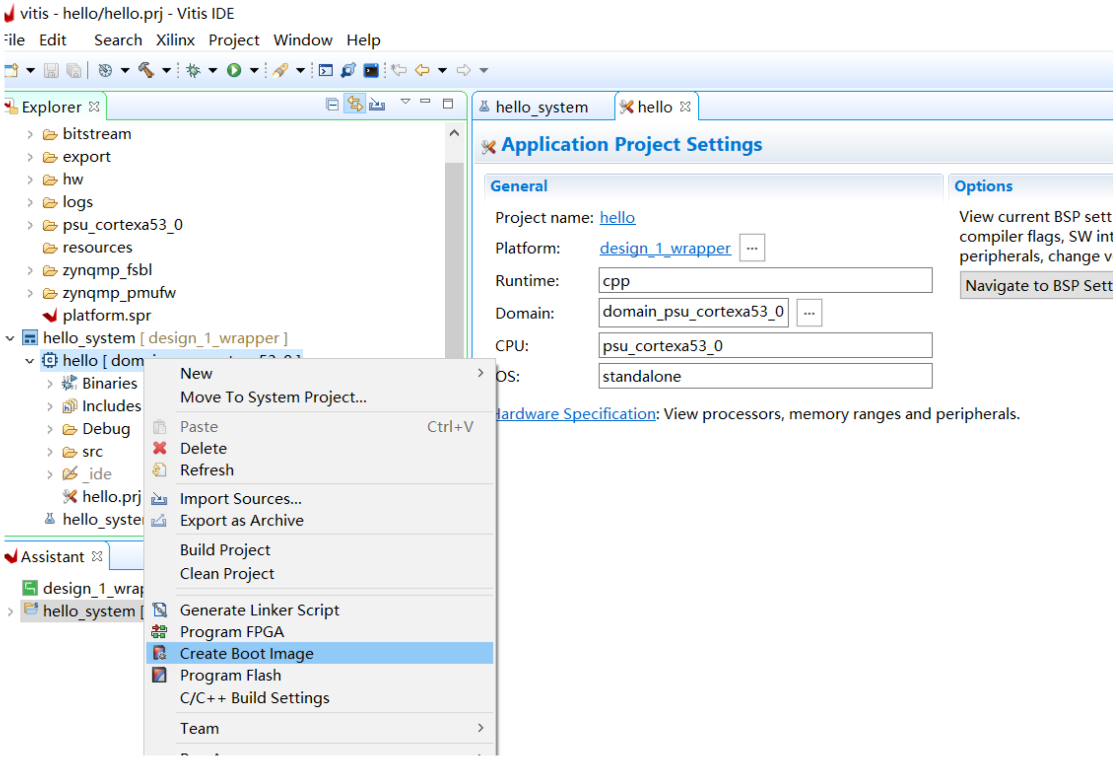
In this chapter, we introduce how to start the development board through SD card, and load the application into MYD-3EG4EV DDR4.

6. 1Boot from MicroSD Card

•Start from the microSD card, generate the boot image and store it on the microSD card.

6.1.1Generating SD Card Startup Image

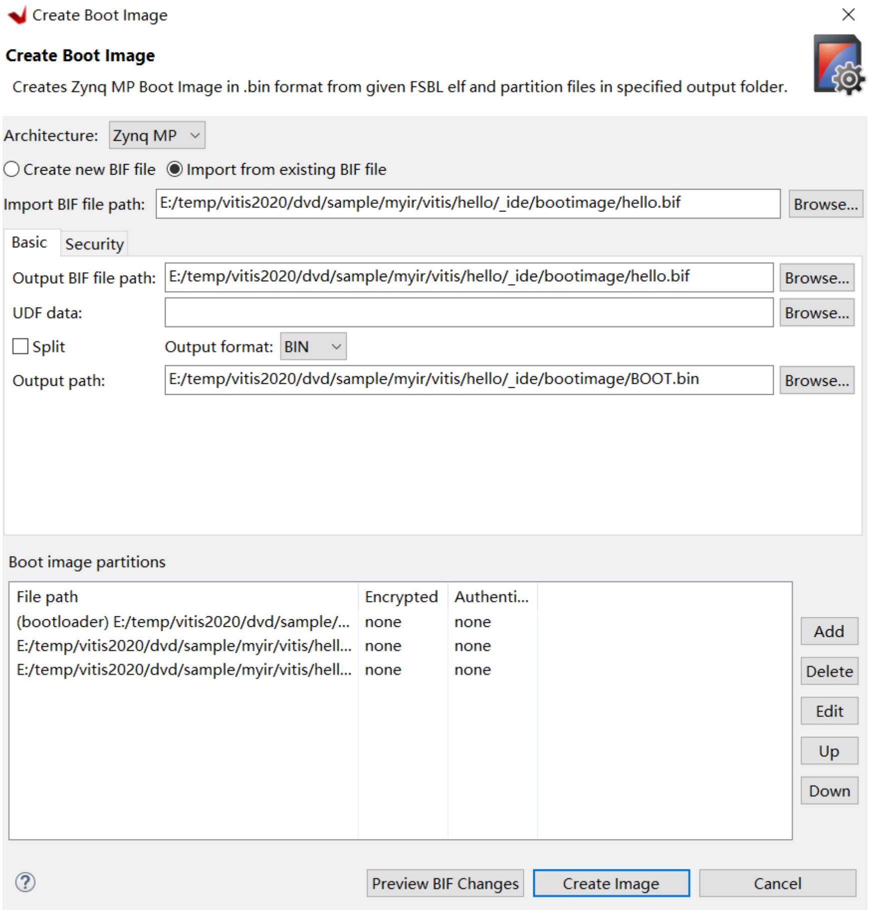
In Vitis 2020.1，click “Create Boot Image” ， Will generate BOOT.bin of hello application.



Project name: hello
Platform: design_1_wrapper
Runtime: cpp
Domain: domain_psu_cortexa53_0
CPU: psu_cortexa53_0
OS: standalone

Hardware Specification: View processors, memory ranges and peripherals.

Click “Create Image”



Create Boot Image

Create Boot Image

Creates Zynq MP Boot Image in .bin format from given FSBL elf and partition files in specified output folder.

Architecture: Zynq MP

○ Create new BIF file ● Import from existing BIF file

Import BIF file path: E:/temp/vitis2020/dvd/sample/myir/vitis/hello/_ide/bootimage/hello.bif

Basic | Security

Output BIF file path: E:/temp/vitis2020/dvd/sample/myir/vitis/hello/_ide/bootimage/hello.bif

UDF data:

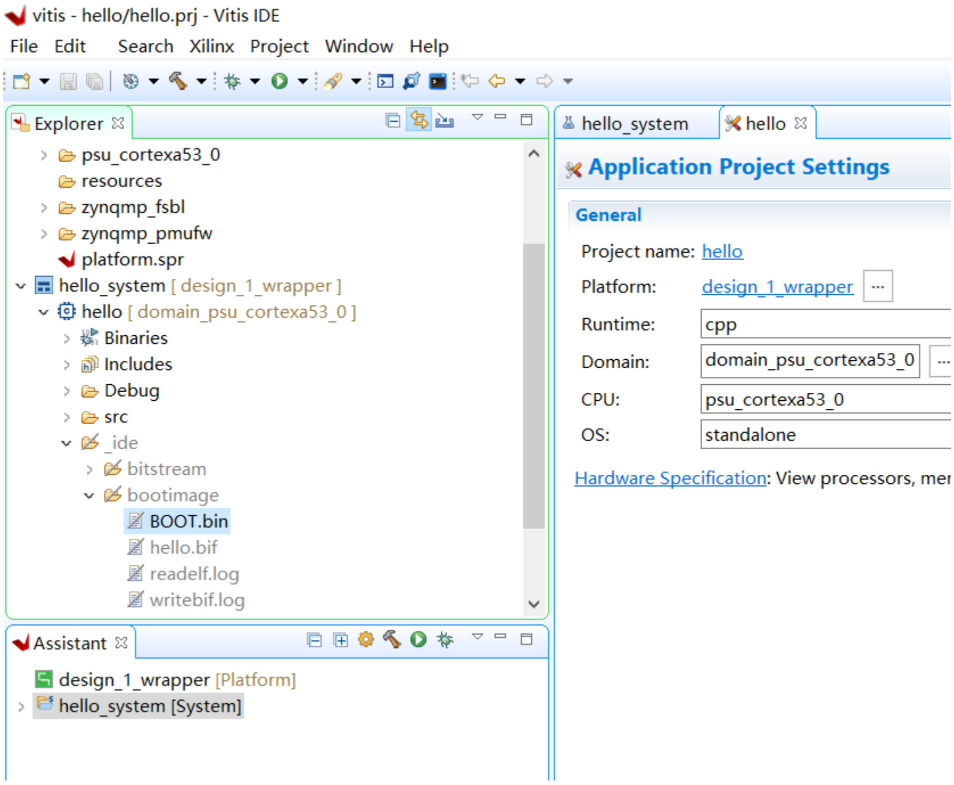
□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/vitis/hello/_ide/bootimage/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:/temp/vitis2020/dvd/sample/...	none	none
E:/temp/vitis2020/dvd/sample/myir/vitis/hell...	none	none
E:/temp/vitis2020/dvd/sample/myir/vitis/hell...	none	none

In the following “bootimage” directory, generate “BOOT.bin”



Application Project Settings

General

Project name: hello

Platform: design_1_wrapper

Runtime: cpp

Domain: domain_psu_cortexa53_0

CPU: psu_cortexa53_0

OS: standalone

Hardware Specification: View processors, mer

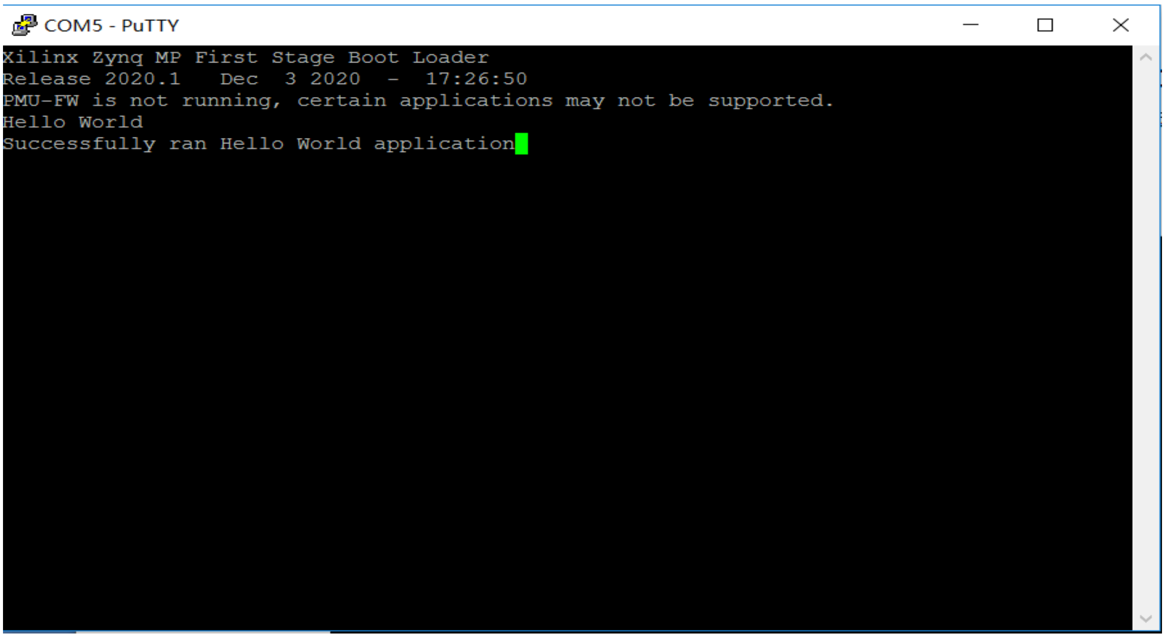
6. 1. 2 Copy the SD card boot image to the microSD card

Using the PC memory card reader slot, copy the BOOT.bin file generated in the previous step to the root directory of the microSD card (you can also use the USB memory card reader adapter if your computer does not have a microSD card reader slot). Remove the microSD card from the card reader.

6. 1. 3Start Development Board with MicroSD Card

- Set the dial switch SW1 of the development board as: 1 foot OFF, 2 foot ON, 3 foot OFF, 4 ON
- Connect 12V power supply to J2 on MYD-3EG4EV development board. The development board will start.

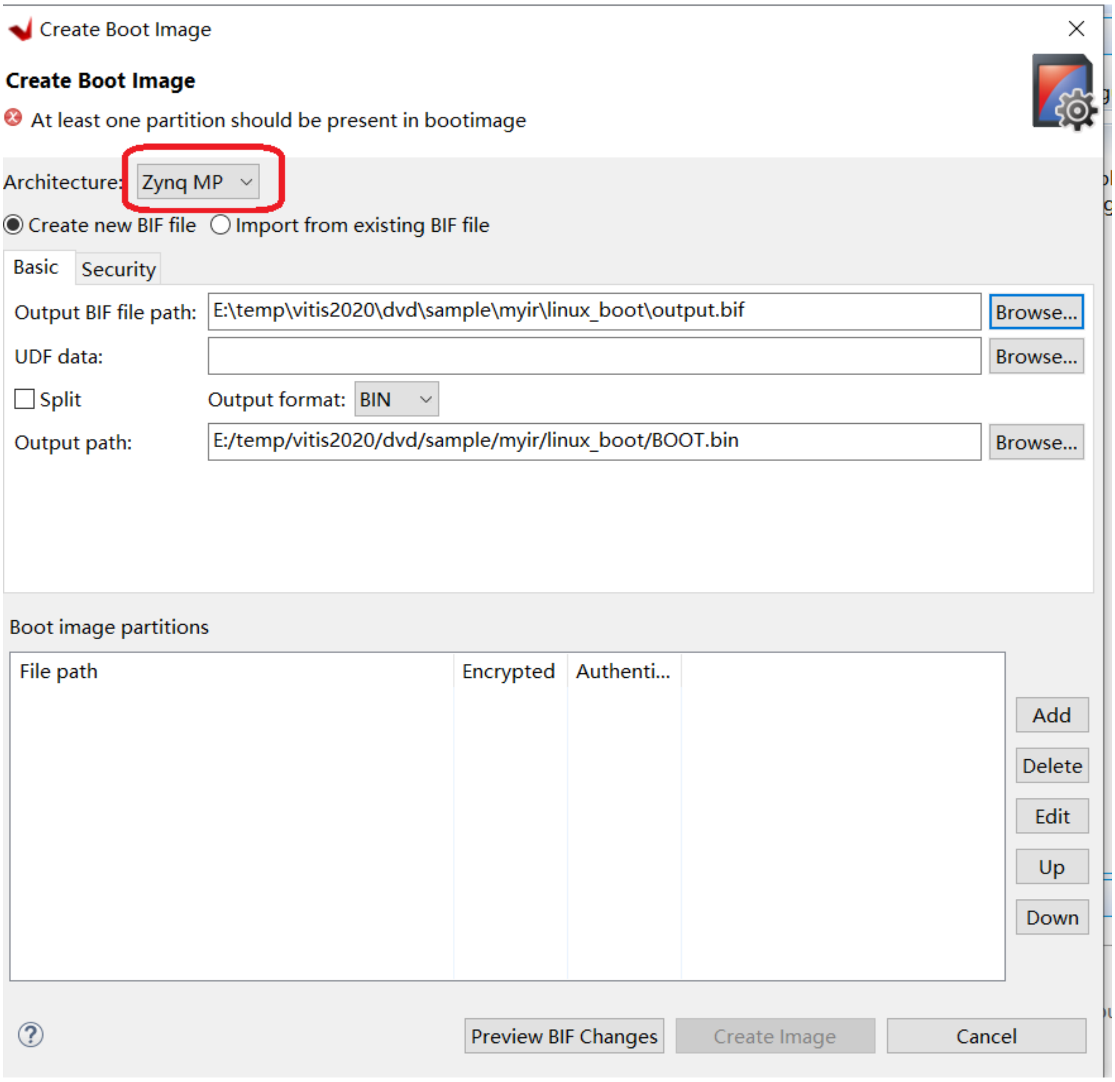
You will see the following on the putty terminal



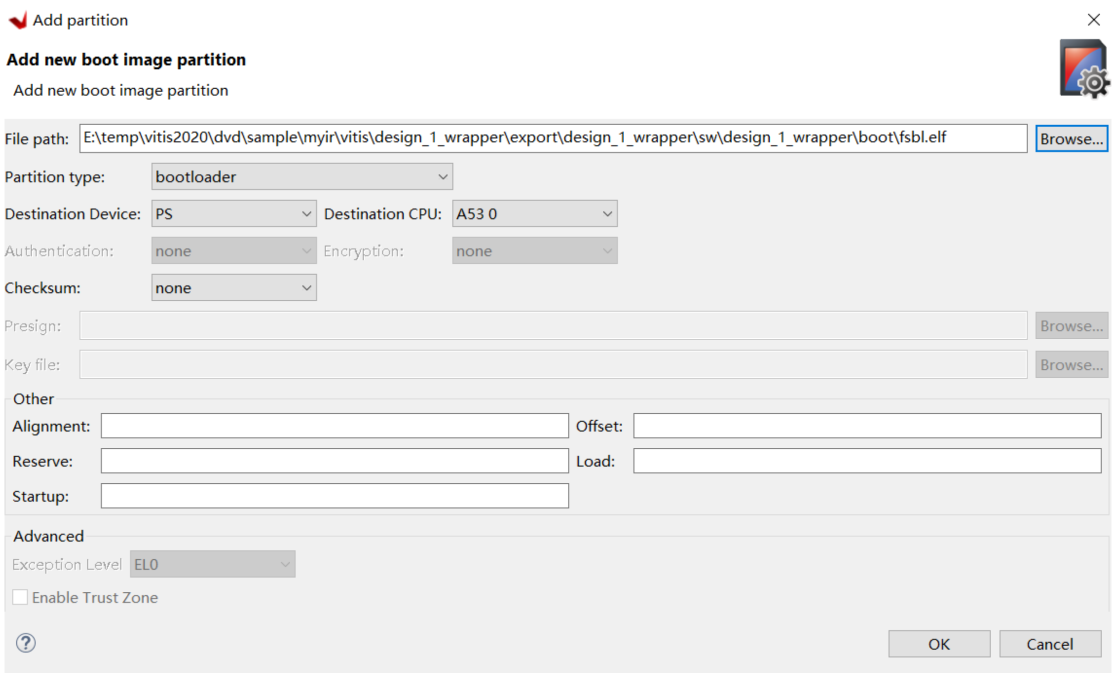
```
Xilinx Zynq MP First Stage Boot Loader
Release 2020.1   Dec  3 2020  -  17:26:50
PMU-FW is not running, certain applications may not be supported.
Hello World
Successfully ran Hello World application
```

7Generate Linux boot.bin file

•In Vitis，slect **Create Boot Image**，as shown in the figure below，select Create new BIF file，click Browse，select the output.bif file in linux_boot directory，click Add


Create Boot Image
Create Boot Image
At least one partition should be present in bootimage
Architecture: Zynq MP
● Create new BIF file ○ Import from existing BIF file
Basic
Security
Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif
Browse...
UDF data:
Browse...
□ Split
Output format: BIN
Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin
Browse...
Boot image partitions
File path
Encrypted
Authenti...
Add
Delete
Edit
Up
Down
Preview BIF Changes
Create Image
Cancel

Click Browser in the pop-up window to add fsbl.elf，click OK



Add partition

Add new boot image partition

Add new boot image partition

File path: E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf

Partition type: bootloader

Destination Device: PS

Destination CPU: A53 0

Authentication: none

Encryption: none

Checksum: none

Presign:

Key file:

Other

Alignment:

Offset:

Reserve:

Load:

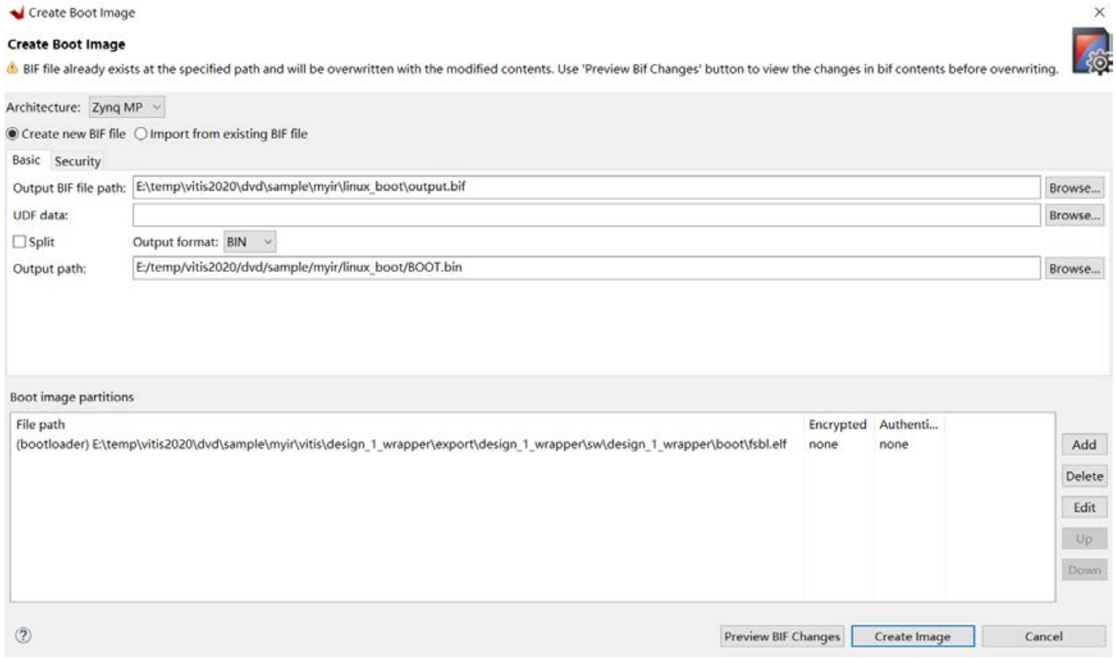
Startup:

Advanced

Exception Level EL0

□ Enable Trust Zone

Click Add



Create Boot Image

Create Boot Image

⚠ BIF file already exists at the specified path and will be overwritten with the modified contents. Use 'Preview Bif Changes' button to view the changes in bif contents before overwriting.

Architecture: Zynq MP

● Create new BIF file ○ Import from existing BIF file

Basic Security

Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif

UDF data:

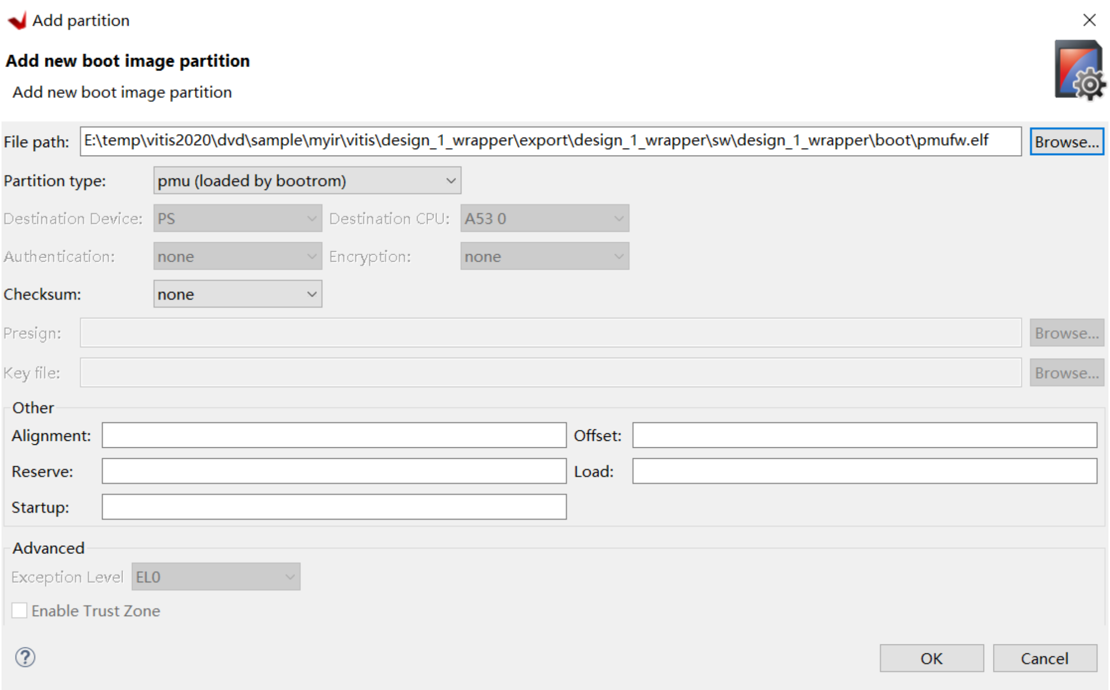
□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf	none	none

Click Browser in the pop-up window to add pmufw.elf，click OK



Add partition

Add new boot image partition

Add new boot image partition

File path: E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\pmufw.elf

Partition type: pmu (loaded by bootrom)

Destination Device: PS Destination CPU: A53 0

Authentication: none Encryption: none

Checksum: none

Presign:

Key file:

Other

Alignment: Offset:

Reserve: Load:

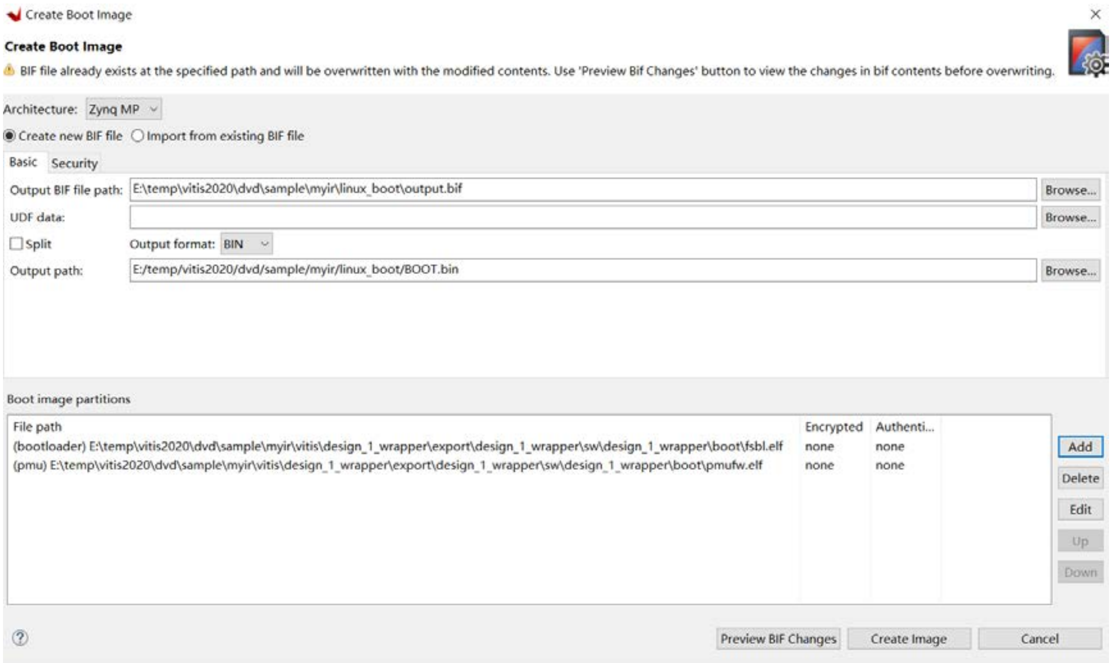
Startup:

Advanced

Exception Level EL0

□ Enable Trust Zone

Click Add



Create Boot Image

Create Boot Image

BIF file already exists at the specified path and will be overwritten with the modified contents. Use 'Preview Bif Changes' button to view the changes in bif contents before overwriting.

Architecture: Zynq MP

● Create new BIF file ○ Import from existing BIF file

Basic Security

Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif

UDF data:

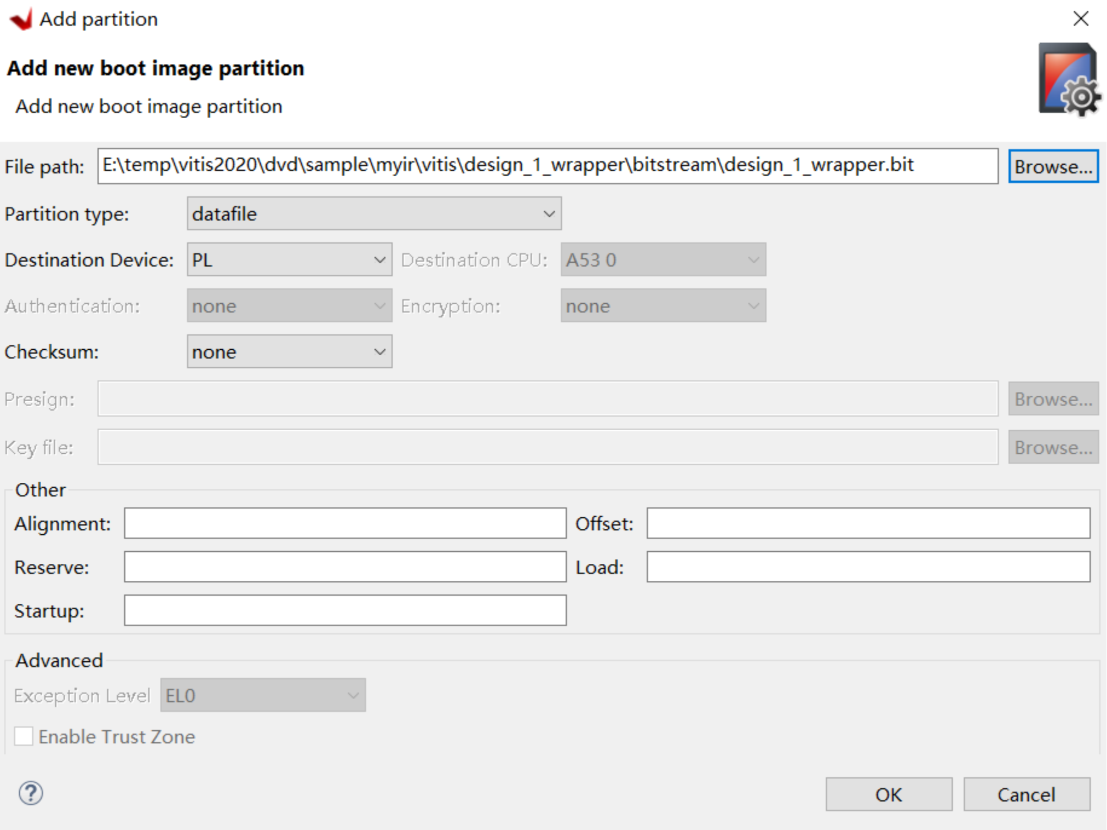
□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf	none	none
(pmu) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\pmufw.elf	none	none

Click Browser in the pop-up window to add bitstream，click OK



Add partition

Add new boot image partition

Add new boot image partition

File path: E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\bitstream\design_1_wrapper.bit

Partition type: datafile

Destination Device: PL

Destination CPU: A53 0

Authentication: none

Encryption: none

Checksum: none

Presign:

Key file:

Other

Alignment:

Offset:

Reserve:

Load:

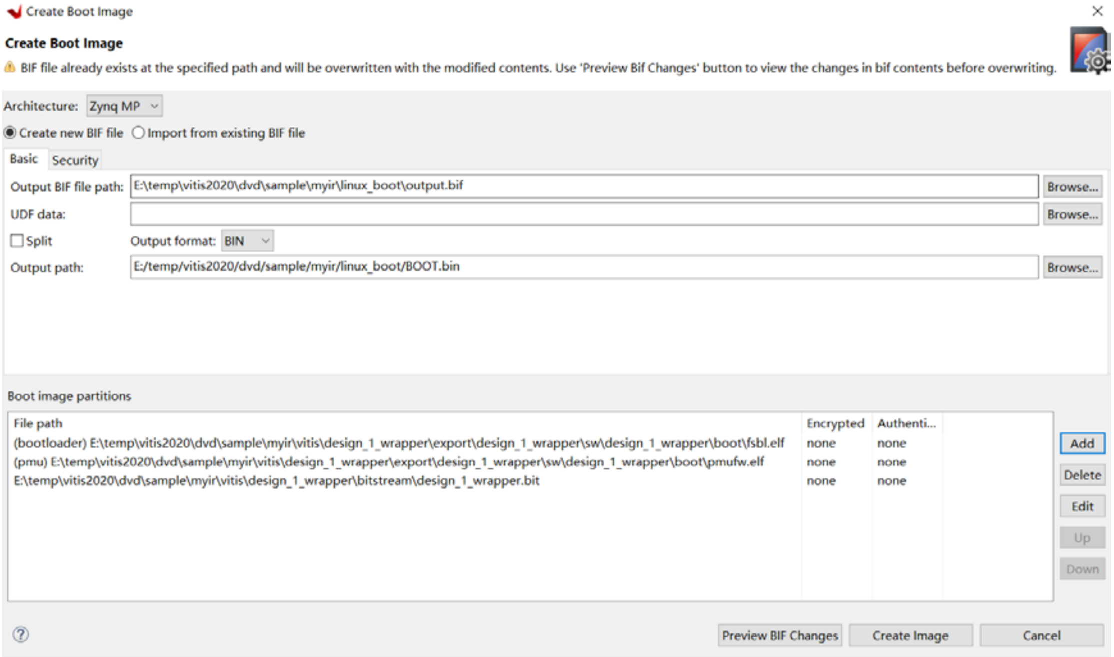
Startup:

Advanced

Exception Level EL0

□ Enable Trust Zone

Click Add



Create Boot Image

Create Boot Image

BIF file already exists at the specified path and will be overwritten with the modified contents. Use 'Preview Bif Changes' button to view the changes in bif contents before overwriting.

Architecture: Zynq MP

● Create new BIF file ○ Import from existing BIF file

Basic | Security

Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif

UDF data:

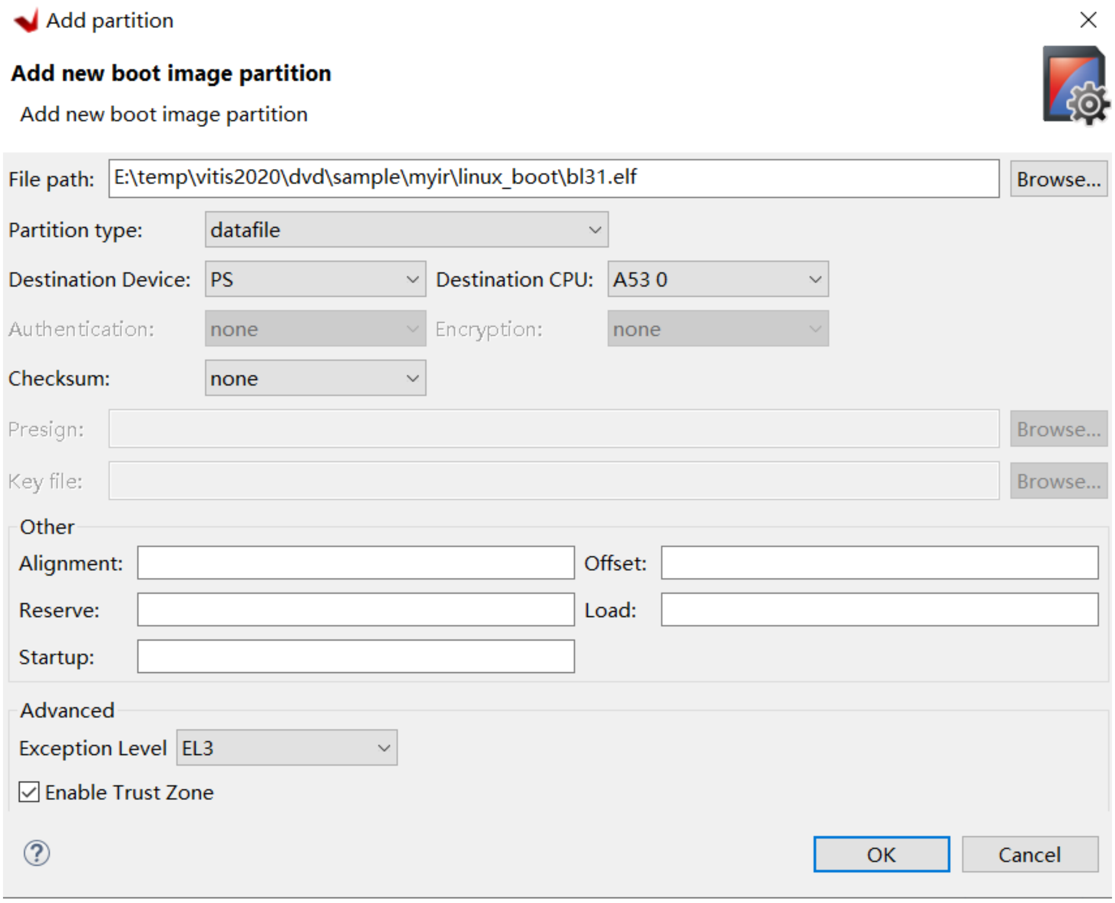
□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf	none	none
(pmu) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\pmufw.elf	none	none
E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\bitstream\design_1_wrapper.bit	none	none

Click Browser in the pop-up window to add bl31.elf，click OK



Add partition

Add new boot image partition

Add new boot image partition

File path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\bl31.elf

Partition type: datafile

Destination Device: PS

Destination CPU: A53 0

Authentication: none

Encryption: none

Checksum: none

Presign:

Key file:

Other

Alignment:

Offset:

Reserve:

Load:

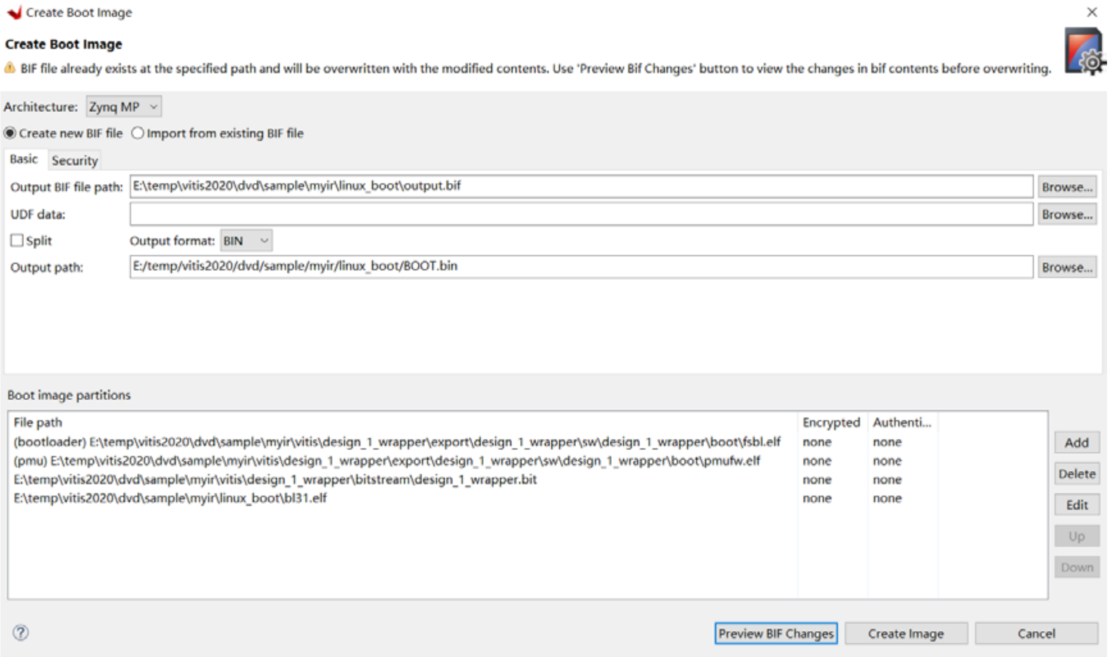
Startup:

Advanced

Exception Level EL3

☑ Enable Trust Zone

Click Add



Create Boot Image

Create Boot Image

BIF file already exists at the specified path and will be overwritten with the modified contents. Use 'Preview Bif Changes' button to view the changes in bif contents before overwriting.

Architecture: Zynq MP

● Create new BIF file ○ Import from existing BIF file

Basic Security

Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif

UDF data:

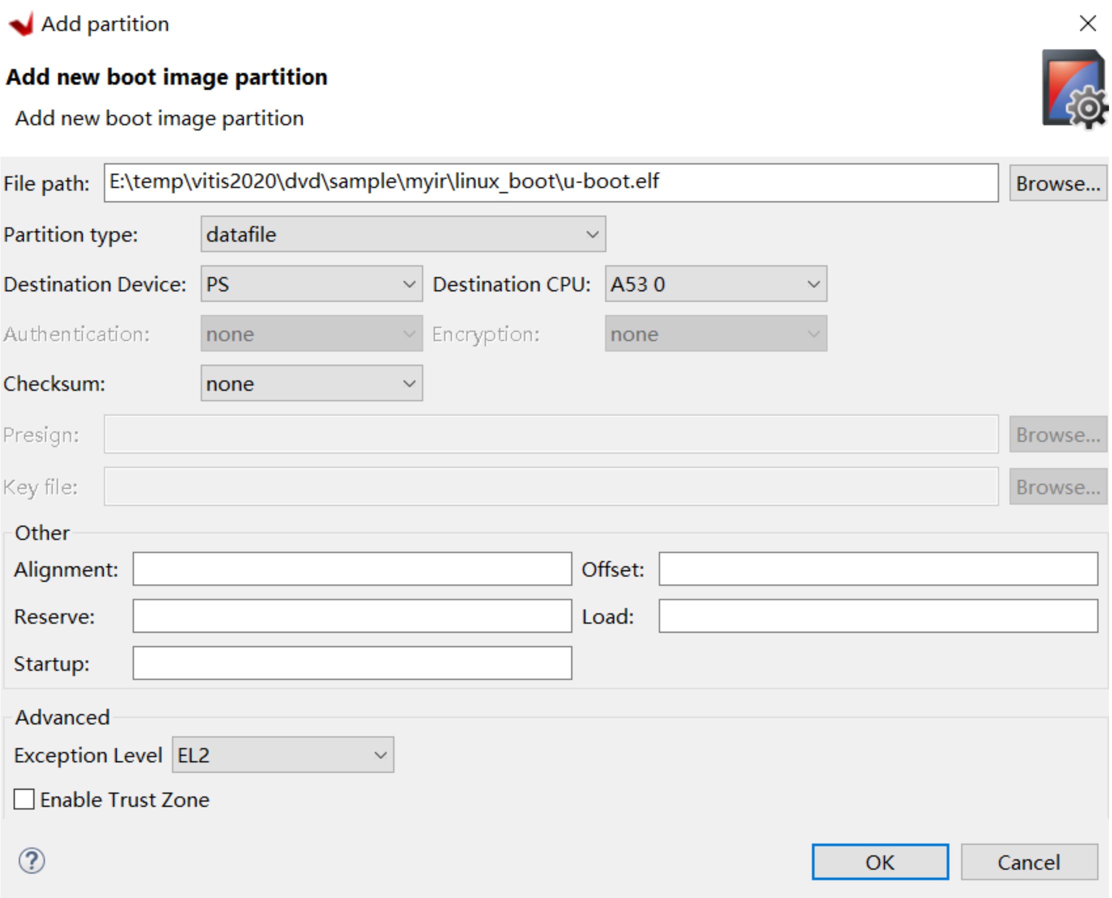
□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf	none	none
(pmu) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\pmufw.elf	none	none
E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\bitstream\design_1_wrapper.bit	none	none
E:\temp\vitis2020\dvd\sample\myir\linux_boot\bl31.elf	none	none

Click Browser in the pop-up window to add u-boot.elf，click OK



Add partition

Add new boot image partition

Add new boot image partition

File path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\u-boot.elf

Partition type: datafile

Destination Device: PS

Destination CPU: A53 0

Authentication: none

Encryption: none

Checksum: none

Presign:

Key file:

Other

Alignment:

Offset:

Reserve:

Load:

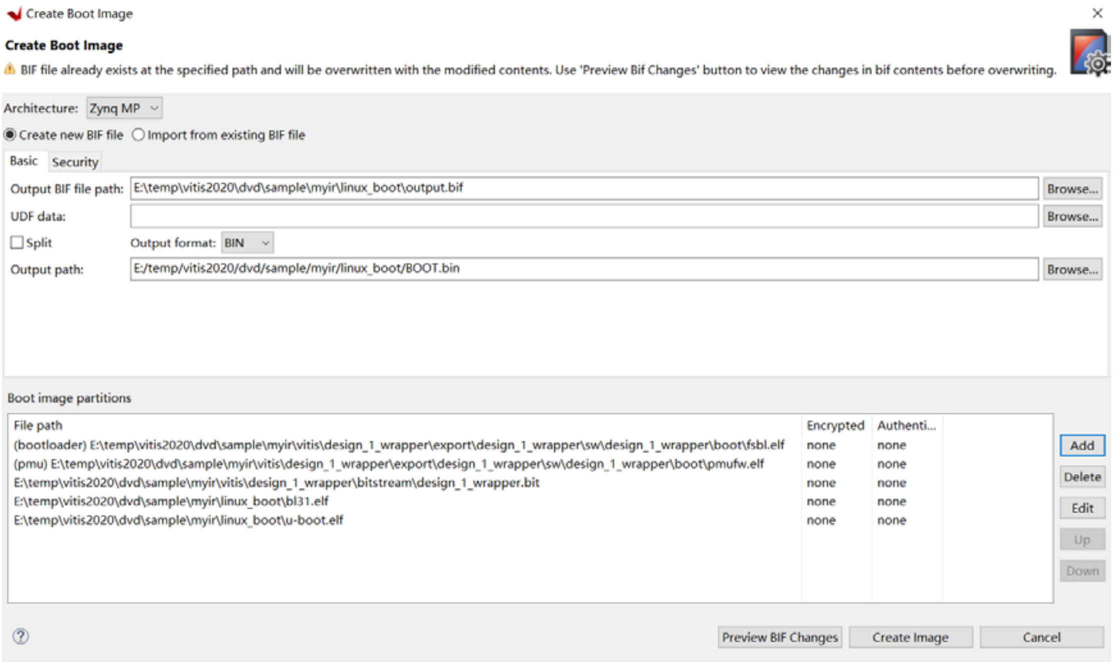
Startup:

Advanced

Exception Level EL2

□ Enable Trust Zone

Click Create Image，in linux_boot directory，generate BOOT.bin



Create Boot Image

Create Boot Image

⚠ BIF file already exists at the specified path and will be overwritten with the modified contents. Use 'Preview Bif Changes' button to view the changes in bif contents before overwriting.

Architecture: Zynq MP

● Create new BIF file ○ Import from existing BIF file

Basic | Security

Output BIF file path: E:\temp\vitis2020\dvd\sample\myir\linux_boot\output.bif

UDF data:

□ Split Output format: BIN

Output path: E:/temp/vitis2020/dvd/sample/myir/linux_boot/BOOT.bin

Boot image partitions

File path	Encrypted	Authenti...
(bootloader) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\fsbl.elf	none	none
(pmu) E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\export\design_1_wrapper\sw\design_1_wrapper\boot\pmufw.elf	none	none
E:\temp\vitis2020\dvd\sample\myir\vitis\design_1_wrapper\bitstream\design_1_wrapper.bit	none	none
E:\temp\vitis2020\dvd\sample\myir\linux_boot\bl31.elf	none	none
E:\temp\vitis2020\dvd\sample\myir\linux_boot\u-boot.elf	none	none

With other files in the Linux development manual, you can run Linux on the development board。

Appendix 1 Warranty & Technical Support Services

MYIR Tech Limited is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support

MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date. The service covers:
1. Technical support service

a) MYIR offers technical support for the hardware and software materials which have provided to customers;

b) To help customers compile and run the source code we offer;

c) To help customers solve problems occurred during operations if users follow the user manual documents;

d) To judge whether the failure exists;

e) To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

a) Hardware or software problems occurred during customers' own development;

b) Problems occurred when customers compile or run the OS which is tailored by themselves;

c) Problems occurred during customers' own applications development;

d) Problems occurred during the modification of MYIR's software source code.

2. After-sales maintenance service
The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following

situations are not included in the scope of our free maintenance service:

a) The warranty period is expired;

b) The customer cannot provide proof-of-purchase or the product has no serial number;

c) The customer has not followed the instruction of the manual which has caused the damage the product;

d) Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;

e) Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;

f) Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;

g) Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips:

1) MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.

2) Please do not use finger nails or hard sharp object to touch the surface of the LCD.

3) MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.

4) Do not clean the surface of the screen with chemicals.

5) Please read through the product user manual before you using MYIR's products.

6) For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

3. Maintenance period and charges

a) MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
b) For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

4. Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

5. Products Life Cycle

MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

MYIR Tech Limited

Room 1306, Wensheng Center, Wenjin Plaza,

North Wenjin Road, Luohu District, Shenzhen, China 518020

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836
Fax: +86-755-25532724

Website: www.myirtech.com