

Creating Your First ArtyZ7 Application

Course Workbook

Table of Contents

About this Workbook	<u>Page 3</u>
Pre-Lab: Workshop Pre-requisites	<u>Page 4</u>
Lab 1: Understanding Project creation & Flow	<u>Page 7</u>

About this Workbook

The contents of this workbook are created by Aduvo Engineering & Training, Ltd.

If you have any questions about the contents, or need assistance, please contact Adam Taylor at adam@adiuvoengineering.com.

Pre-Lab

Workshop Pre-requisites

Required Hardware

ArtyZ7-20

Usb cable

Downloads and Installations

Step 1 – Download and install the following at least 1 day prior to the workshop. This may take a significant amount of time and drive space.

Watch the video available [here](#) to show how to configure the installation

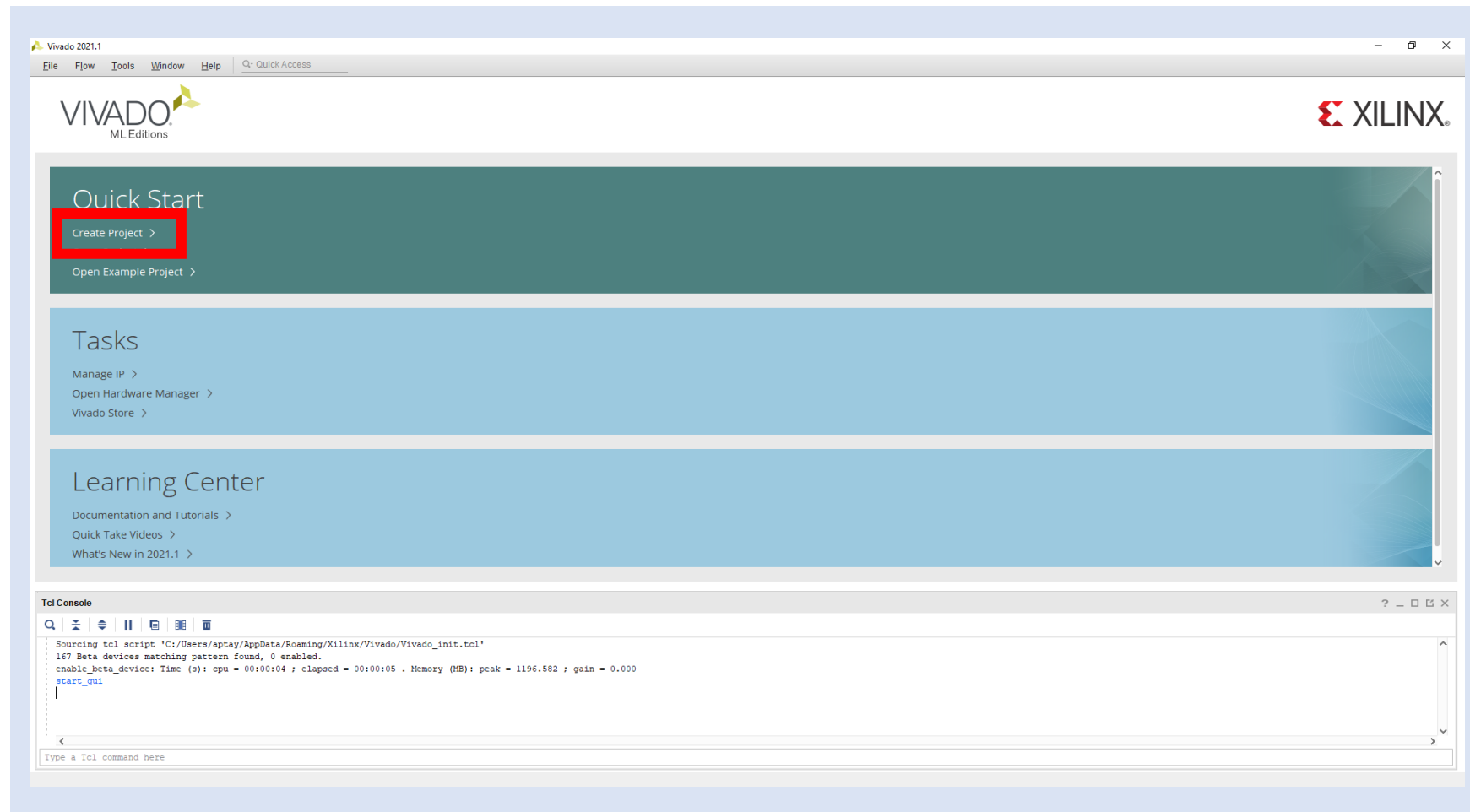
Vitis 2021.1	Download
--------------	--------------------------

Lab 1

Project creation & Flow

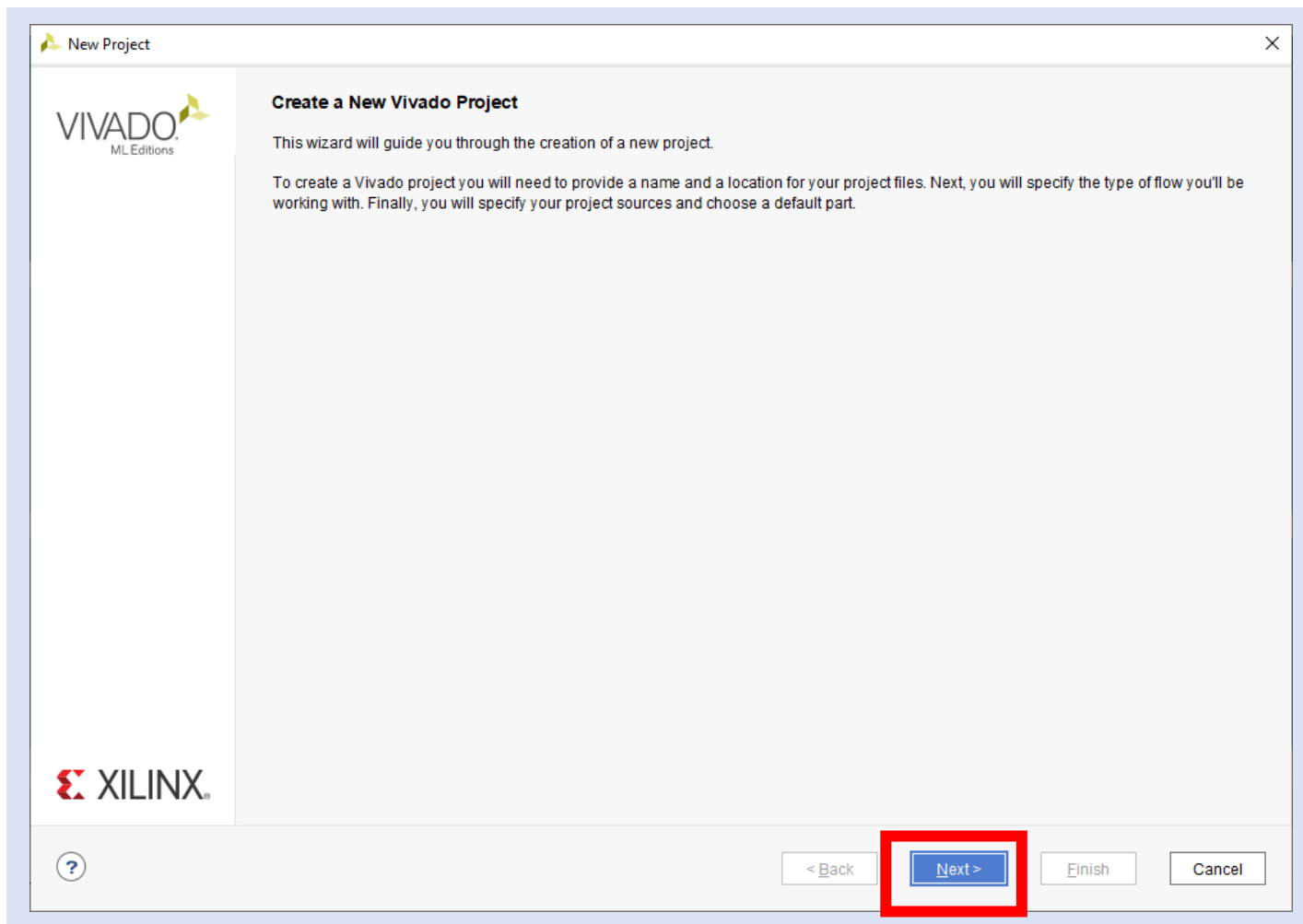
Lab 1: Understanding Vitis Project creation & Flow

Step 1 – Open Vivado 2021.1 – Click Create New Project



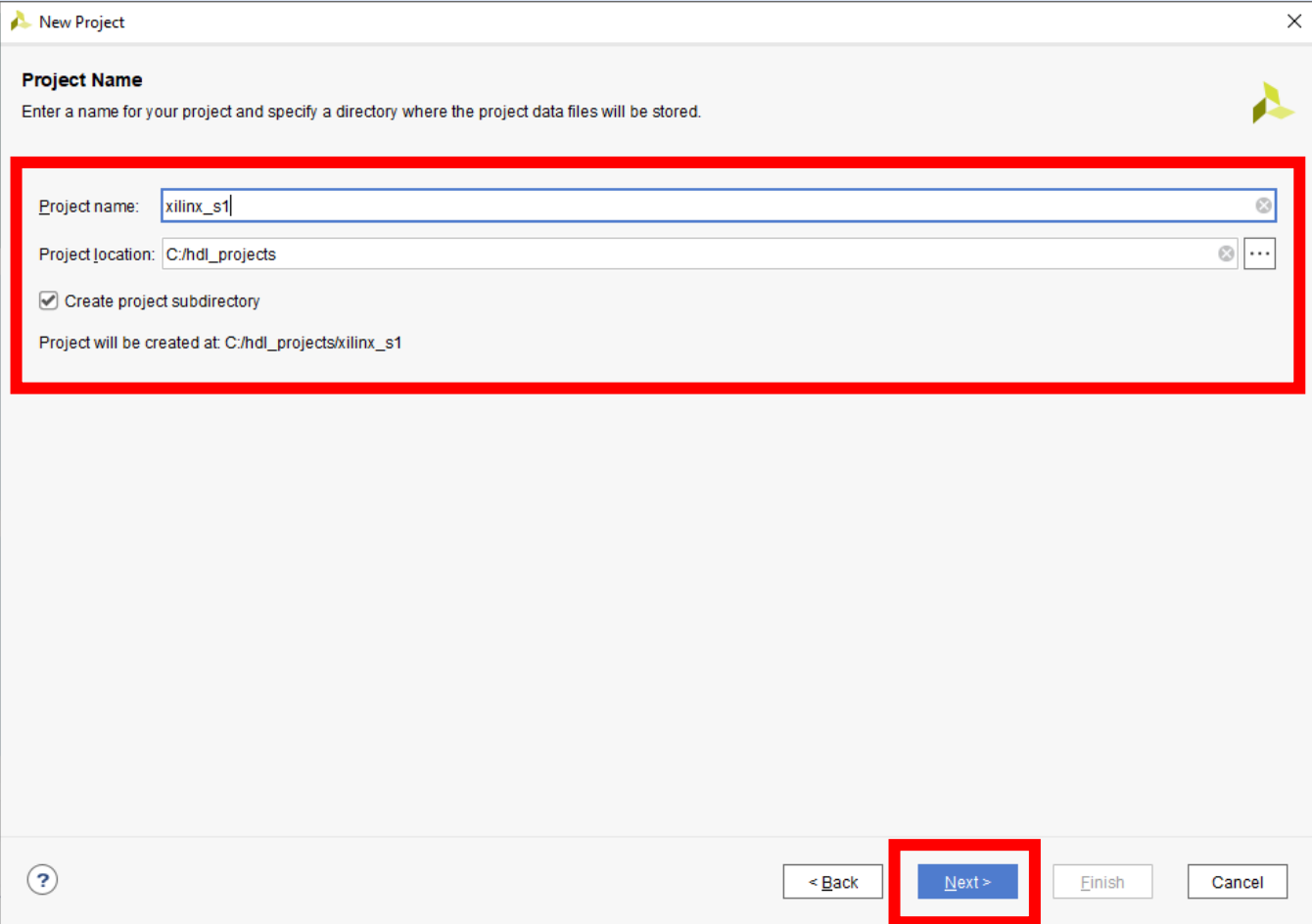
Lab 1: Understanding Vitis Project creation & Flow

Step 2 – Click Next



Lab 1: Understanding Vitis Project creation & Flow

Step 3 – Enter a project name and location to save the project, 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: xilinx_s1

Project location: C:/hdl_projects

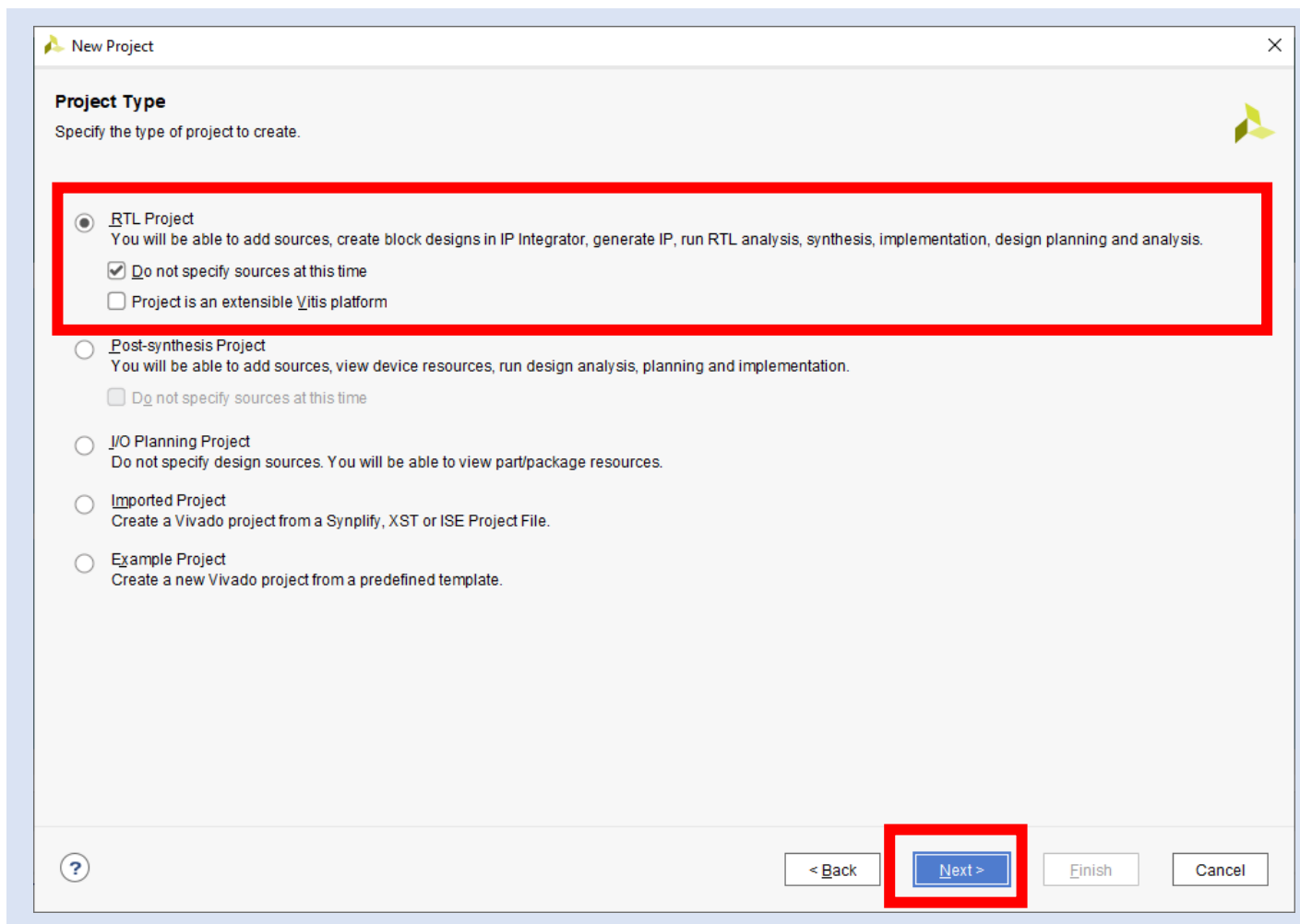
Create project subdirectory

Project will be created at: C:/hdl_projects/xilinx_s1

? < Back Next > Finish Cancel

Lab 1: Understanding Vitis Project creation & Flow

Step 4 – Select RTL project and check do not include sources, 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
 Project is an extensible Vitis platform

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.

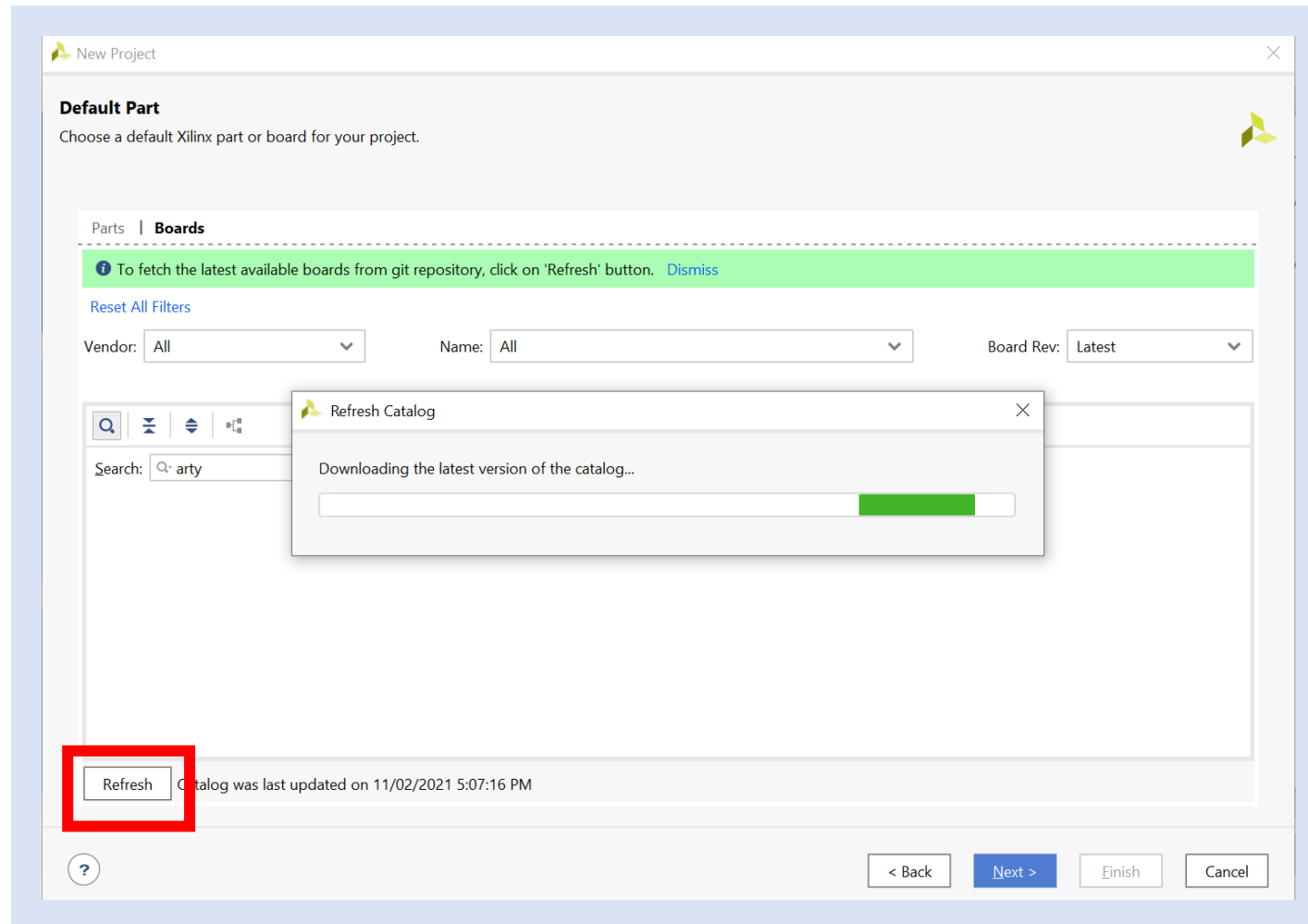
Imported Project
Create a Vivado project from a Synplify, XST or ISE Project File.

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

? < Back Next > Finish Cancel

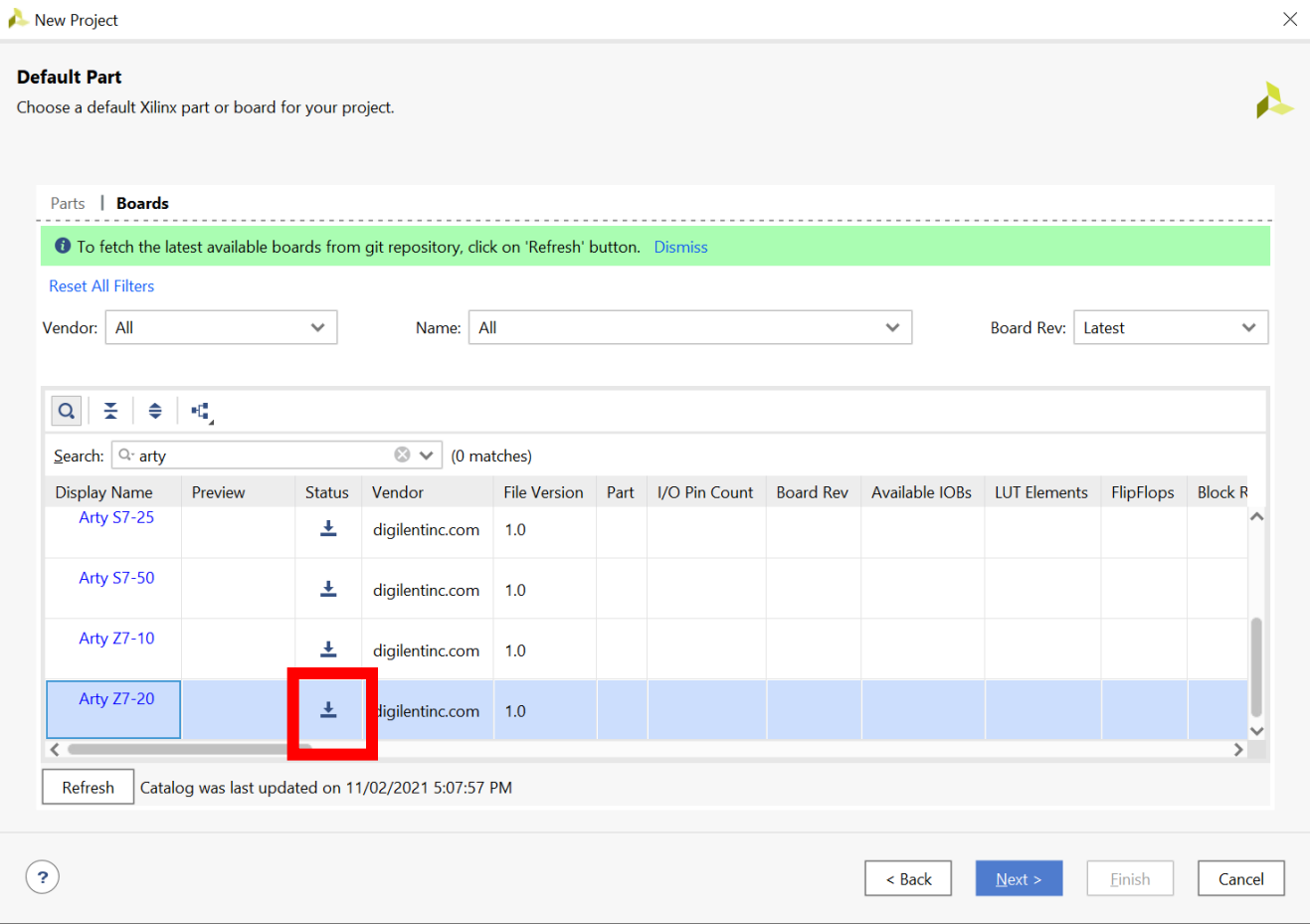
Lab 1: Understanding Vitis Project creation & Flow

Step 5 – If the Arty Z7 isn't in the list of boards select Refresh and the board list will be updated



Lab 1: Understanding Vitis Project creation & Flow

Step 6 – Select the download and the board will install

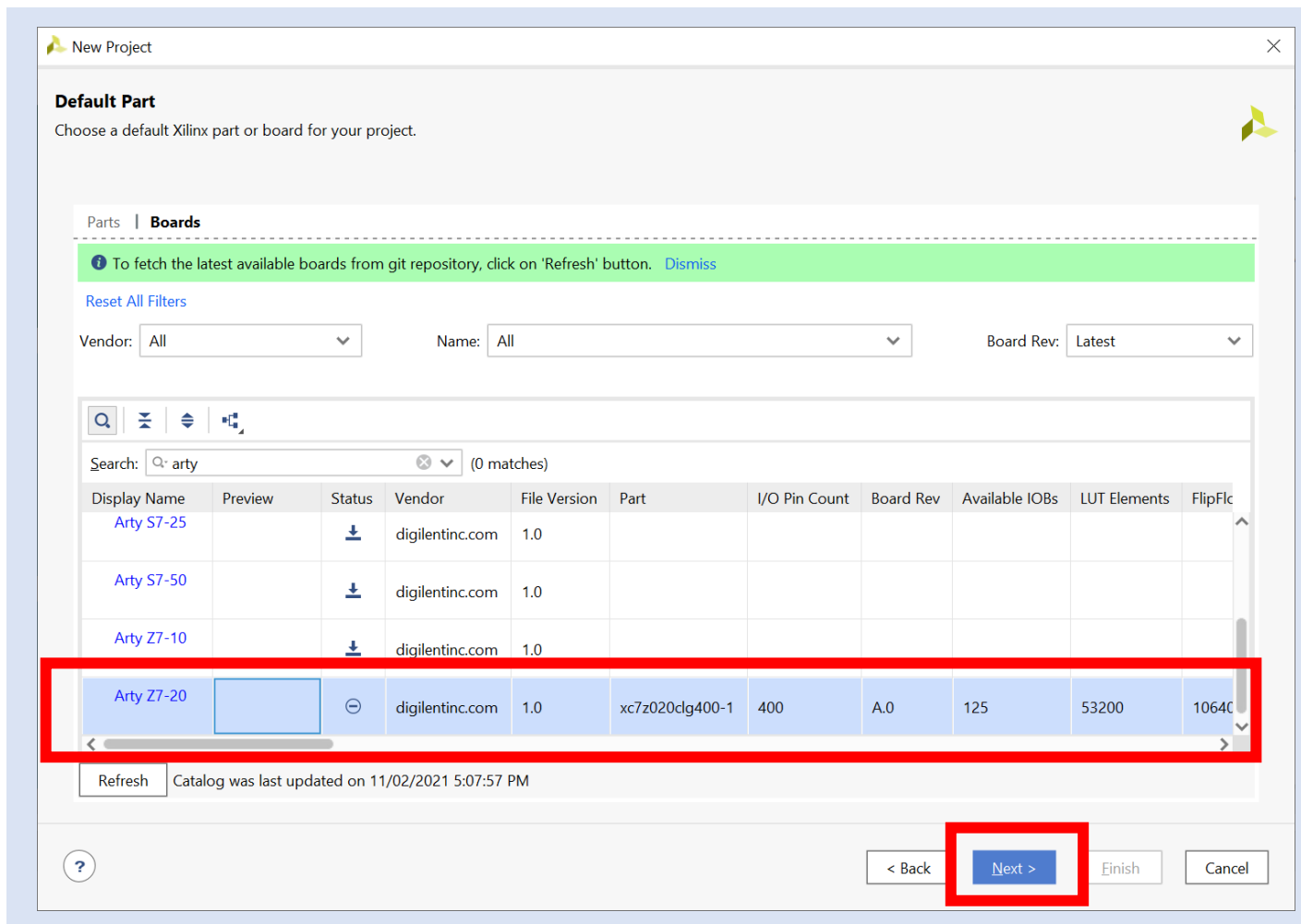


The screenshot shows the 'New Project' dialog box in Vitis. The 'Default Part' section is active, and the 'Boards' tab is selected. A message at the top indicates that the user should click the 'Refresh' button to fetch the latest boards from the git repository. Below this, there are filter options for Vendor (All), Name (All), and Board Rev (Latest). A search bar contains the text 'arty' and shows '(0 matches)'. A table of boards is displayed, with the 'Arty Z7-20' board highlighted. The download icon for this board is circled in red. At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board Rev	Available IOBs	LUT Elements	FlipFlops	Block R
Arty S7-25		↓	digilentinc.com	1.0							
Arty S7-50		↓	digilentinc.com	1.0							
Arty Z7-10		↓	digilentinc.com	1.0							
Arty Z7-20		↓	digilentinc.com	1.0							

Lab 1: Understanding Vitis Project creation & Flow

Step 7 – Select Vendor Digilentinc and Arty Z7-20, click next



The screenshot shows the 'New Project' dialog in Vitis. The 'Default Part' section is active, and the 'Boards' tab is selected. A search for 'arty' has been performed, resulting in a list of boards. The 'Arty Z7-20' board is selected and highlighted with a red box. The 'Next >' button is also highlighted with a red box.

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: arty (0 matches)

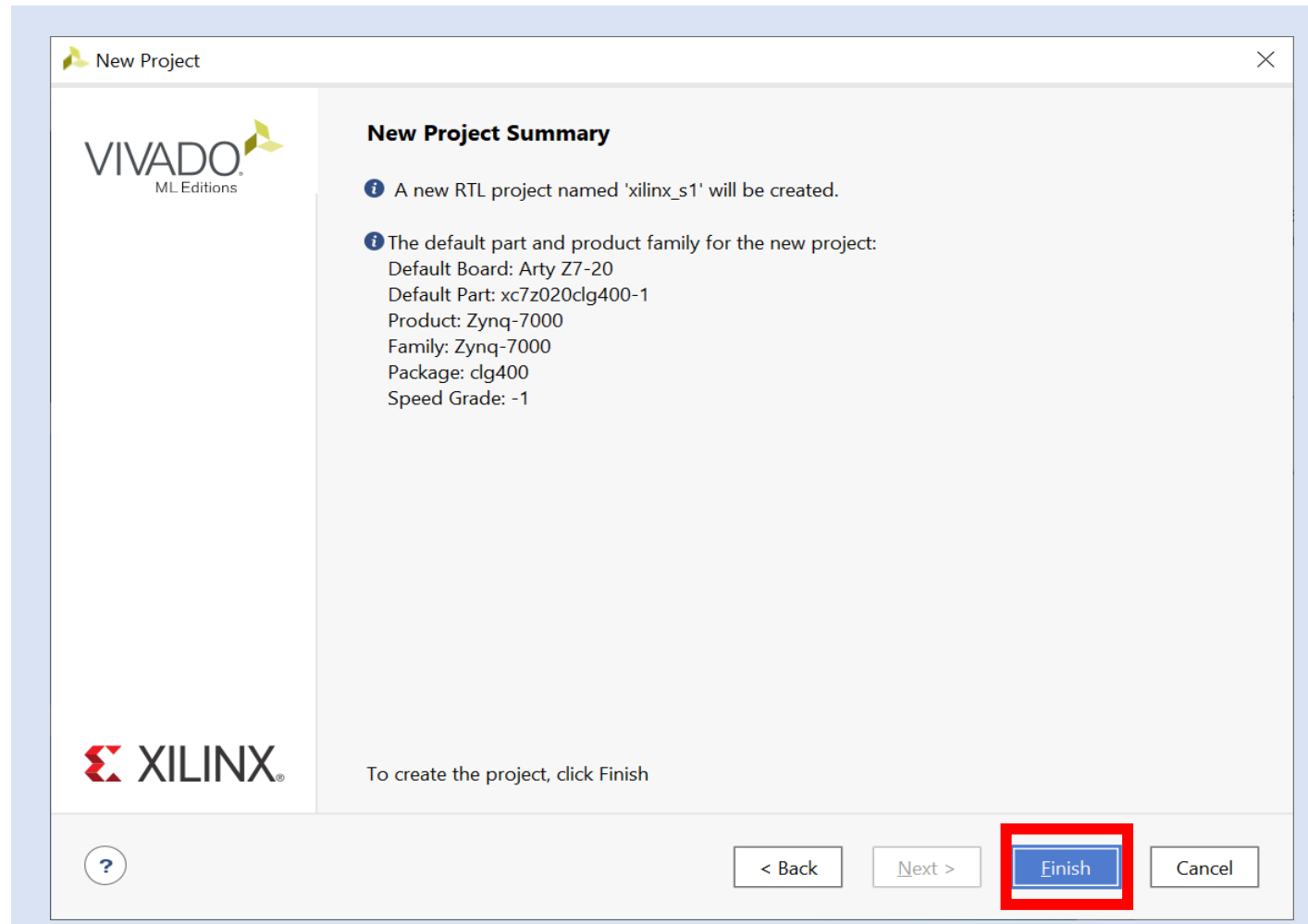
Display Name	Preview	Status	Vendor	File Version	Part	I/O Pin Count	Board Rev	Available IOBs	LUT Elements	FlipFlops
Arty S7-25		↓	digilentinc.com	1.0						
Arty S7-50		↓	digilentinc.com	1.0						
Arty Z7-10		↓	digilentinc.com	1.0						
Arty Z7-20		⊖	digilentinc.com	1.0	xc7z020clg400-1	400	A.0	125	53200	10640

Refresh Catalog was last updated on 11/02/2021 5:07:57 PM

< Back **Next >** Finish Cancel

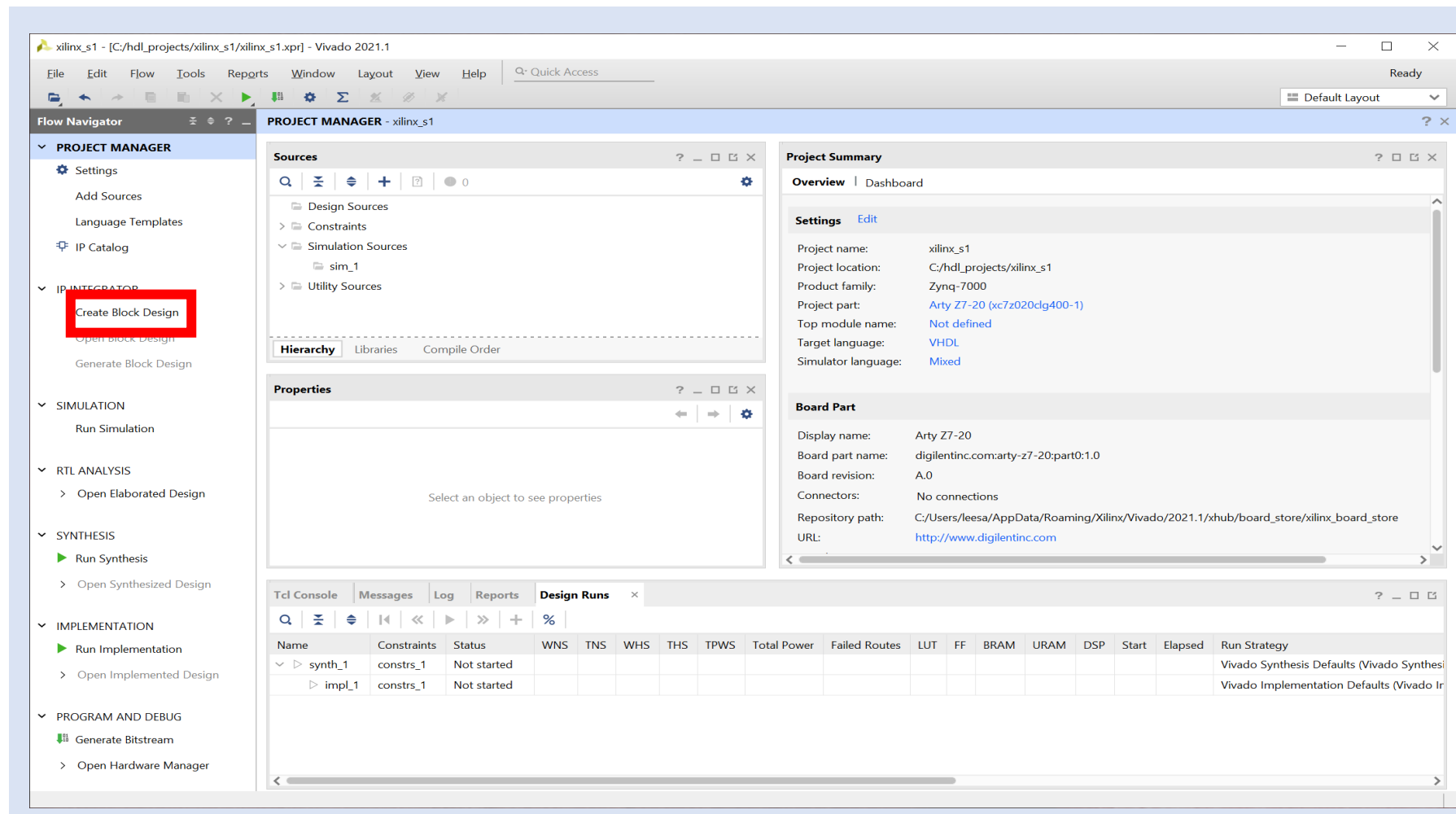
Lab 1: Understanding Vitis Project creation & Flow

Step 8 – Click Finish



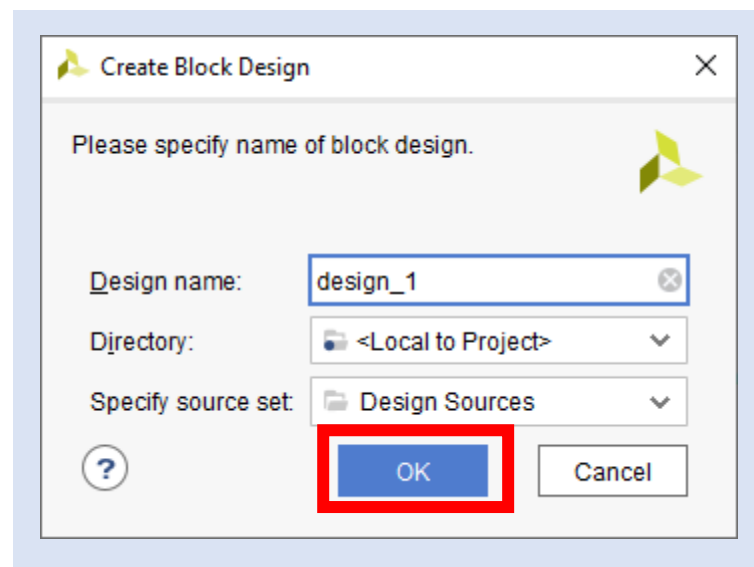
Lab 1: Understanding Vitis Project creation & Flow

Step 9 – Click on Create Block Diagram – This will open IP editor



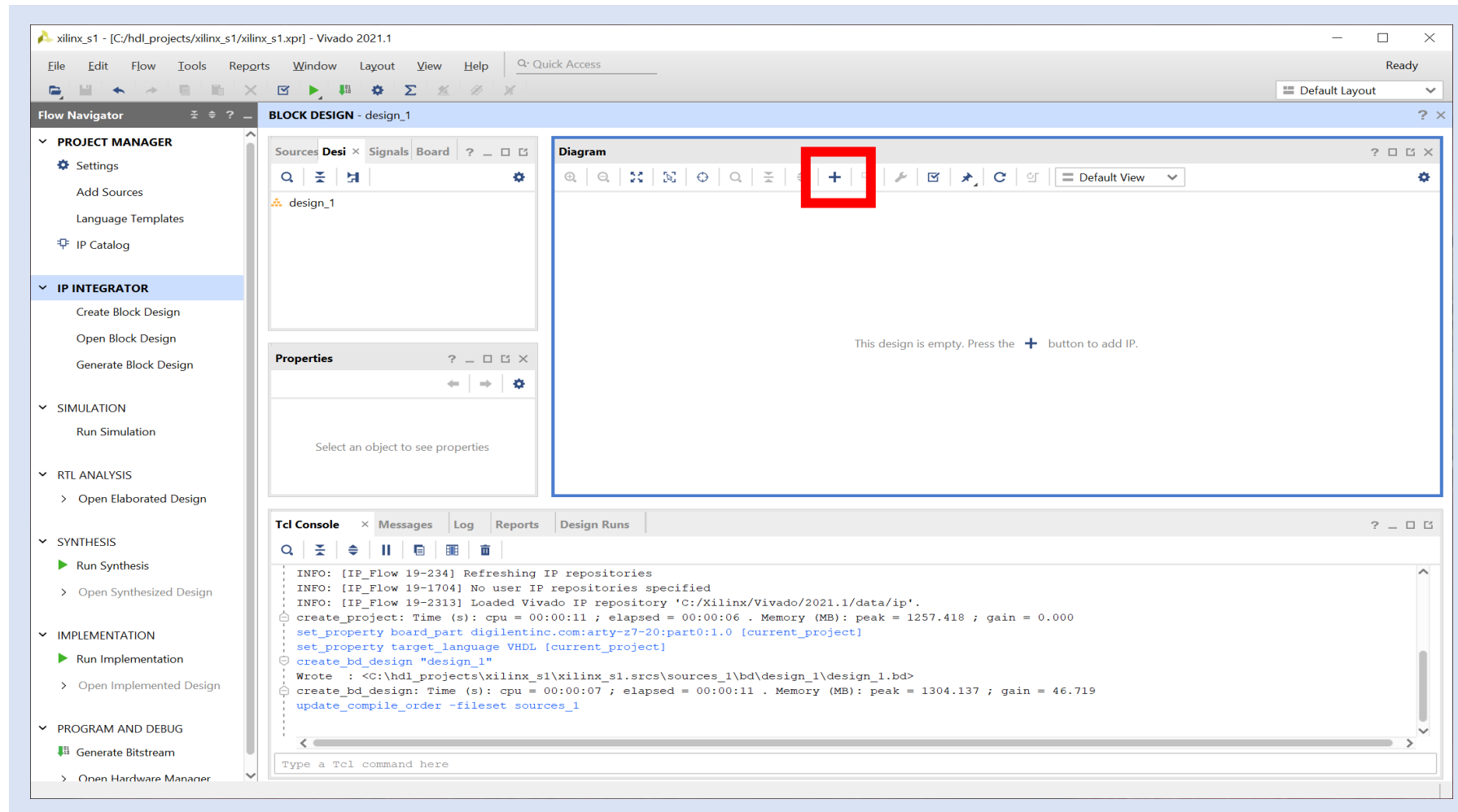
Lab 1: Understanding Vitis Project creation & Flow

Step 10 – Leave the settings as default and click OK



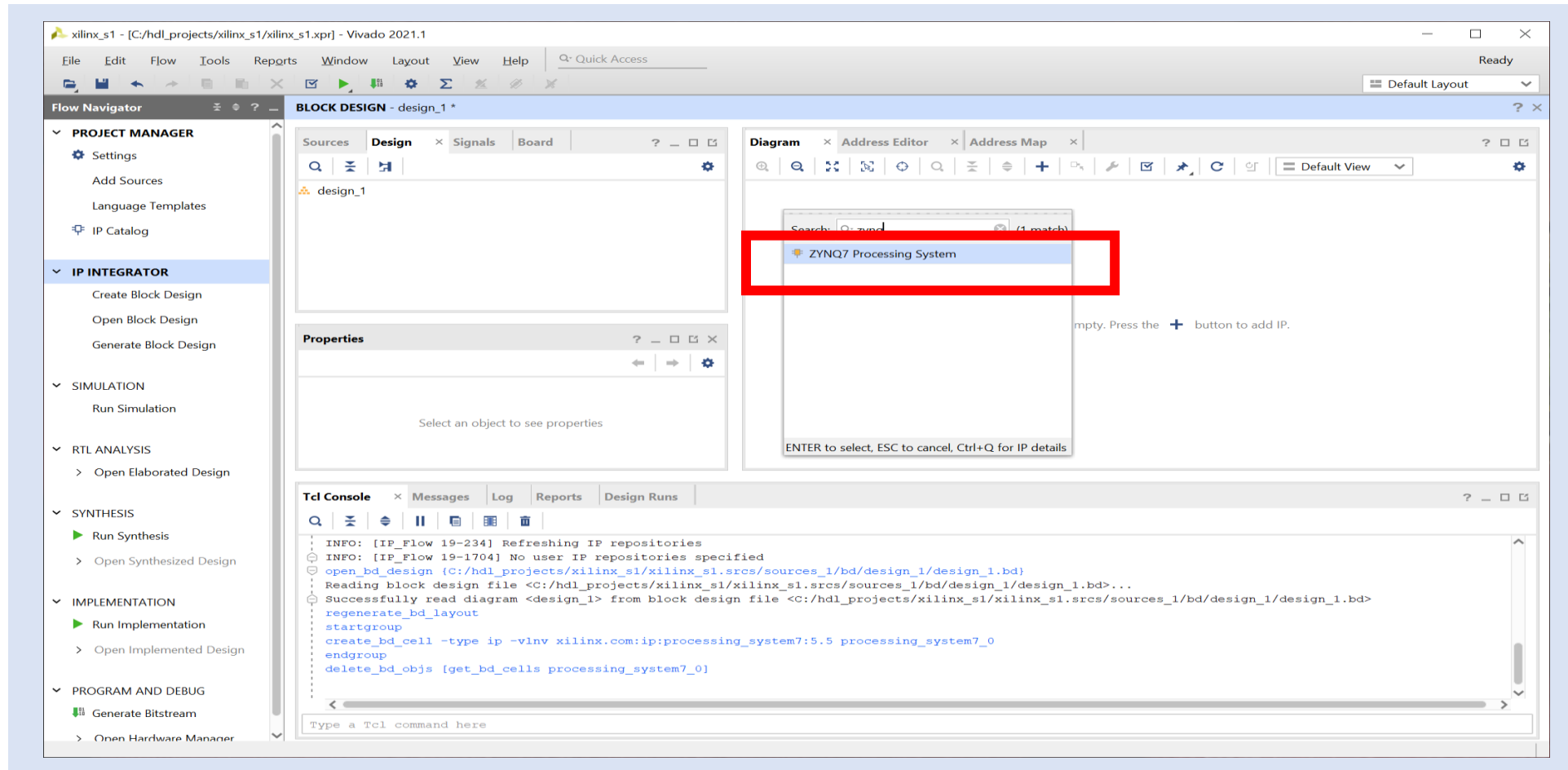
Lab 1: Understanding Vitis Project creation & Flow

Step 11 – Click on the + button to open the IP list



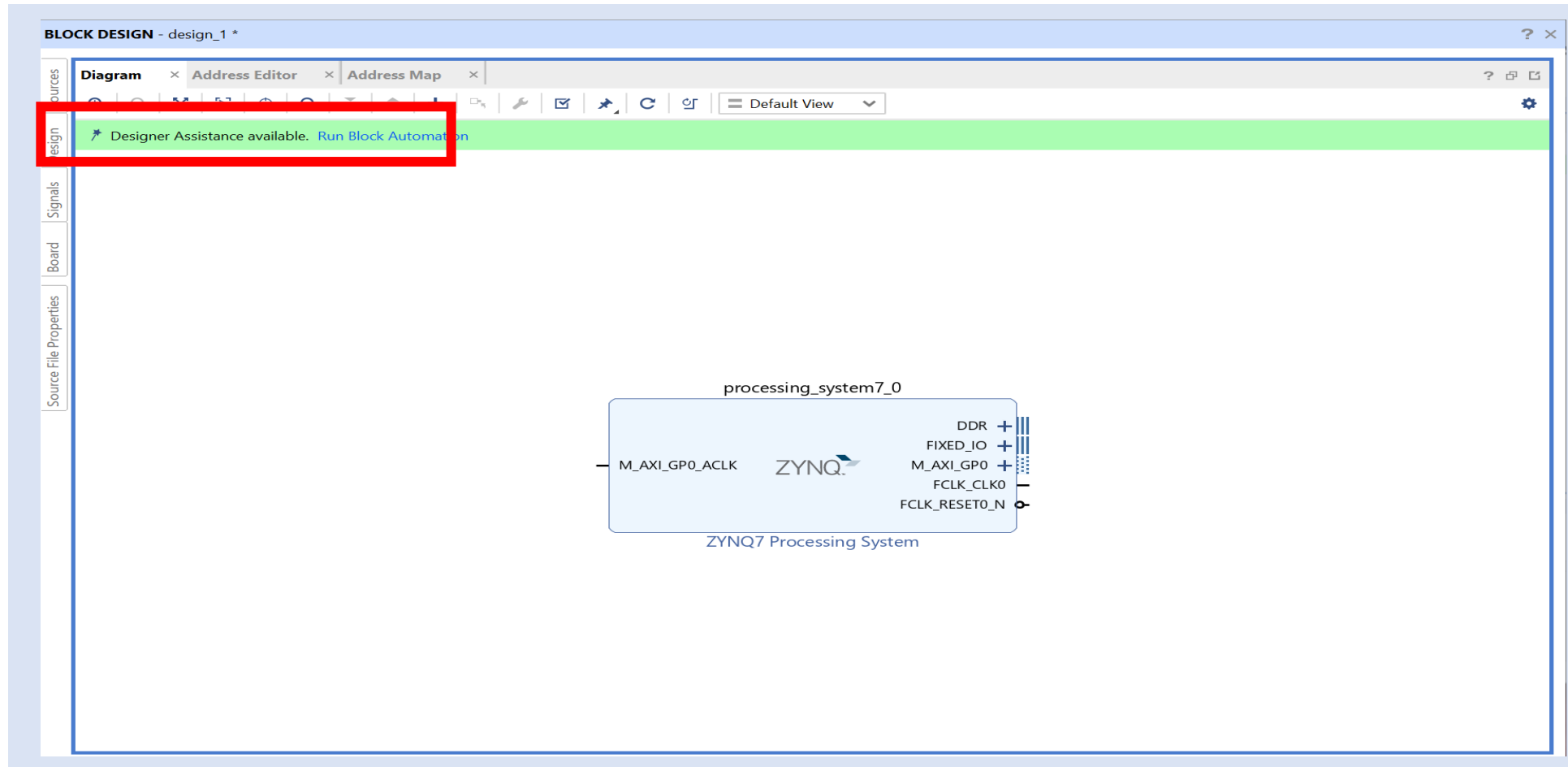
Lab 1: Understanding Vitis Project creation & Flow

Step 12 – In the search bar, type Ultra and select the Zynq UltraScale+ MPSoC block. Double click on this to insert the IP block.



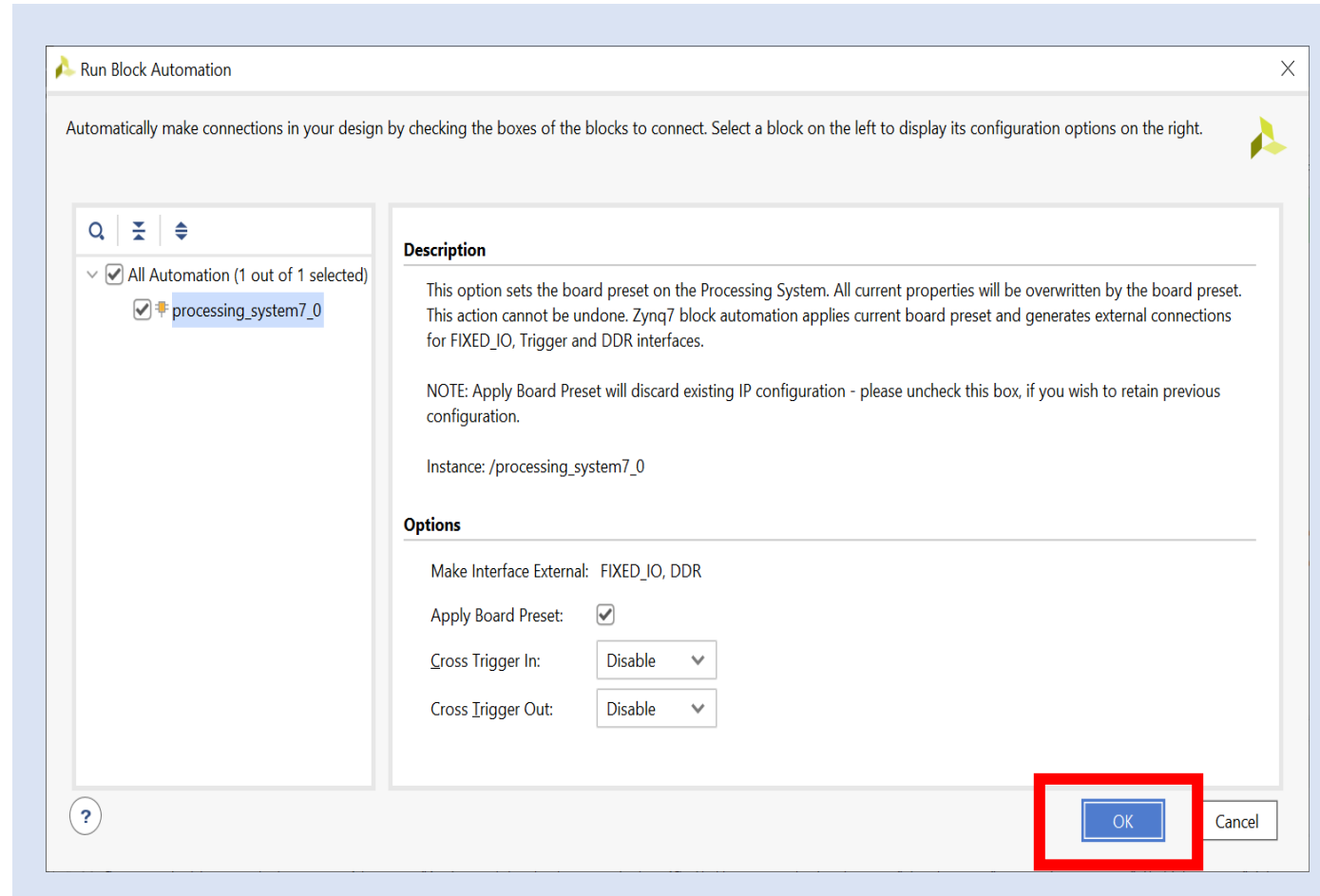
Lab 1: Understanding Vitis Project creation & Flow

Step 13 – Select Run Block Automation – This will configure the Processing System for the ArtyZ7 setting e.g. DDR timing, Clocking etc.



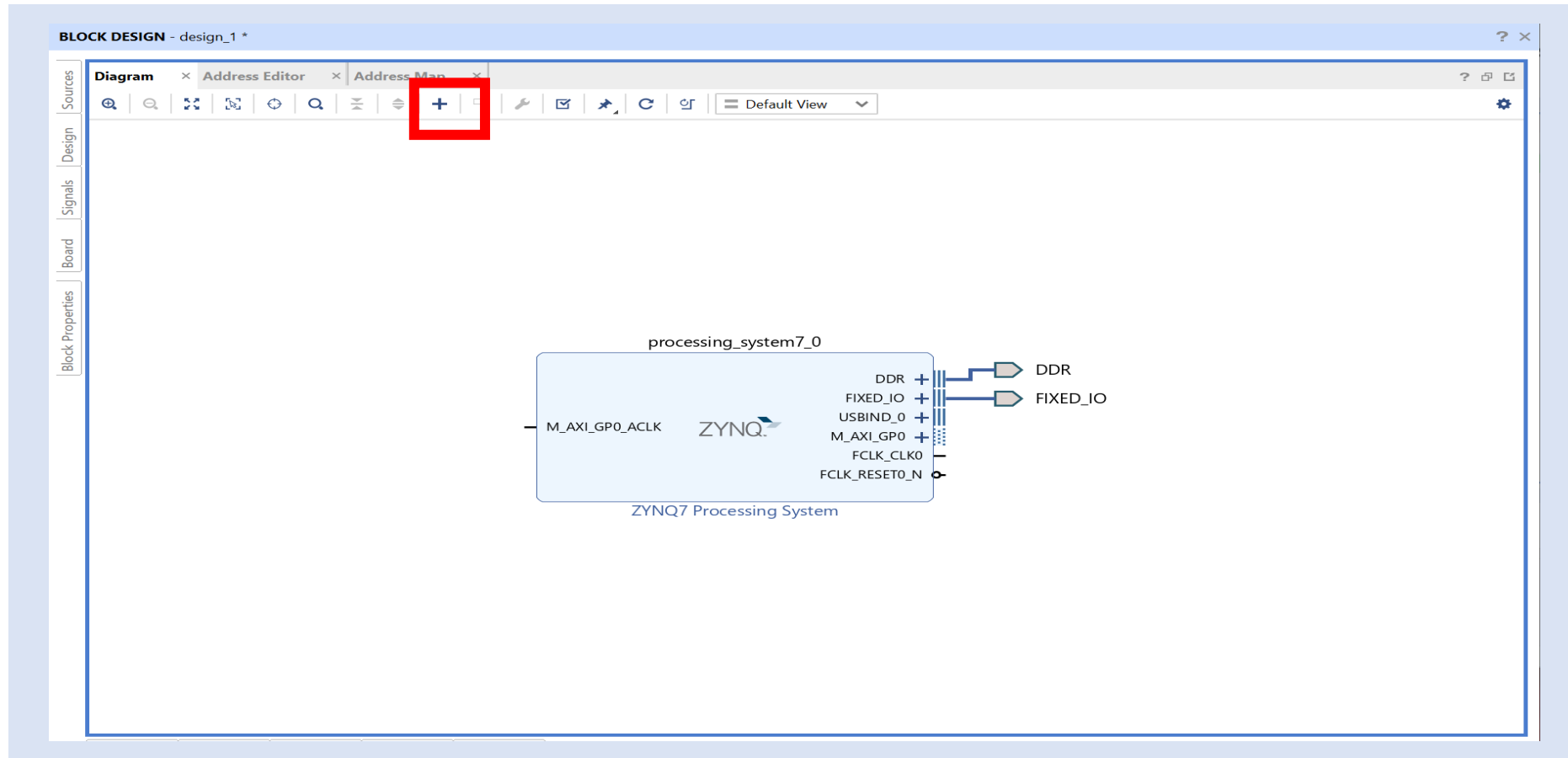
Lab 1: Understanding Vitis Project creation & Flow

Step 14 – Click on the OK button, the automation will run and configure the processing block for the ArtyZ7.



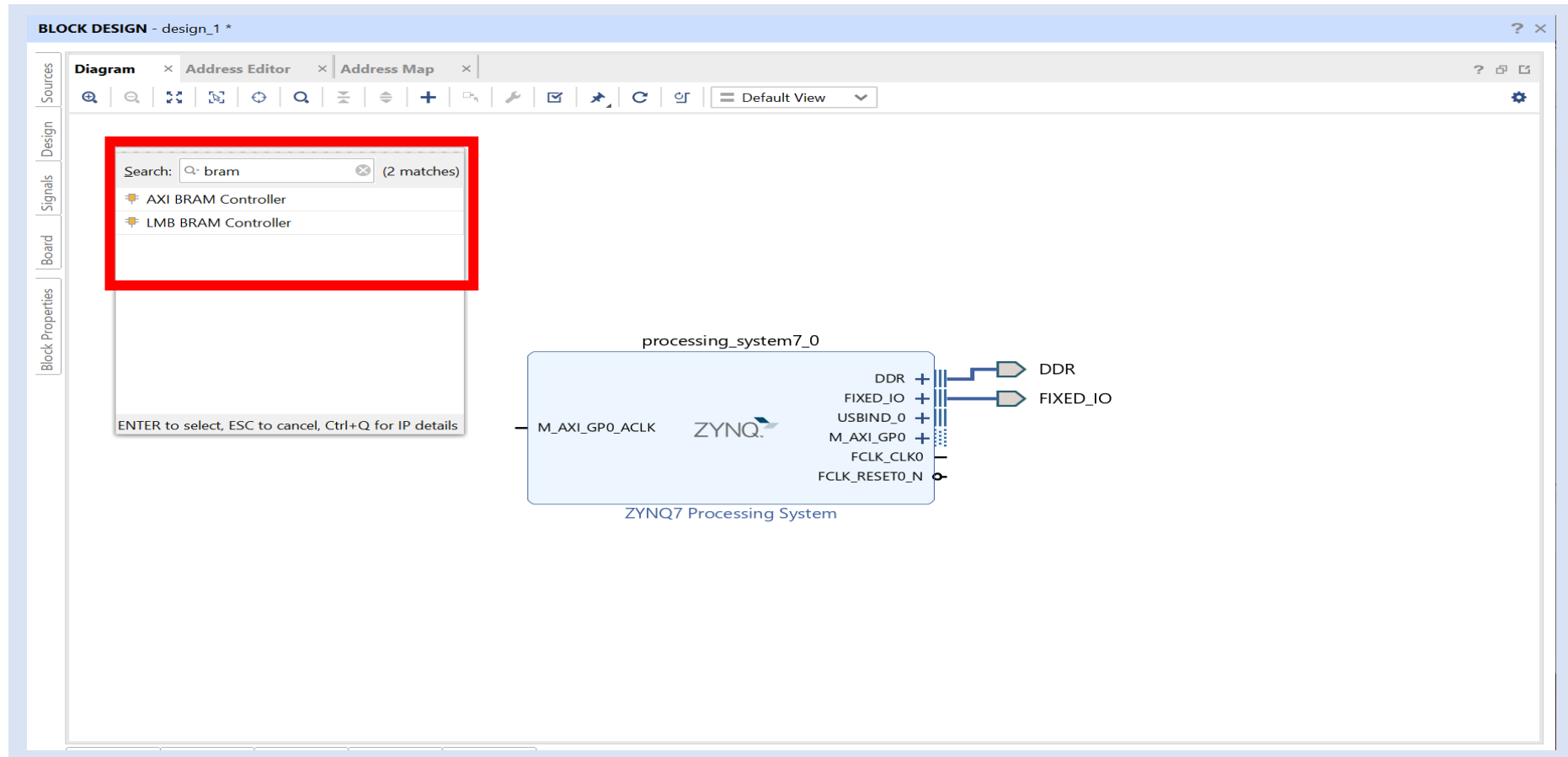
Lab 1: Understanding Vitis Project creation & Flow

Step 15 – Click on the + Symbol



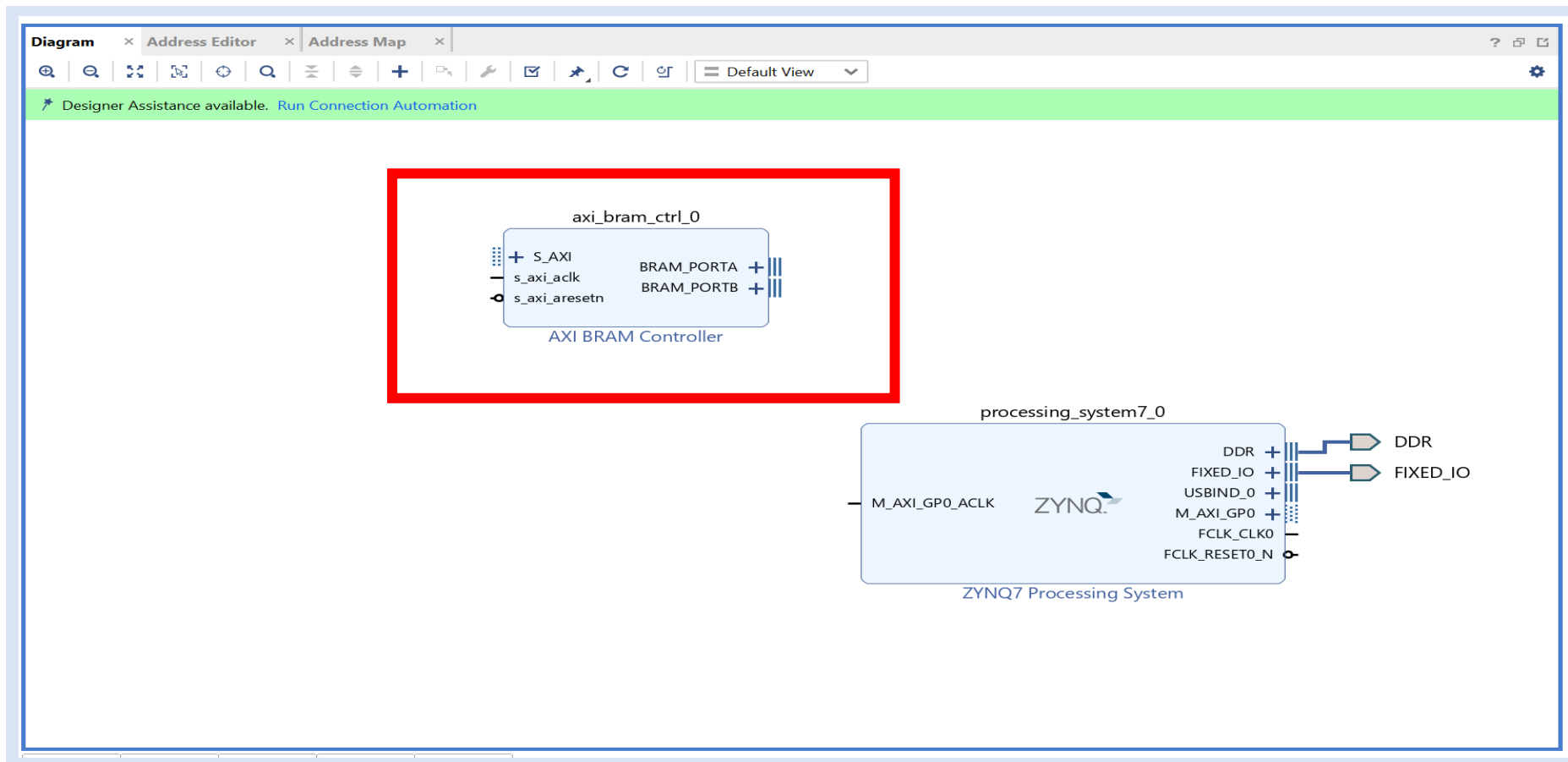
Lab 1: Understanding Vitis Project creation & Flow

Step 16 – Type in BRAM and double click on AXI BRAM Controller to add the IP



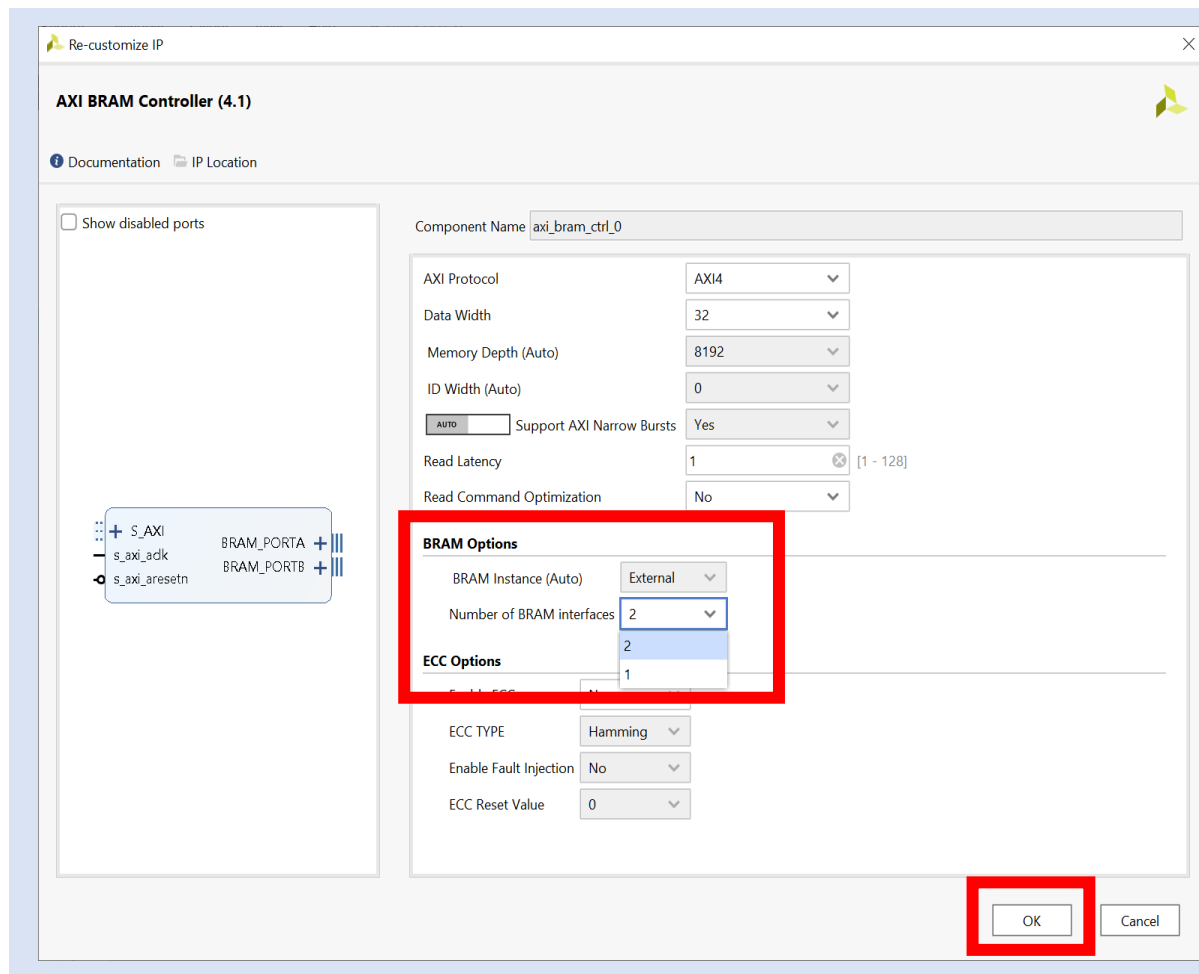
Lab 1: Understanding Vitis Project creation & Flow

Step 17 – Double click on the AXI BRAM Controller



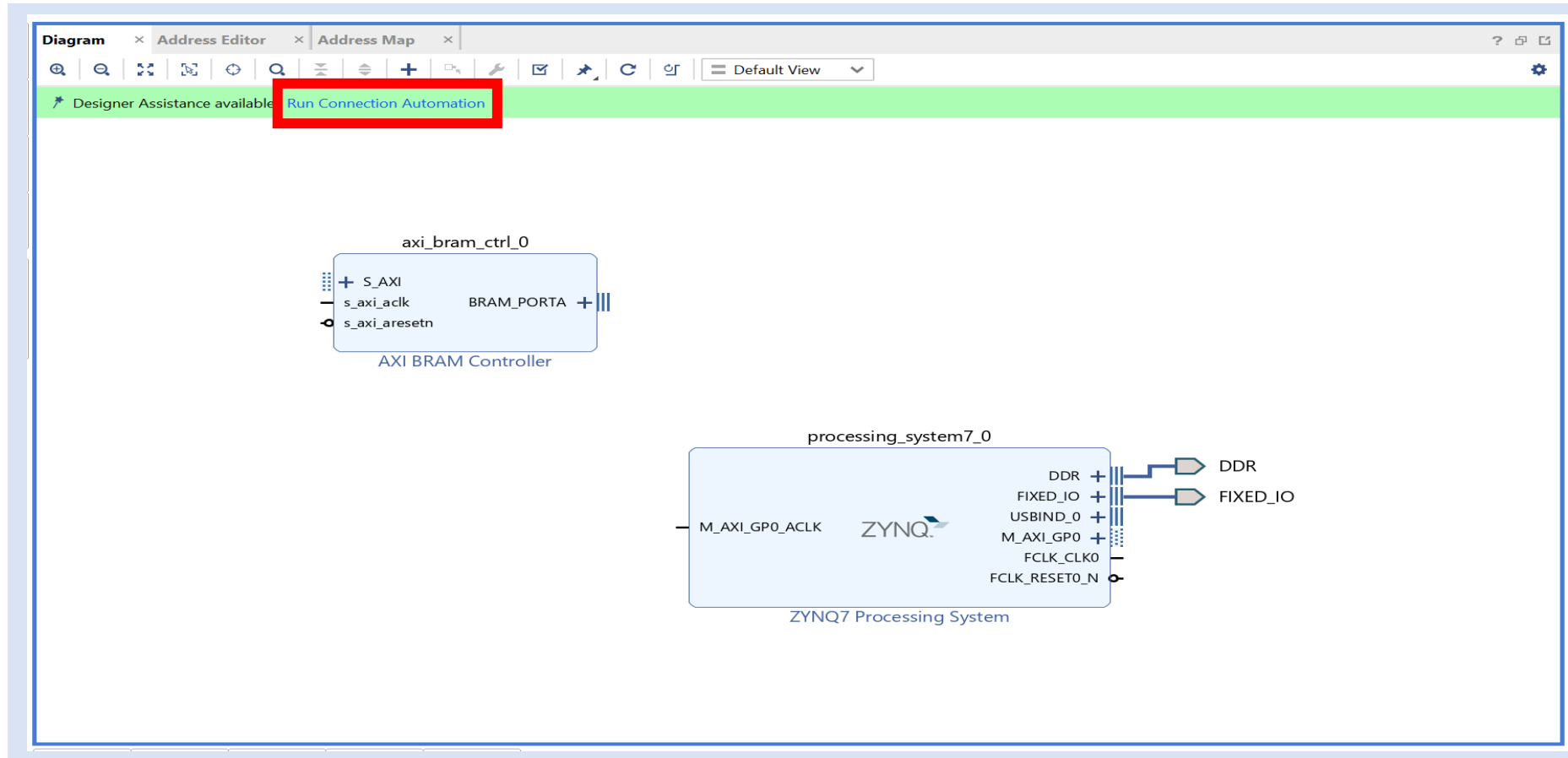
Lab 1: Understanding Vitis Project creation & Flow

Step 18 – Change the Number of BRAM Interfaces to 1, click OK



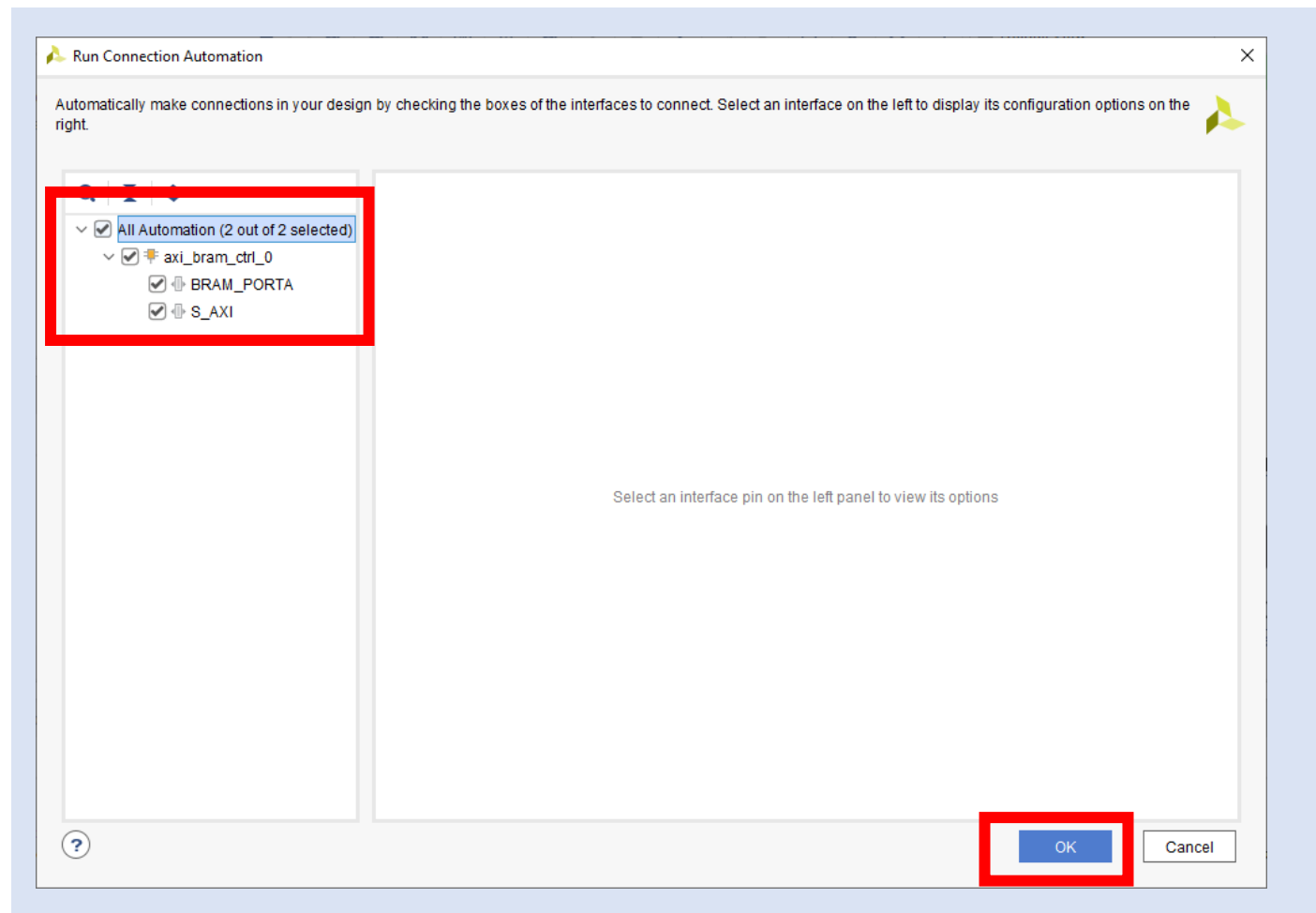
Lab 1: Understanding Vitis Project creation & Flow

Step 19 – Click on Run Connection Automation



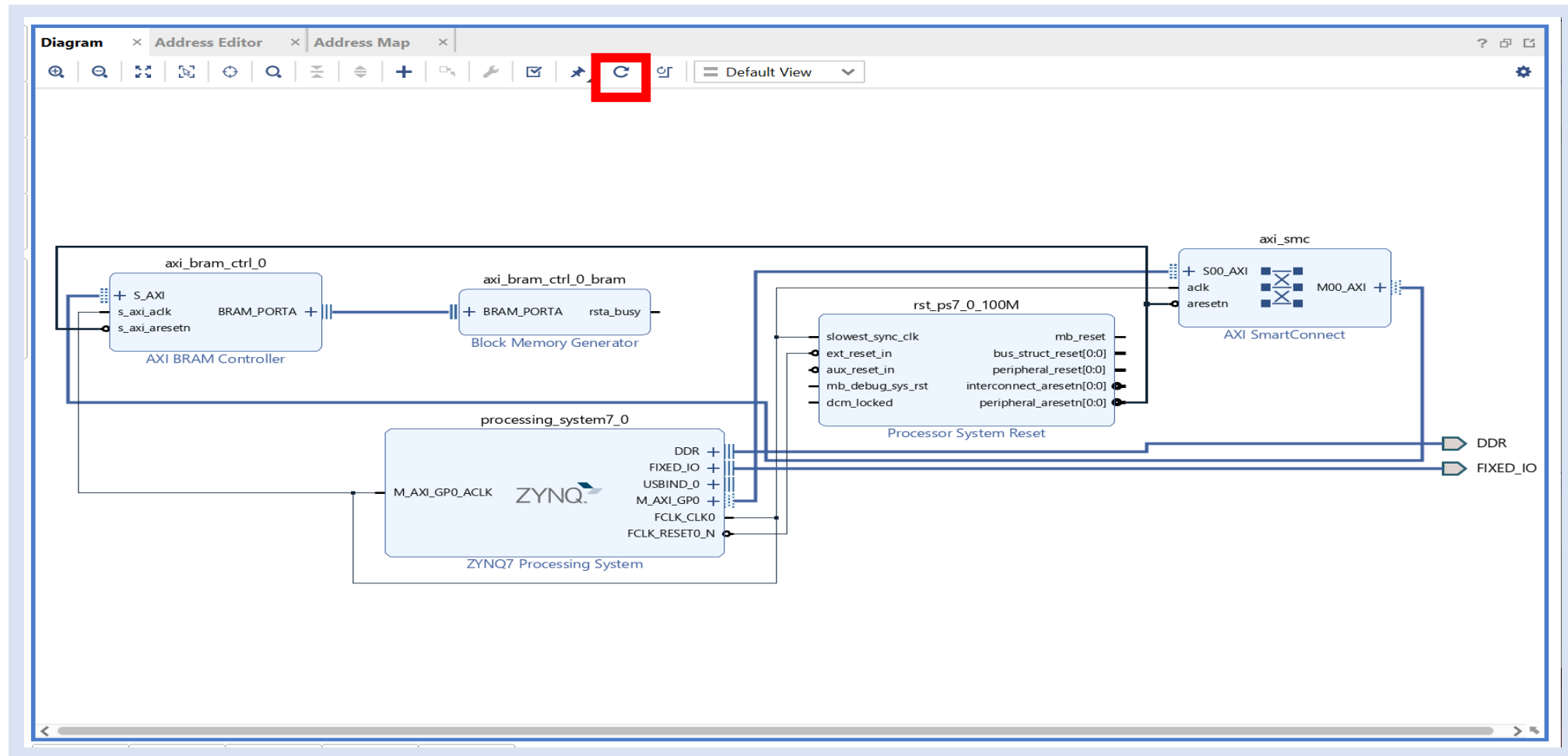
Lab 1: Understanding Vitis Project creation & Flow

Step 20 – Check all the boxes, click OK



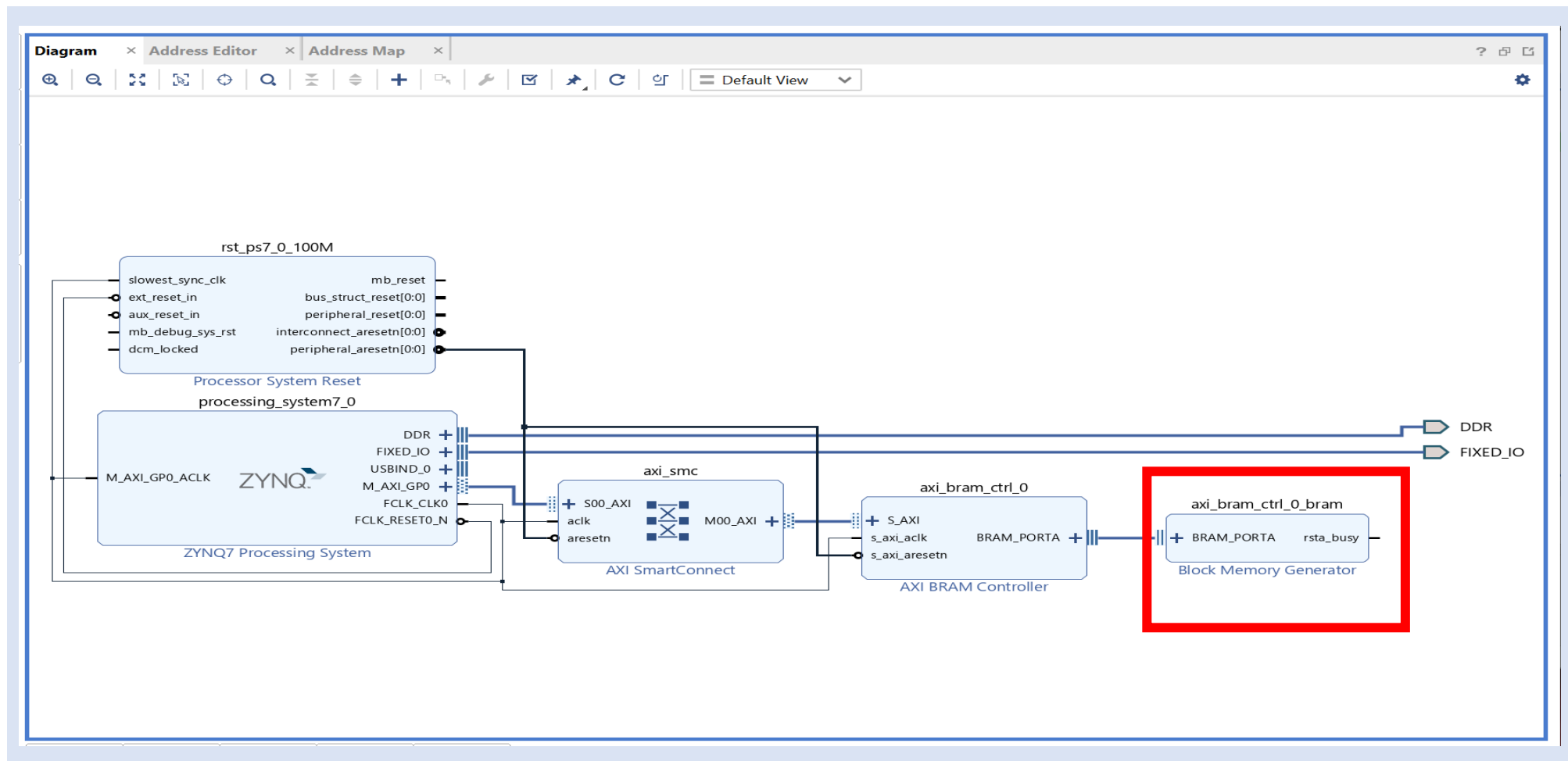
Lab 1: Understanding Vitis Project creation & Flow

Step 21 – Click on the Regenerate Layout – This will make the diagram more logical



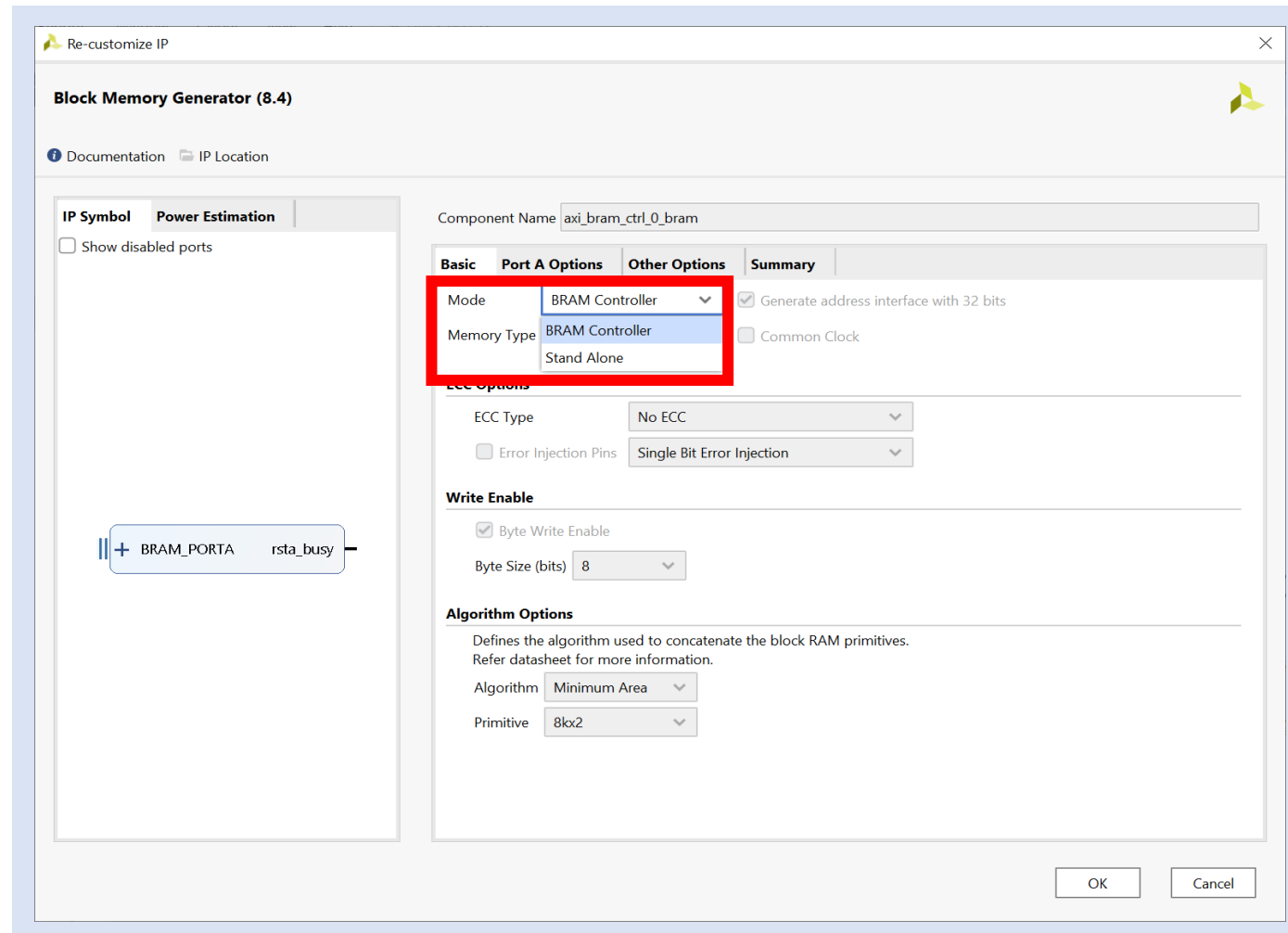
Lab 1: Understanding Vitis Project creation & Flow

Step 22 – Double Click on the BRAM Block



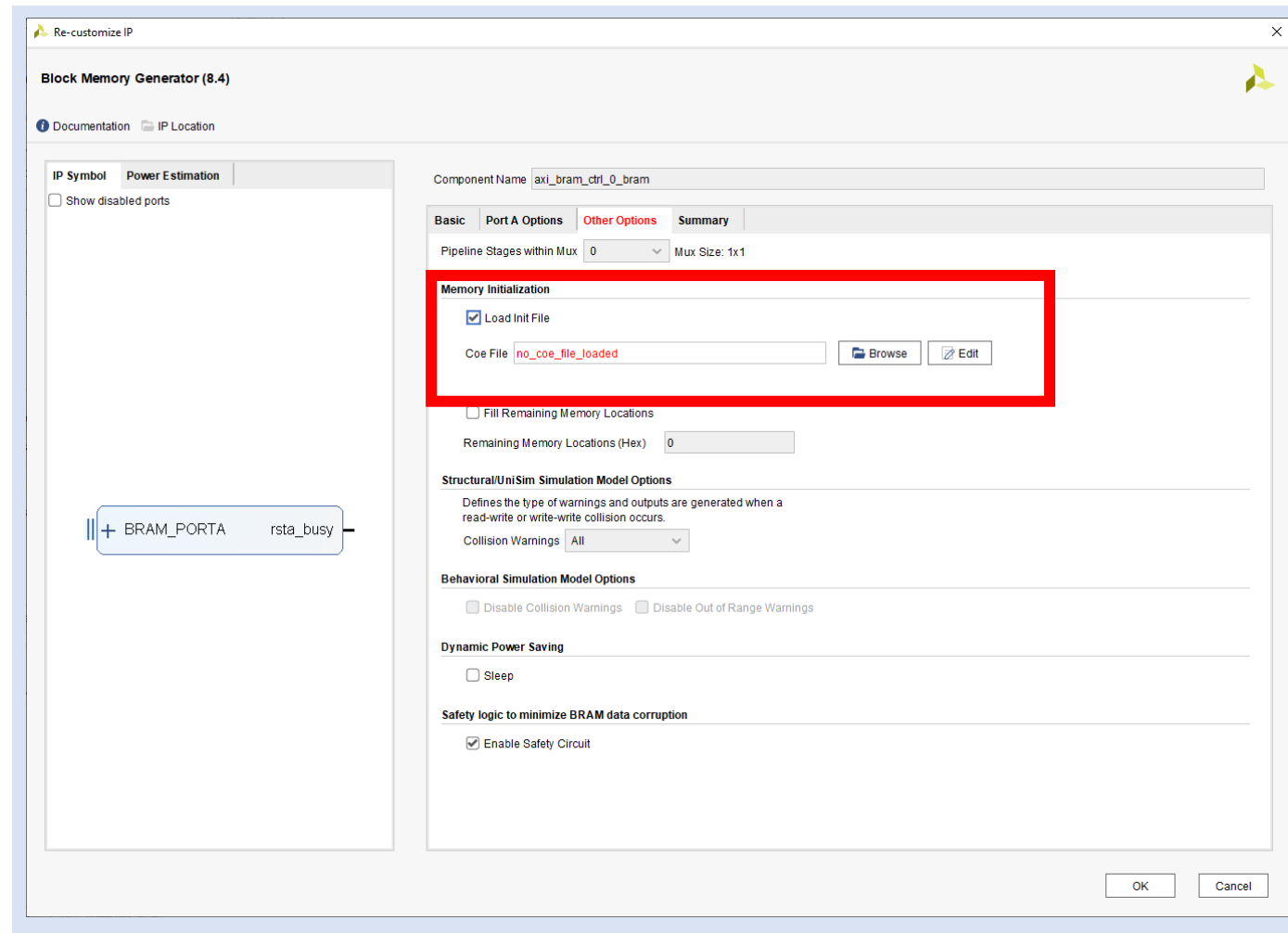
Lab 1: Understanding Vitis Project creation & Flow

Step 23 – Change the mode to Standalone, make sure check use 32 bit addressing is set



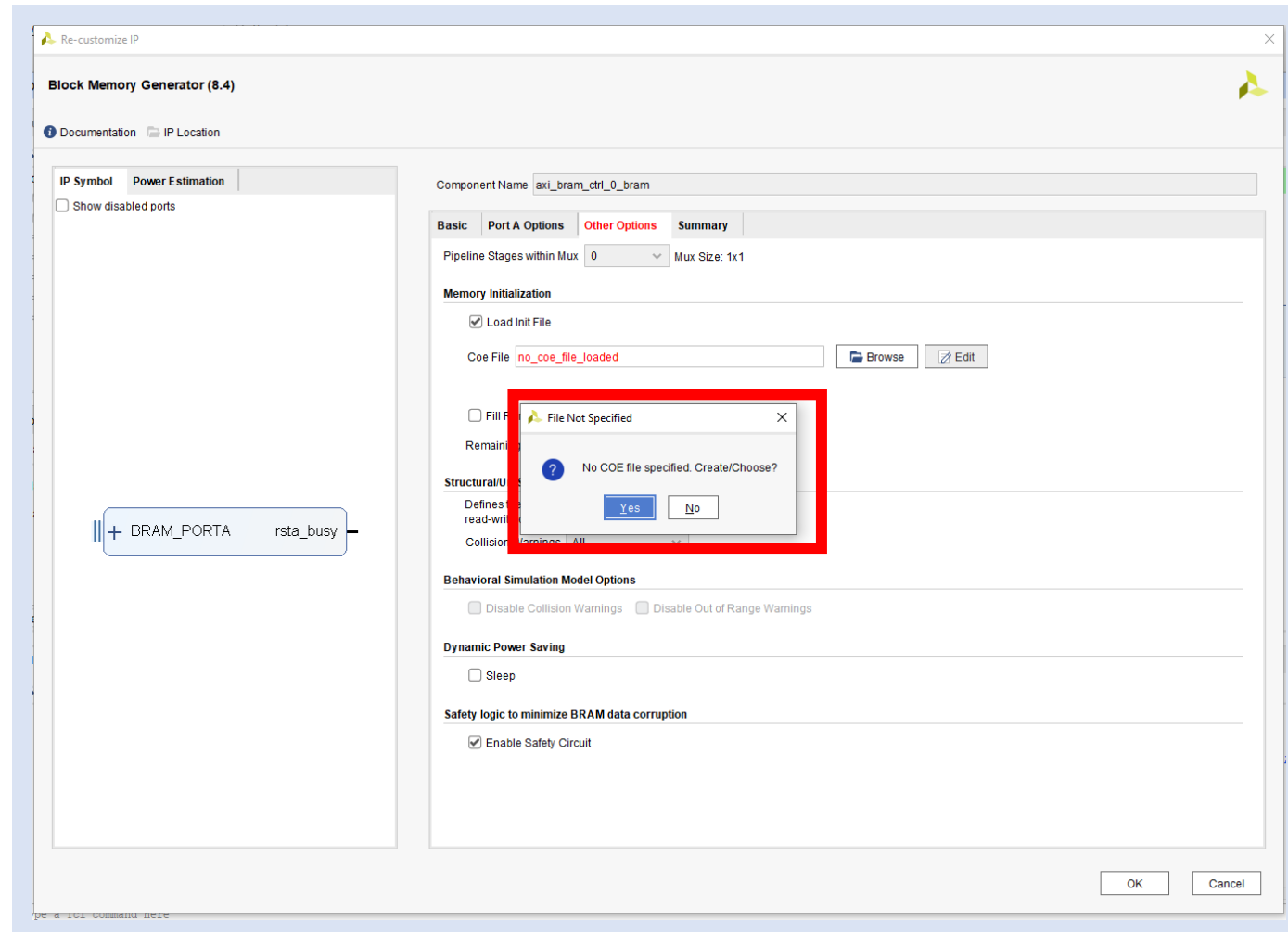
Lab 1: Understanding Vitis Project creation & Flow

Step 24 – Check load Init File, click on edit continue to step 25



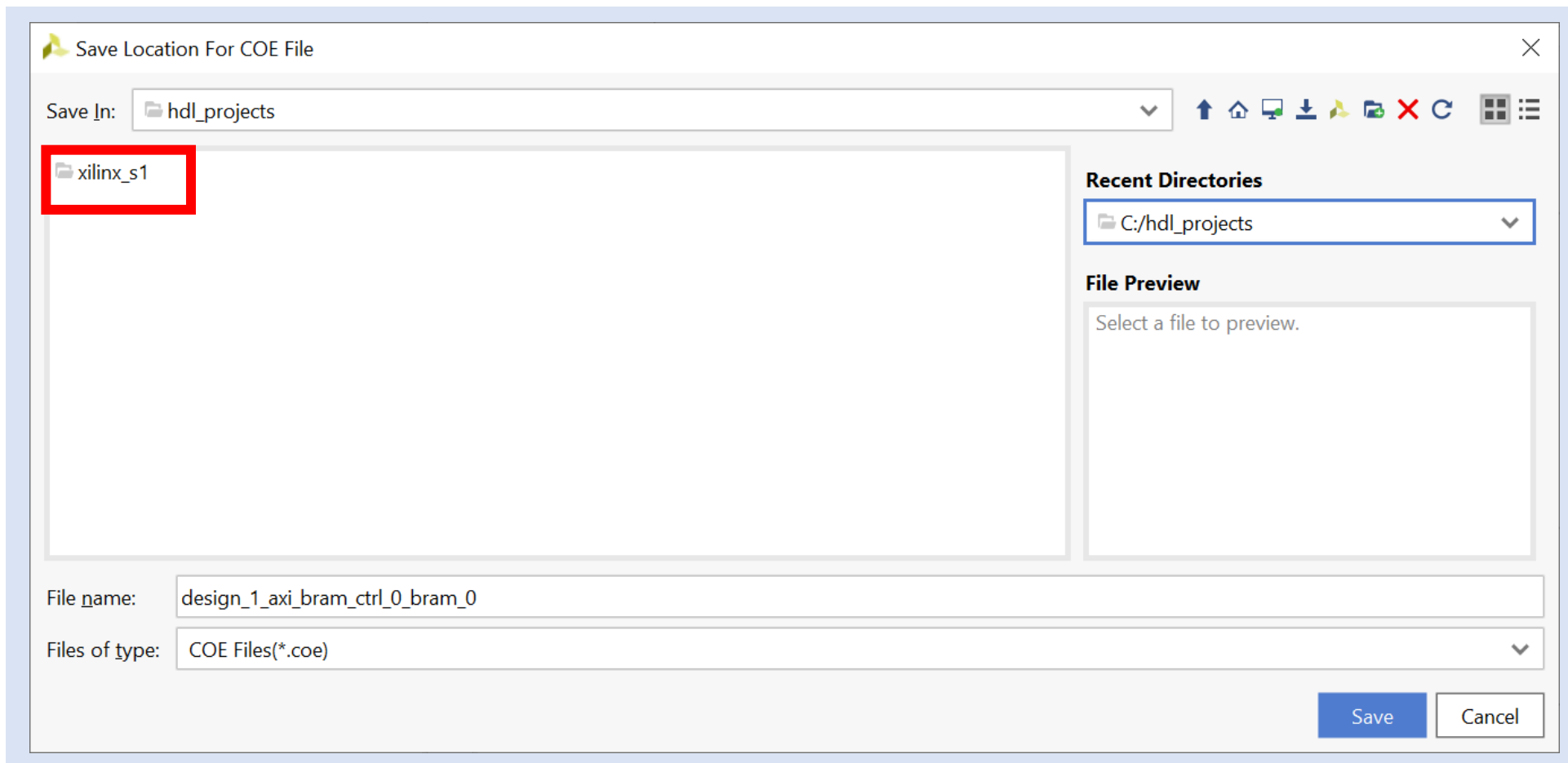
Lab 1: Understanding Vitis Project creation & Flow

Step 25 – Click Yes



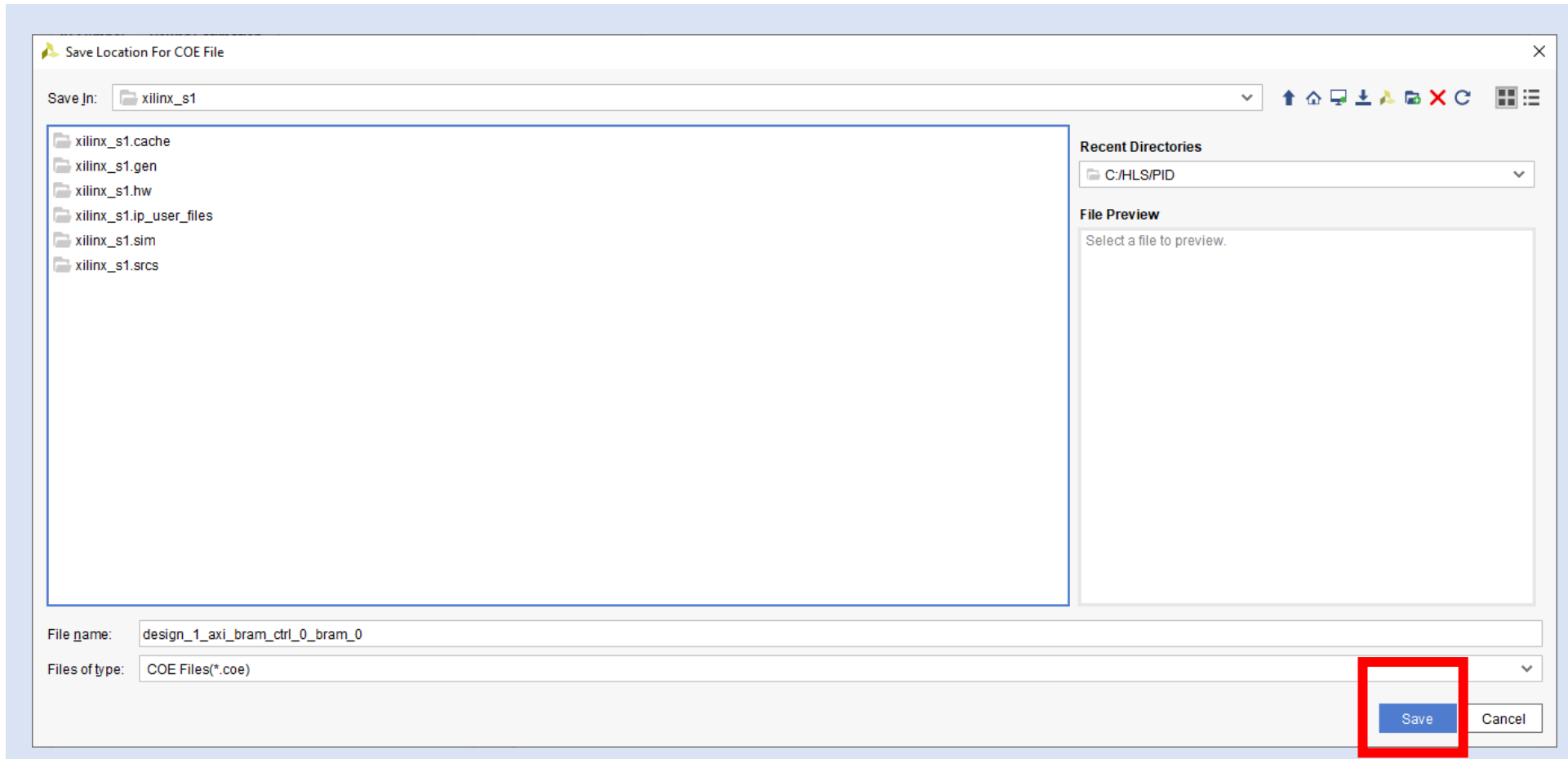
Lab 1: Understanding Vitis Project creation & Flow

Step 26 – Select the project folder and continue to step 27



Lab 1: Understanding Vitis Project creation & Flow

Step 27 – Select a location inside the project to save the file and select OK



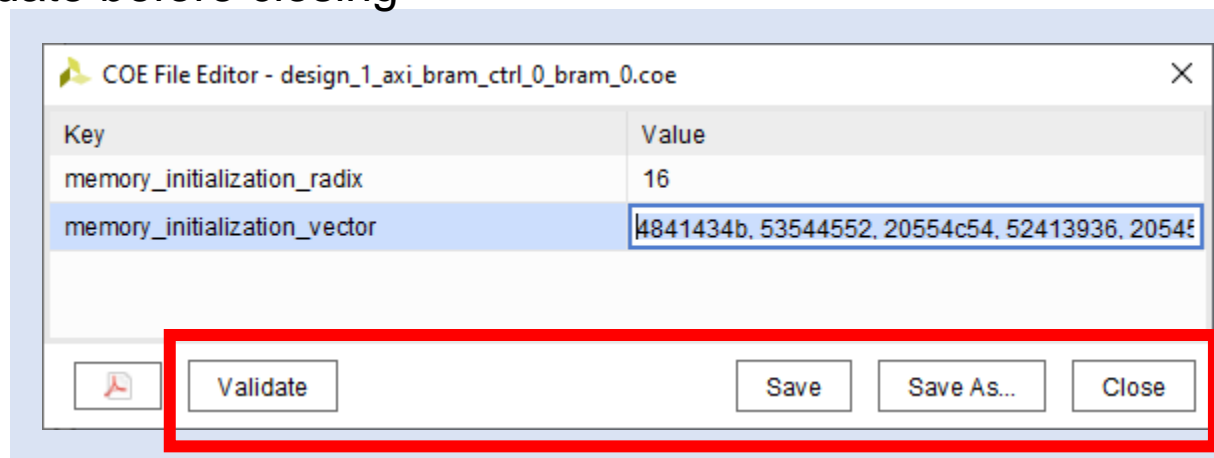
Lab 1: Understanding Vitis Project creation & Flow

Step 28 – Enter the following

Memory Initialization Radix = 16

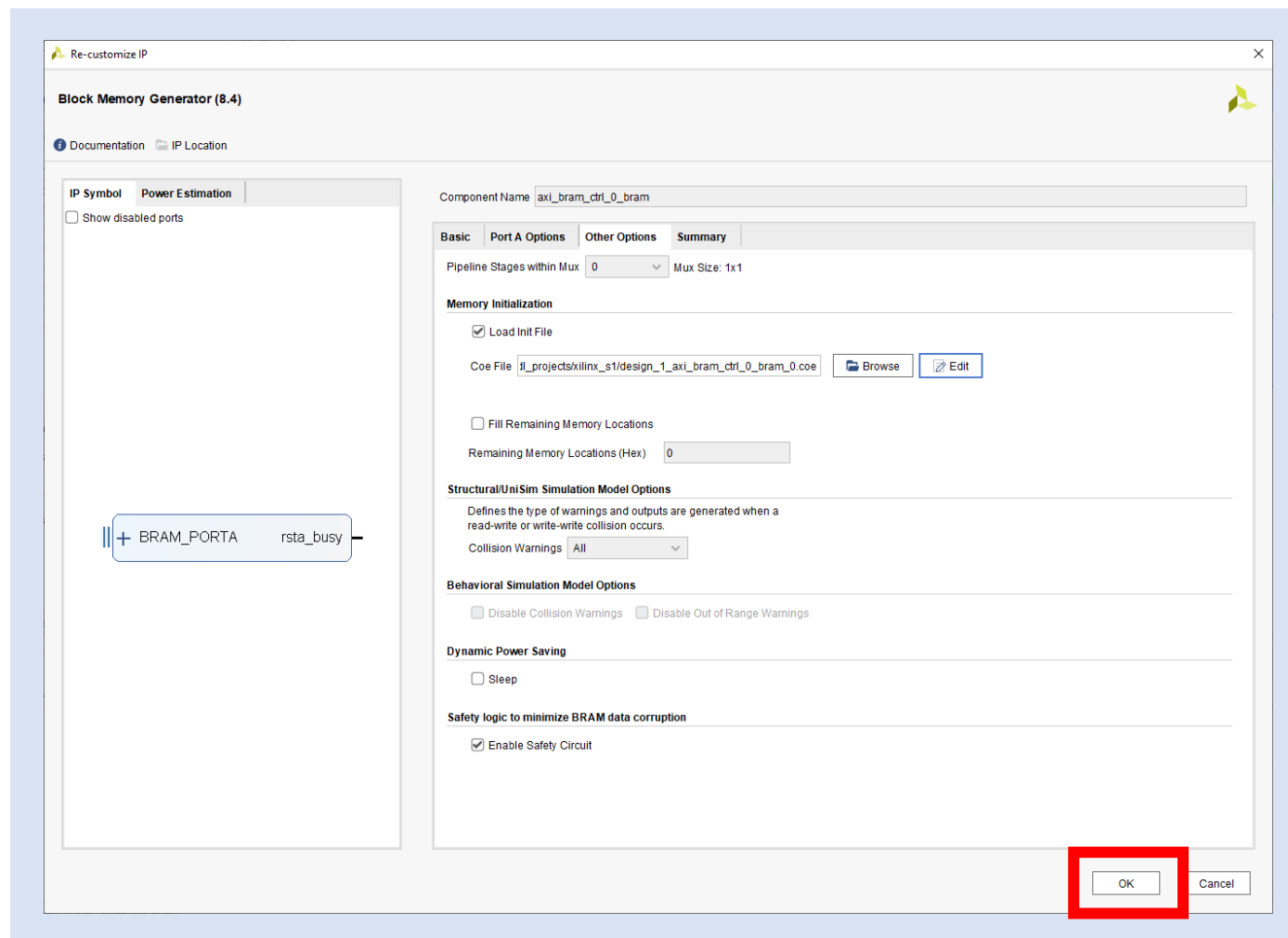
Memory Initialization Vector = 4841434b, 53544552, 20554c54, 52413936, 20545554, 4f524941, 0000004c,

Click Save, then Validate before closing



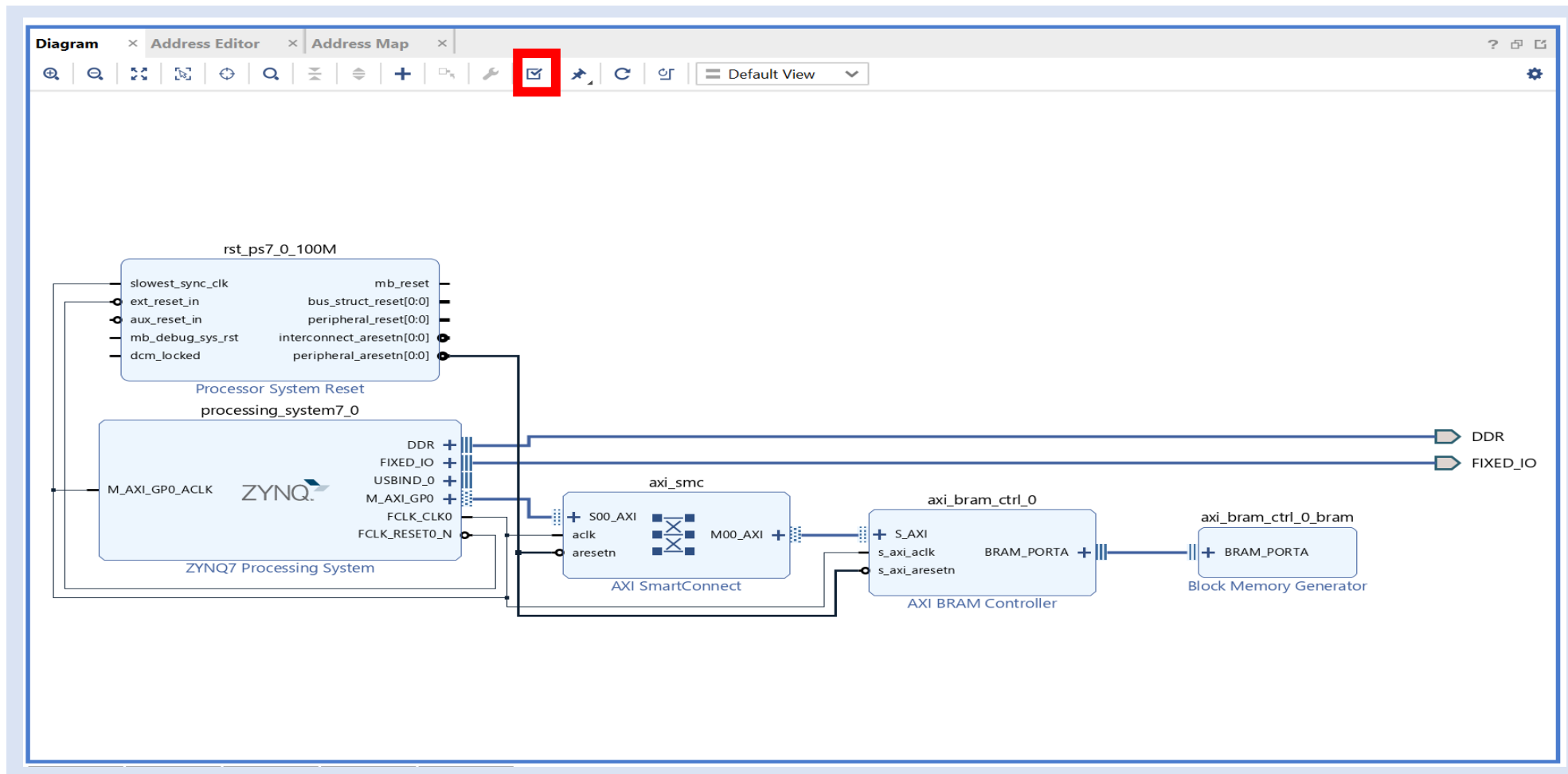
Lab 1: Understanding Vitis Project creation & Flow

Step 29 – Click OK



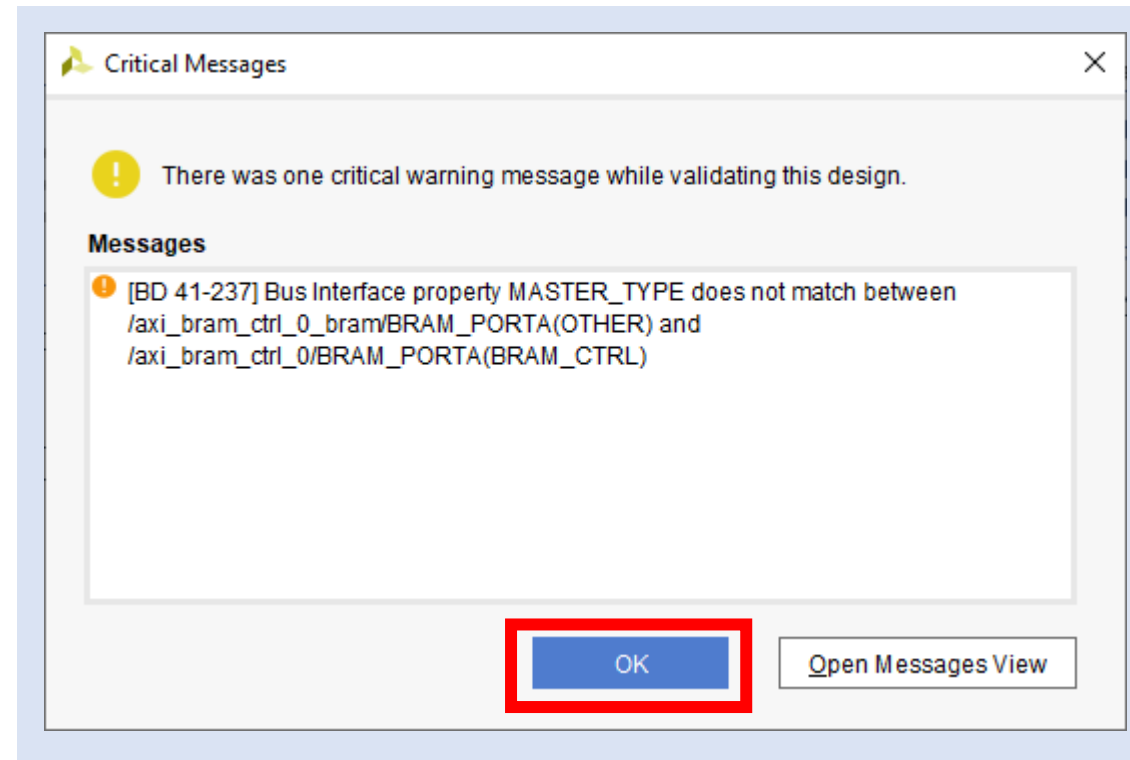
Lab 1: Understanding Vitis Project creation & Flow

Step 30 – Click on Validate the design



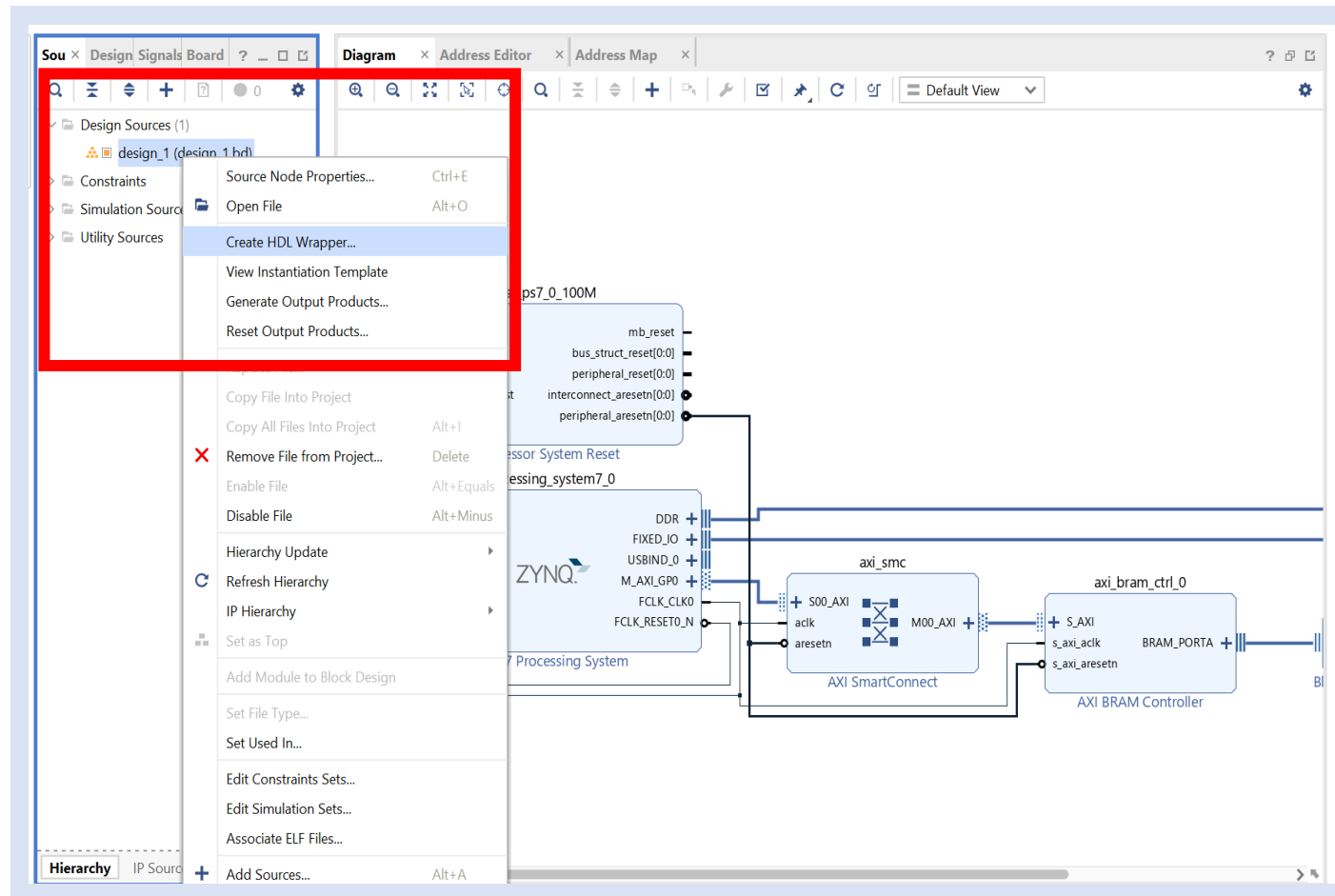
Lab 1: Understanding Vitis Project creation & Flow

Step 31 – Click On OK the warning is due to the change from AXI BRAM control



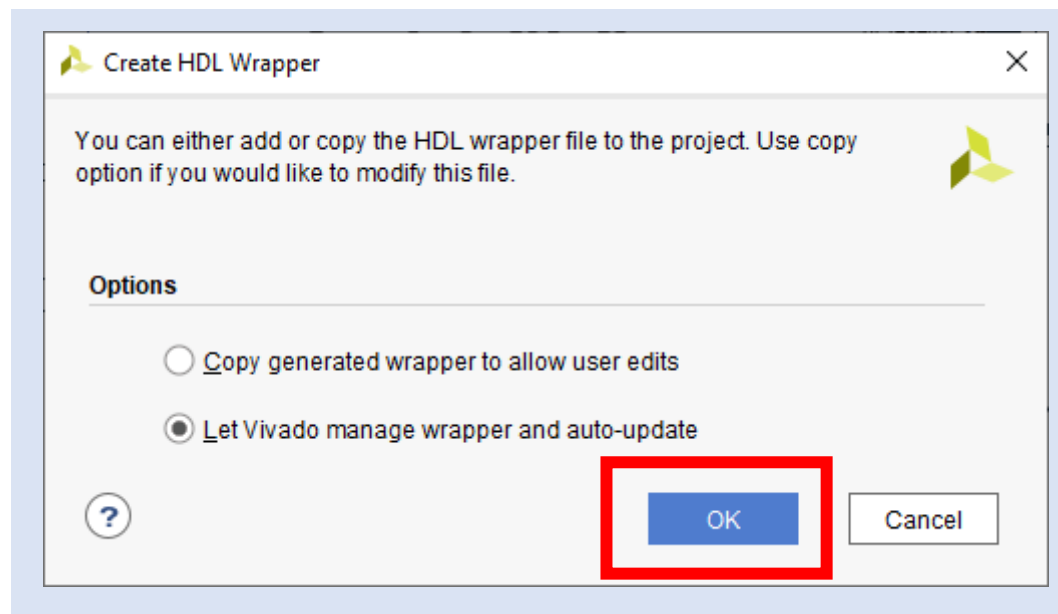
Lab 1: Understanding Vitis Project creation & Flow

Step 32 – In the sources tab, right click on the block diagram and select create HDL Wrapper



Lab 1: Understanding Vitis Project creation & Flow

Step 33 – leave the options as set and let Vivado manage the wrapper, click OK



Lab 1: Understanding Vitis Project creation & Flow

Step 34 – Click on Generate Bit Stream

The screenshot displays the Vivado 2021.1 software interface for a Vitis project. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The 'Layout' menu is highlighted with a red box, and the 'Generate Bitstream' icon is visible. The left sidebar shows the 'Flow Navigator' with the following sections:

- PROJECT MANAGER
 - Settings
 - Add Sources
 - Language Templates
 - IP Catalog
- IP INTEGRATOR
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- SIMULATION
 - Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG
 - Generate Bitstream
 - Open Hardware Manager

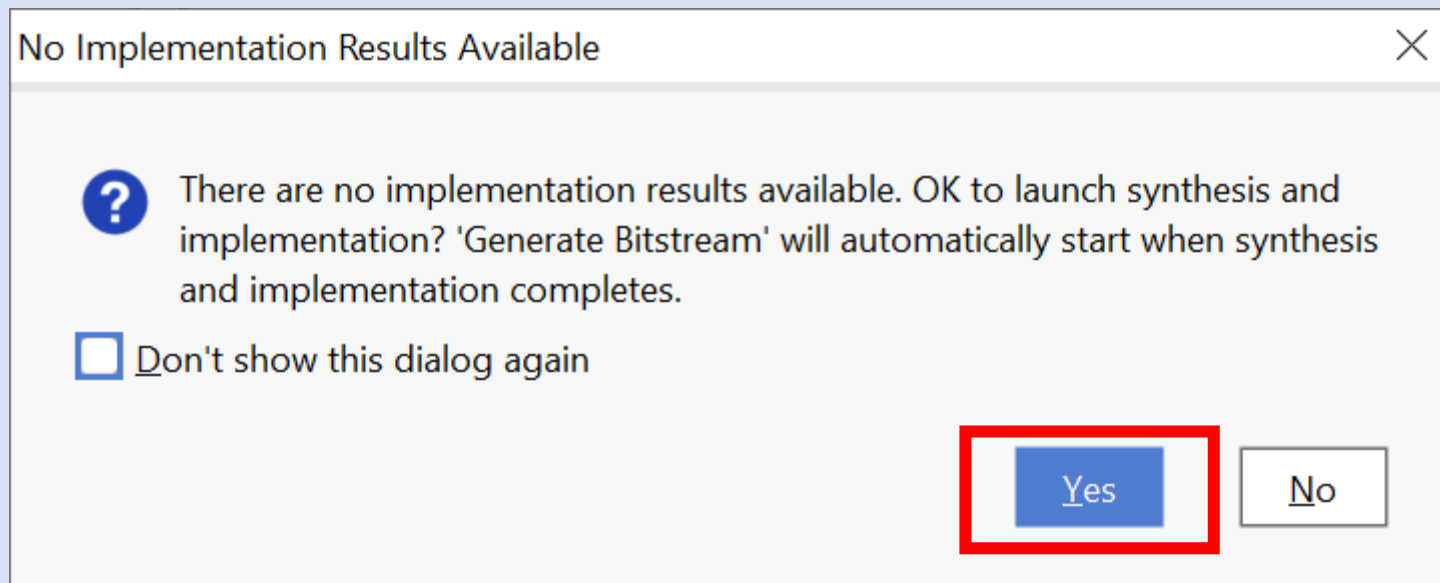
The main workspace shows a block diagram of the 'ZYNQ7 Processing System'. The diagram includes the following components and connections:

- rst_ps7_0_100M**: Processor System Reset block, connected to the ZYNQ7 Processing System.
- processing_system7_0**: The central ZYNQ7 Processing System block.
- axi_smartconnect**: AXI SmartConnect block, connected to the ZYNQ7 Processing System.
- axi_bram_ctrl_0**: AXI BRAM Controller block, connected to the AXI SmartConnect.
- axi_bram_ctrl_0_bram**: Block Memory Generator block, connected to the AXI BRAM Controller.
- DDR** and **FIXED_IO**: External memory and I/O blocks connected to the ZYNQ7 Processing System.

The bottom status bar shows the current interface connection: 'Interface Connection: processing_system7_0_DDR'.

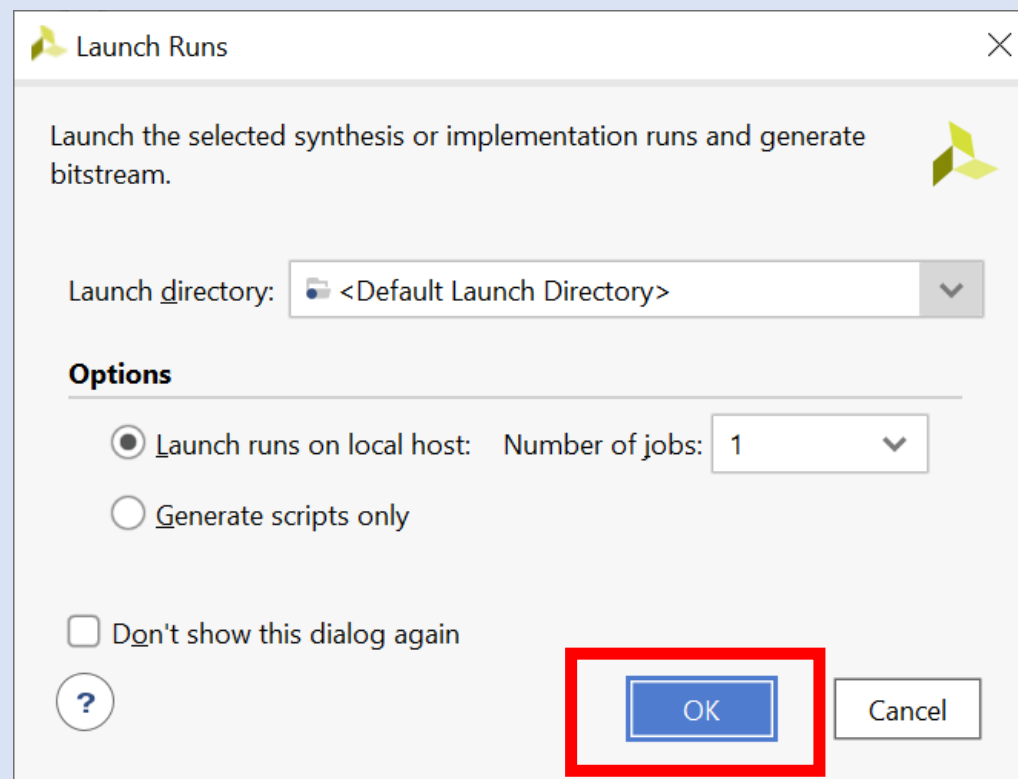
Lab 1: Understanding Vitis Project creation & Flow

Step 35 – Click yes



Lab 1: Understanding Vitis Project creation & Flow

Step 36 – Click Ok



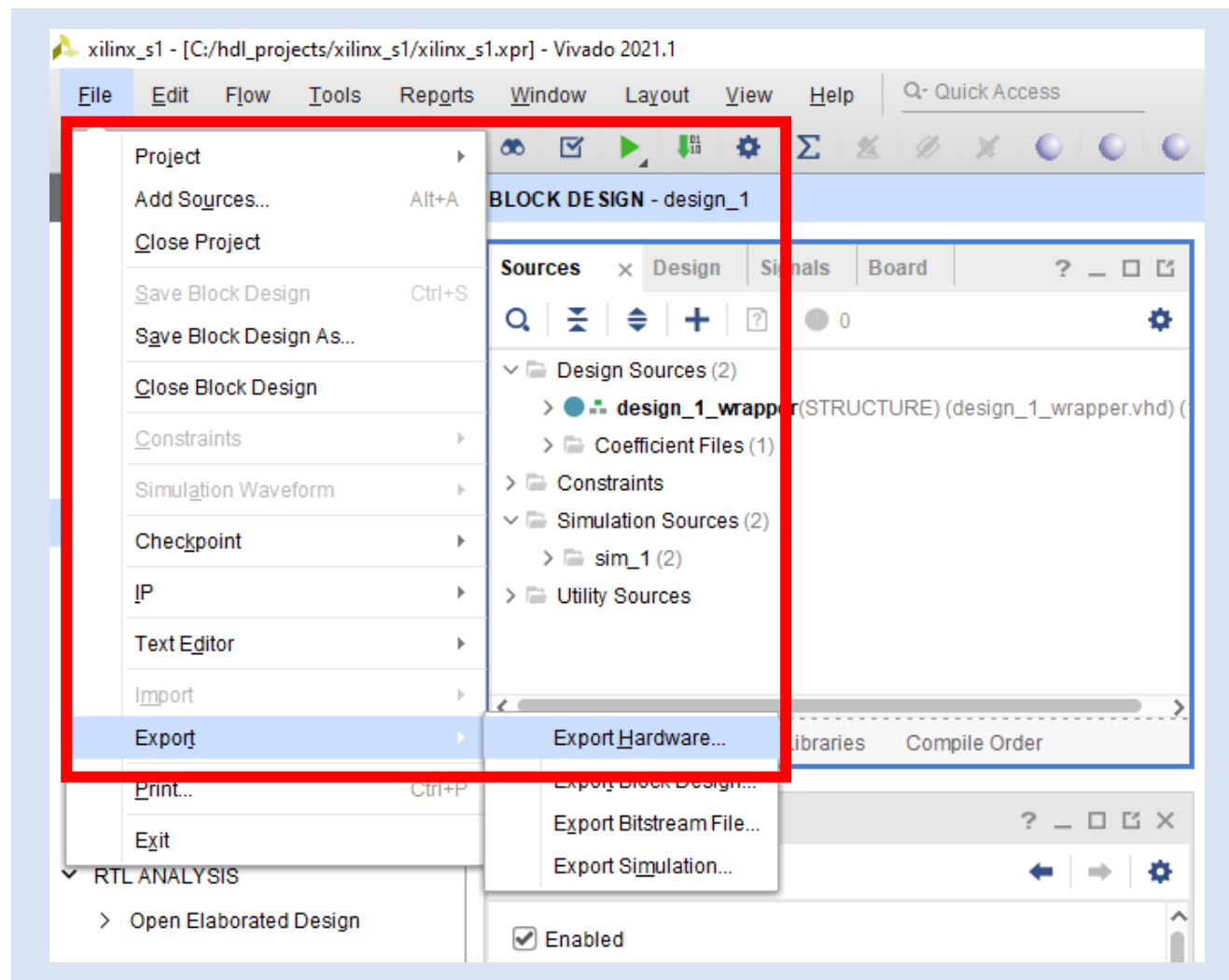
Lab 1: Understanding Vitis Project creation & Flow

Step 37 – Wait until the bit stream is complete

The screenshot displays the Vivado 2021.1 interface for a project named 'xilinx_s1'. The 'Flow Navigator' on the left shows the 'IMPLEMENTATION' phase, with 'Run Implementation' and 'Open Implemented Design' options. The 'Diagram' window on the right shows a block diagram of the design, including a ZYNQ7 Processing System, an AXI SmartConnect, an AXI BRAM Controller, and a Block Memory Generator. A red box highlights the 'write_bitstream Complete' message in the top right corner of the IDE, indicating that the bitstream generation is finished. The status bar at the bottom shows 'Design Runs'.

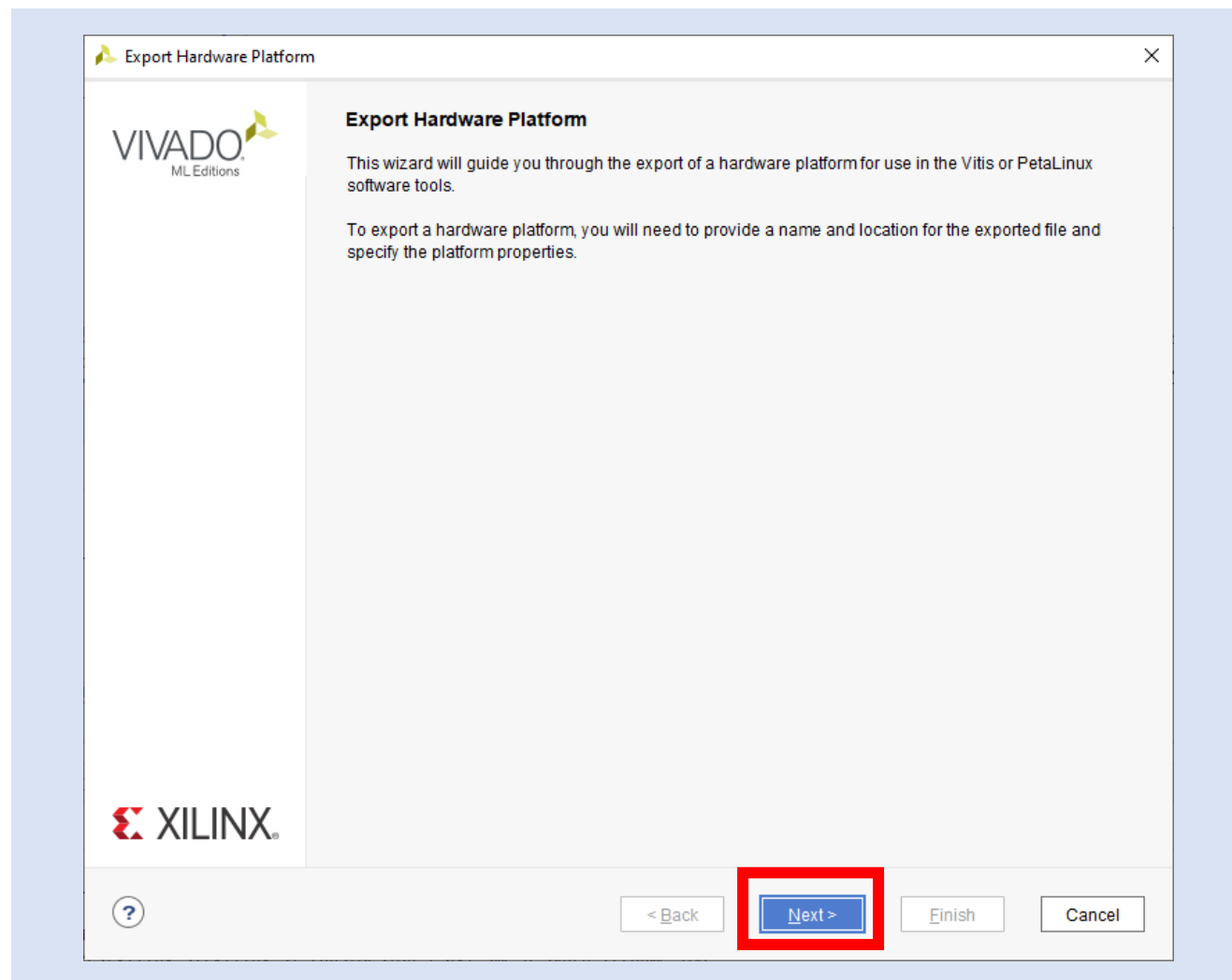
Lab 1: Understanding Vitis Project creation & Flow

Step 38 – From the File Menu select Export->Export Hardware



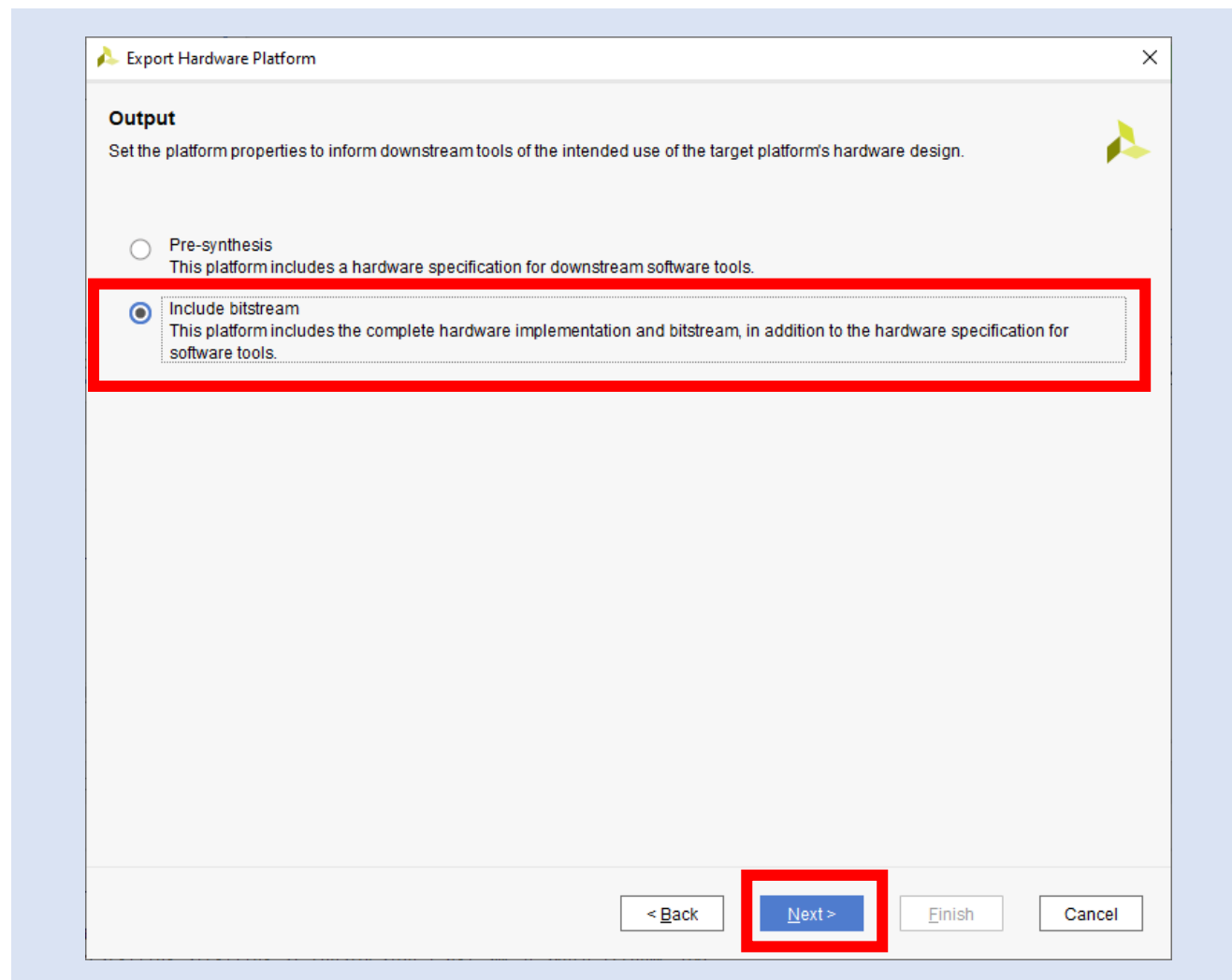
Lab 1: Understanding Vitis Project creation & Flow

Step 39 – Click on Next



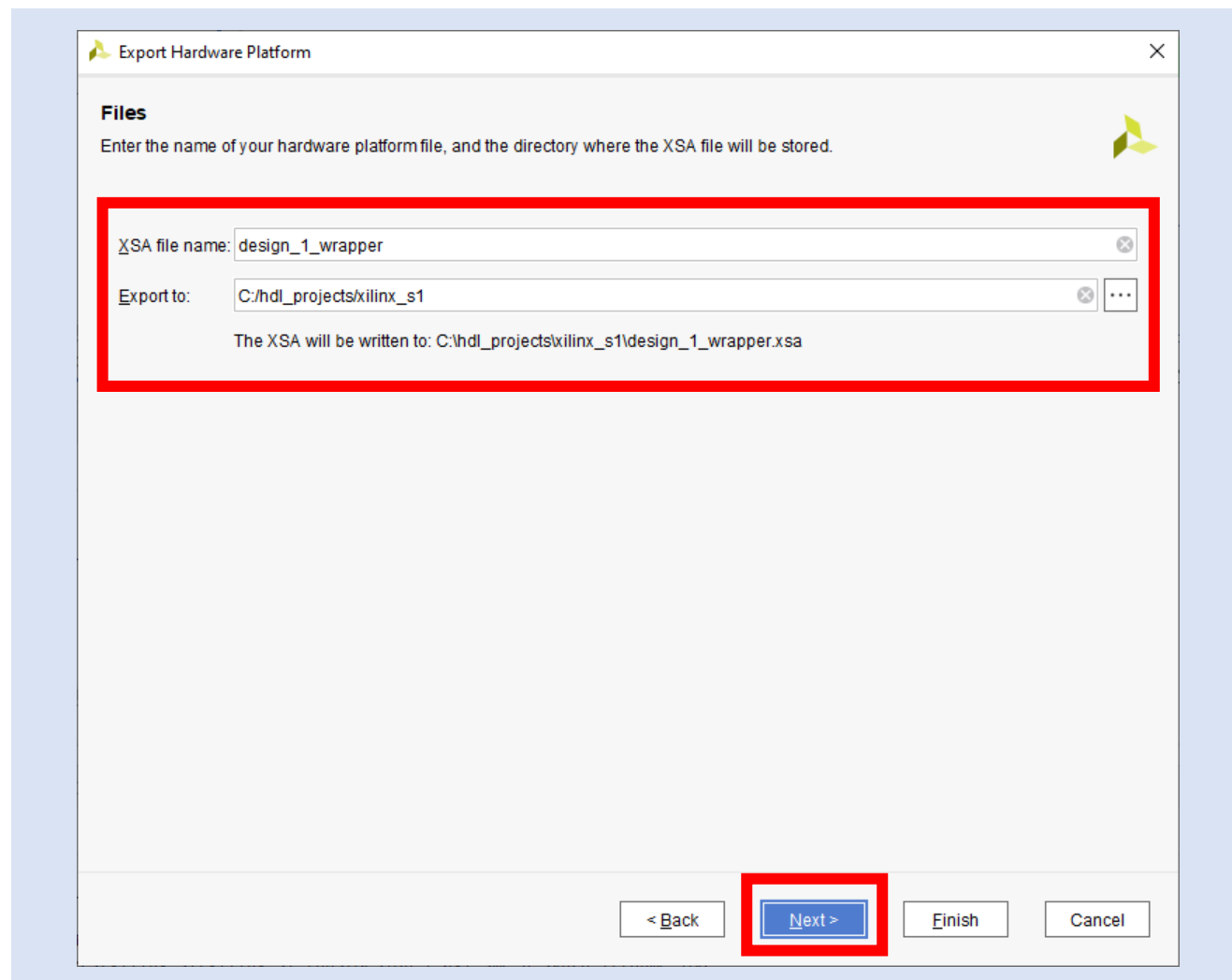
Lab 1: Understanding Vitis Project creation & Flow

Step 40 – Ensure Include Bitstream is selected and click OK



Lab 1: Understanding Vitis Project creation & Flow

Step 41 – Leave the defaults as is and click on 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:

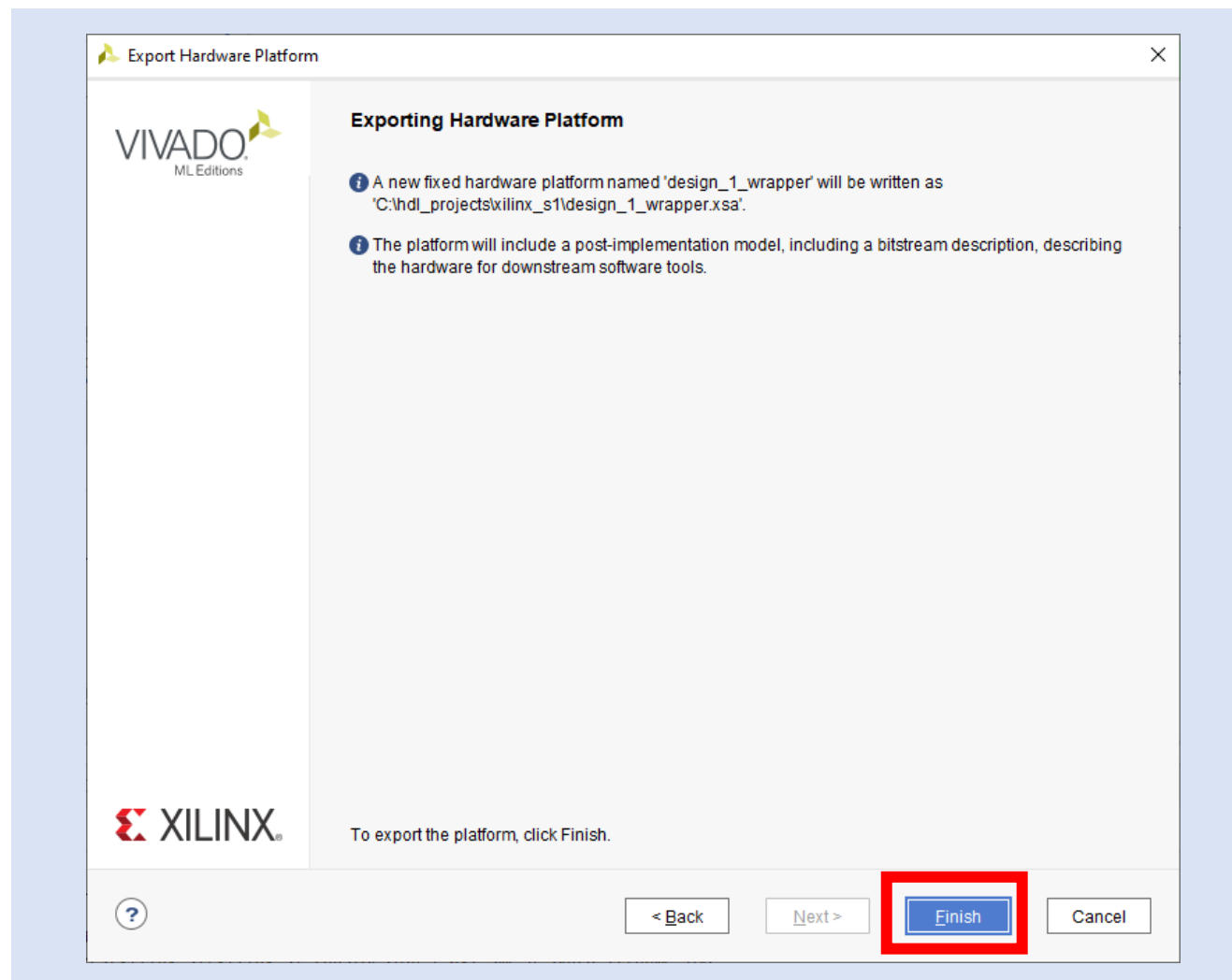
Export to:

The XSA will be written to: C:\hdl_projects\xilinx_s1\design_1_wrapper.xsa

< Back **Next >** Finish Cancel

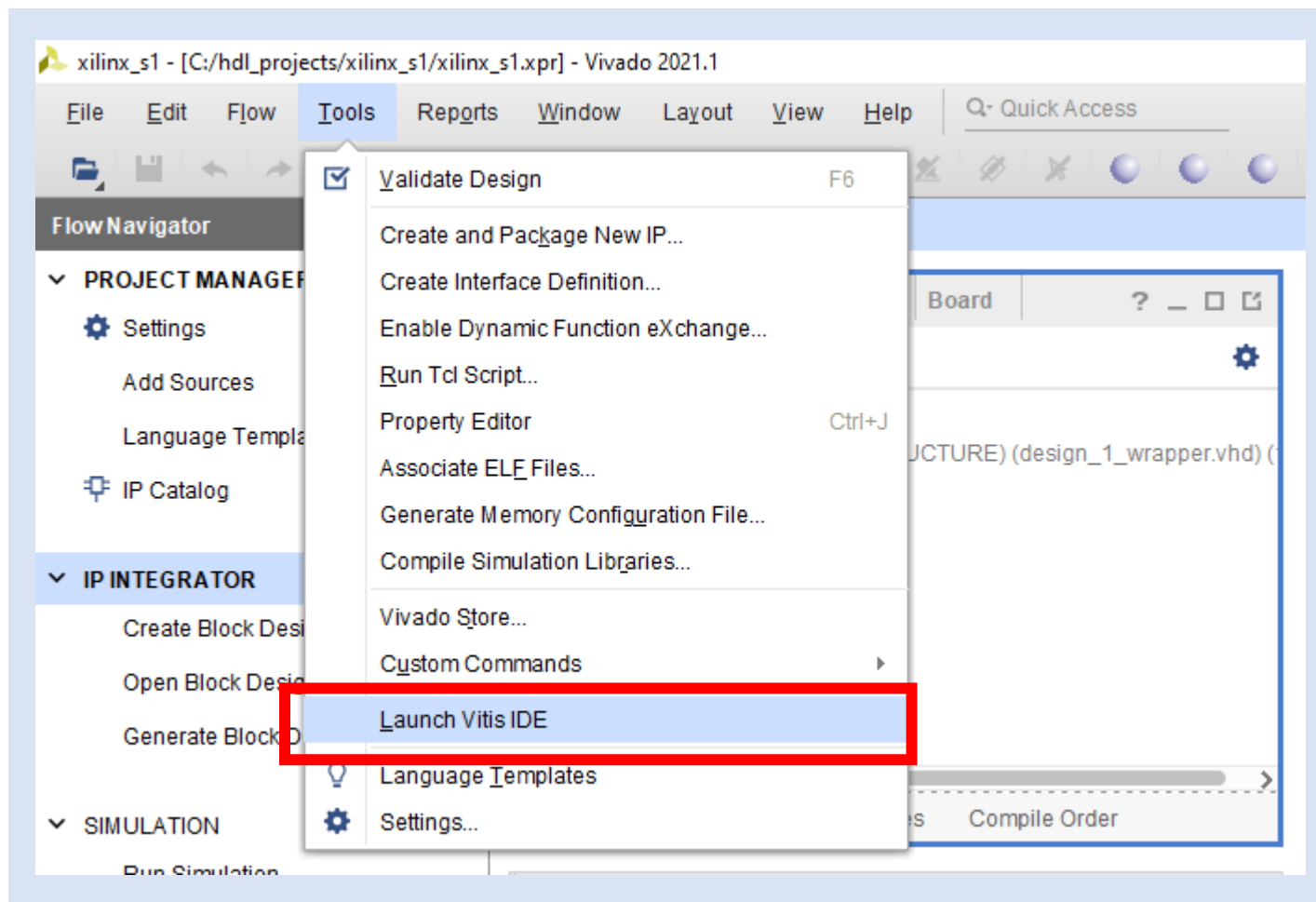
Lab 1: Understanding Vitis Project creation & Flow

Step 42 – Click Finish



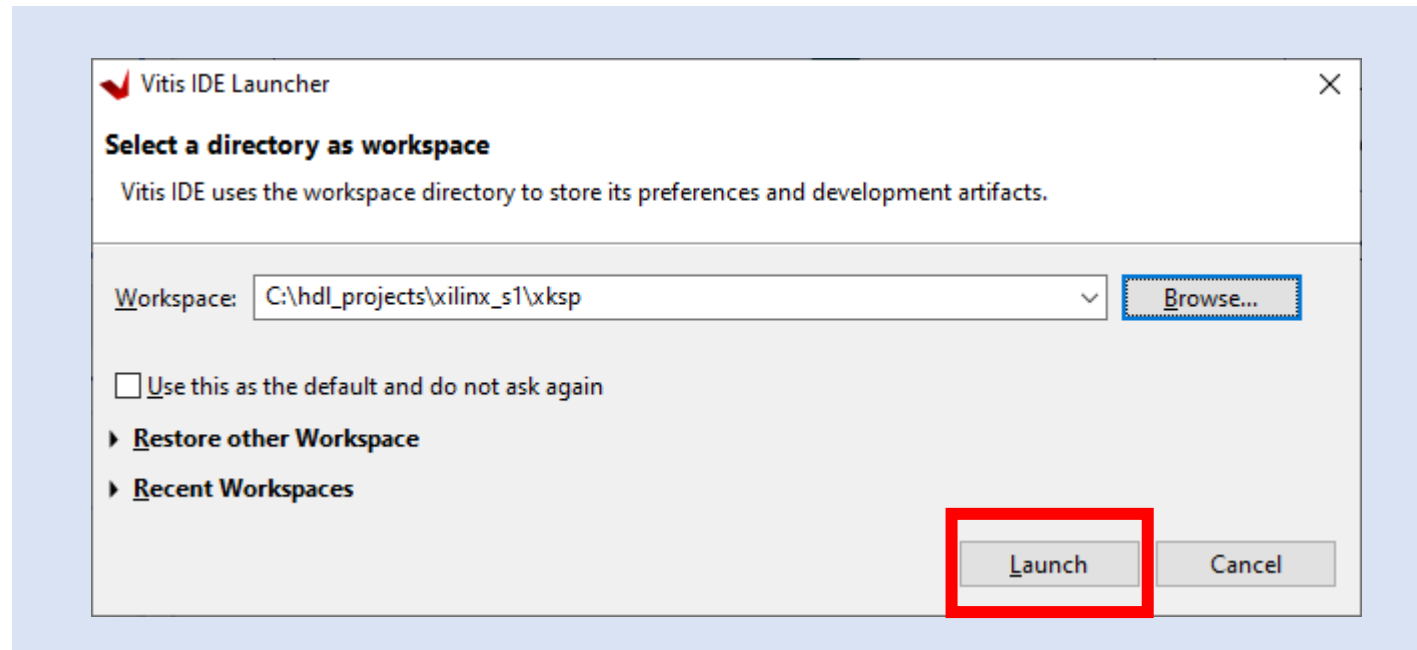
Lab 1: Understanding Vitis Project creation & Flow

Step 43 – From the Tools menu select Launch Vitis IDE



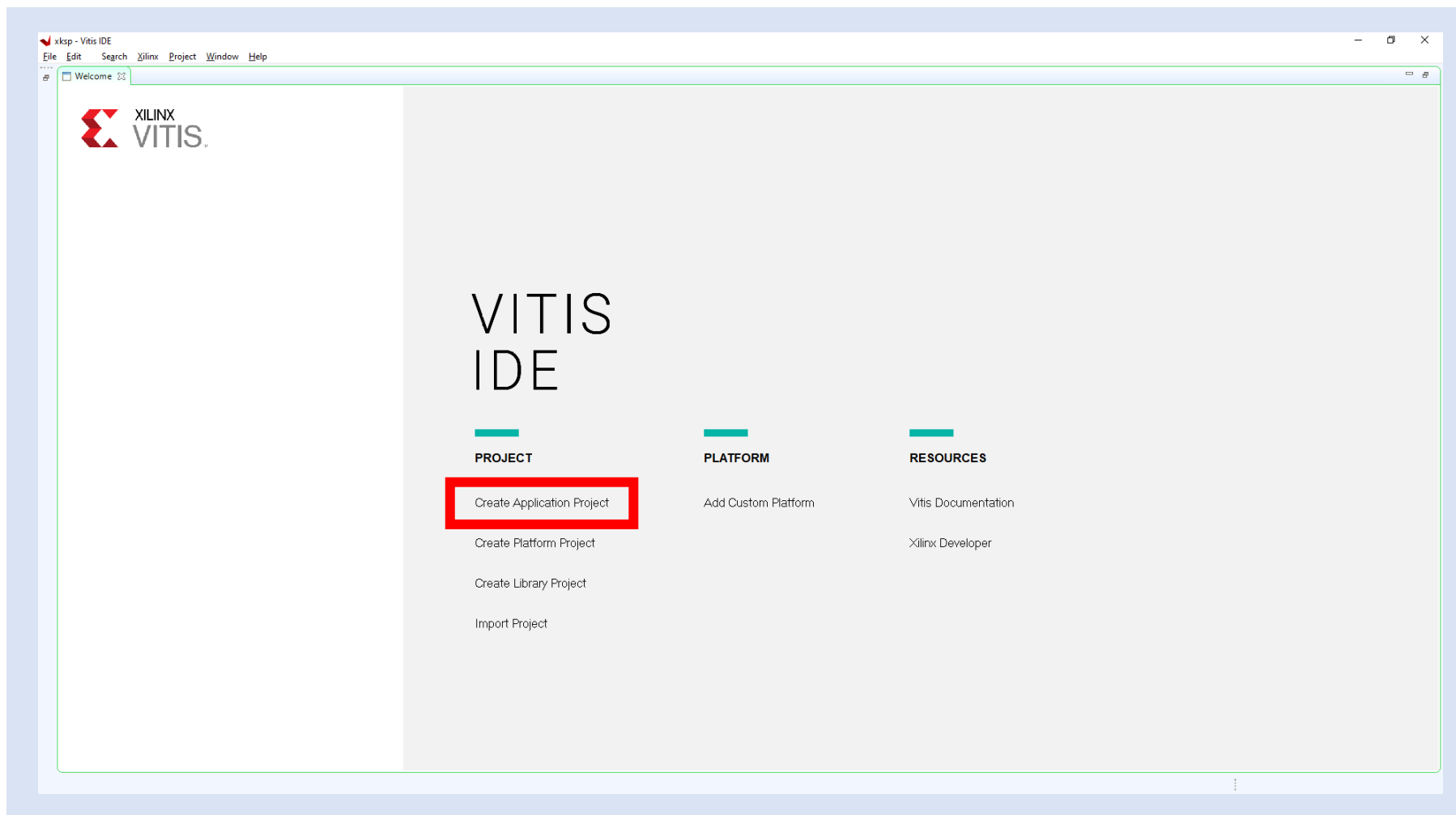
Lab 1: Understanding Vitis Project creation & Flow

Step 44 – At the dialog, create a new folder in your project directory and select launch



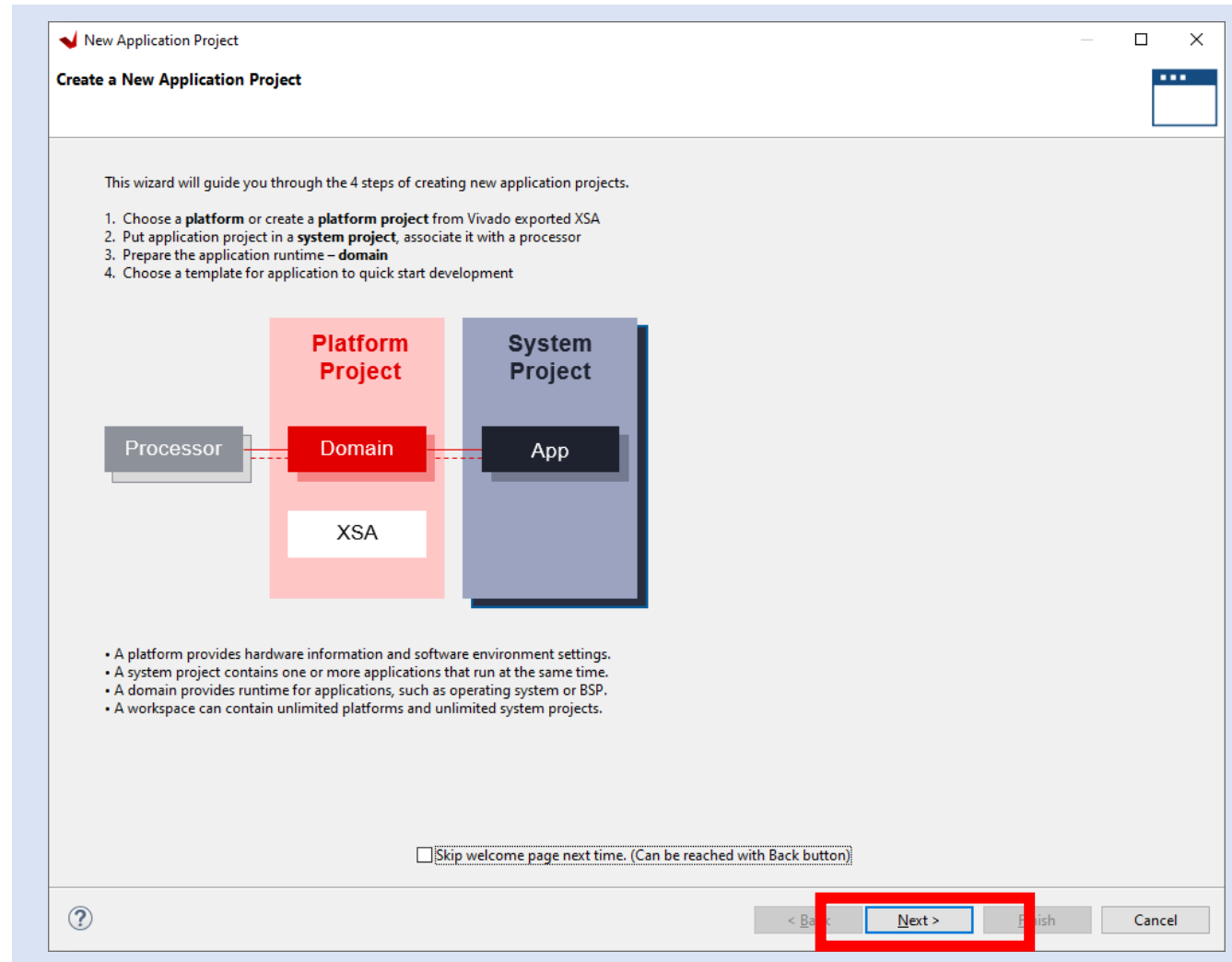
Lab 1: Understanding Vitis Project creation & Flow

Step 45 – Click on Create Application Project



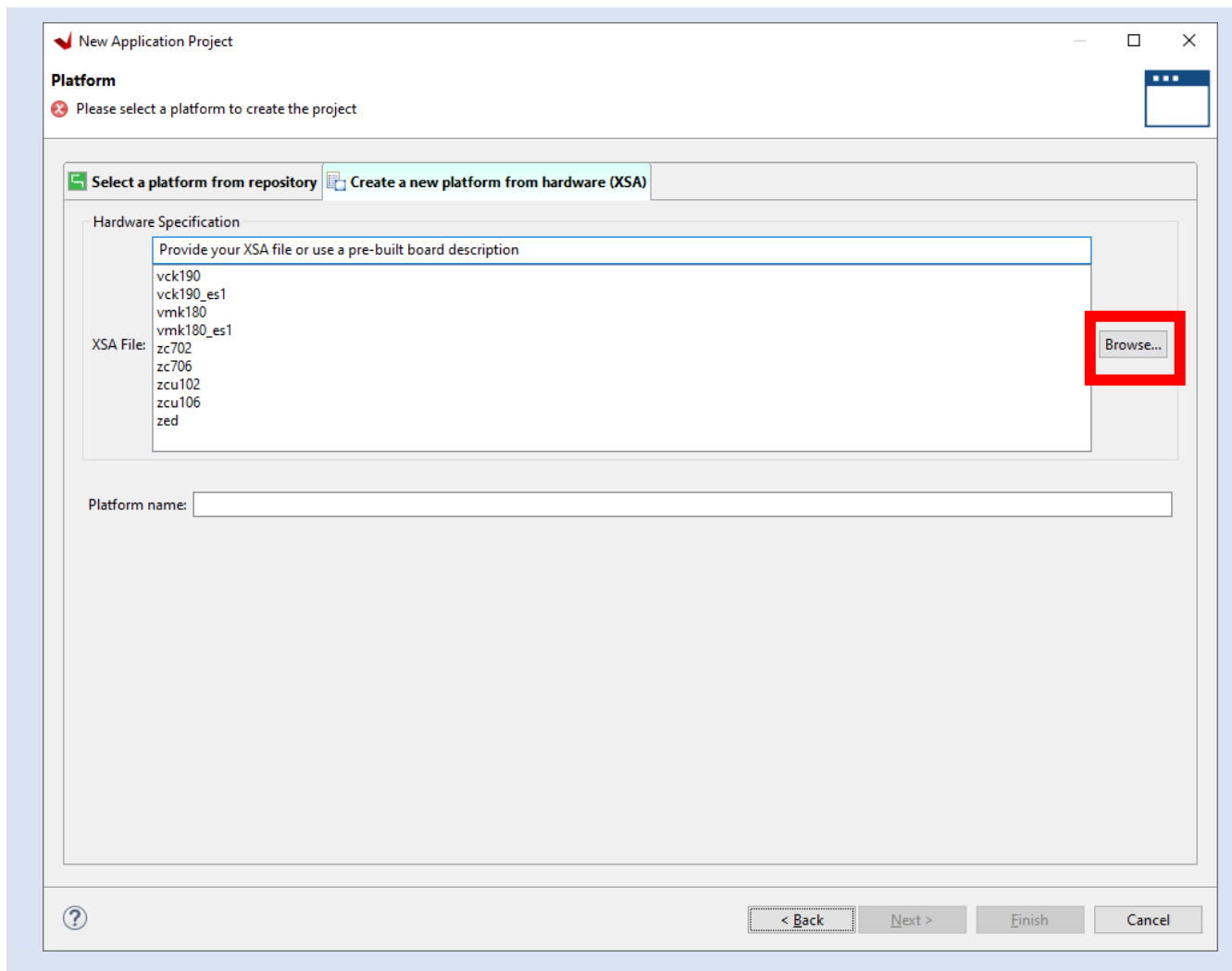
Lab 1: Understanding Vitis Project creation & Flow

Step 46 – Click on next



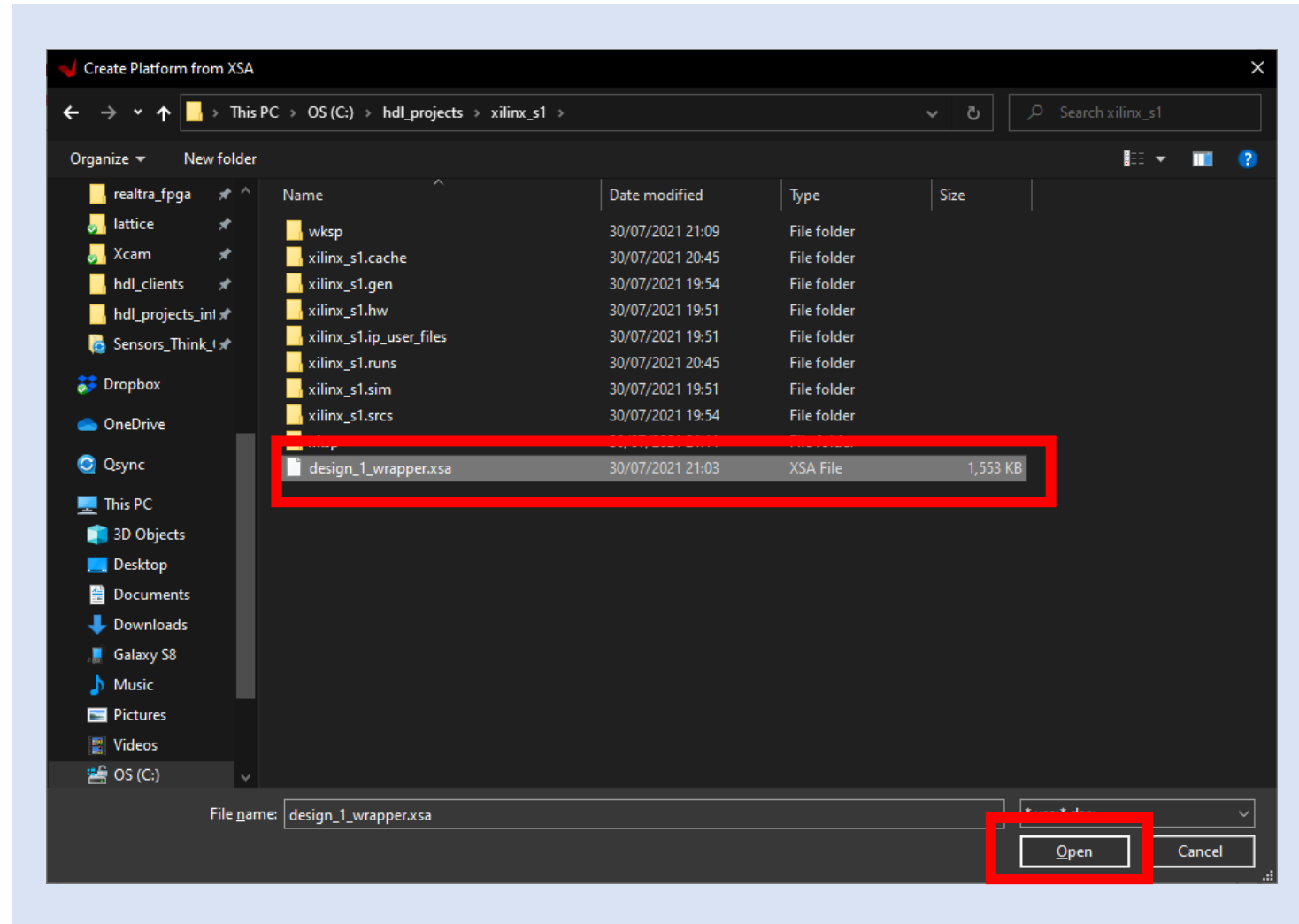
Lab 1: Understanding Vitis Project creation & Flow

Step 47 – Click on Create a New Platform from Hardware (XSA) and select browse



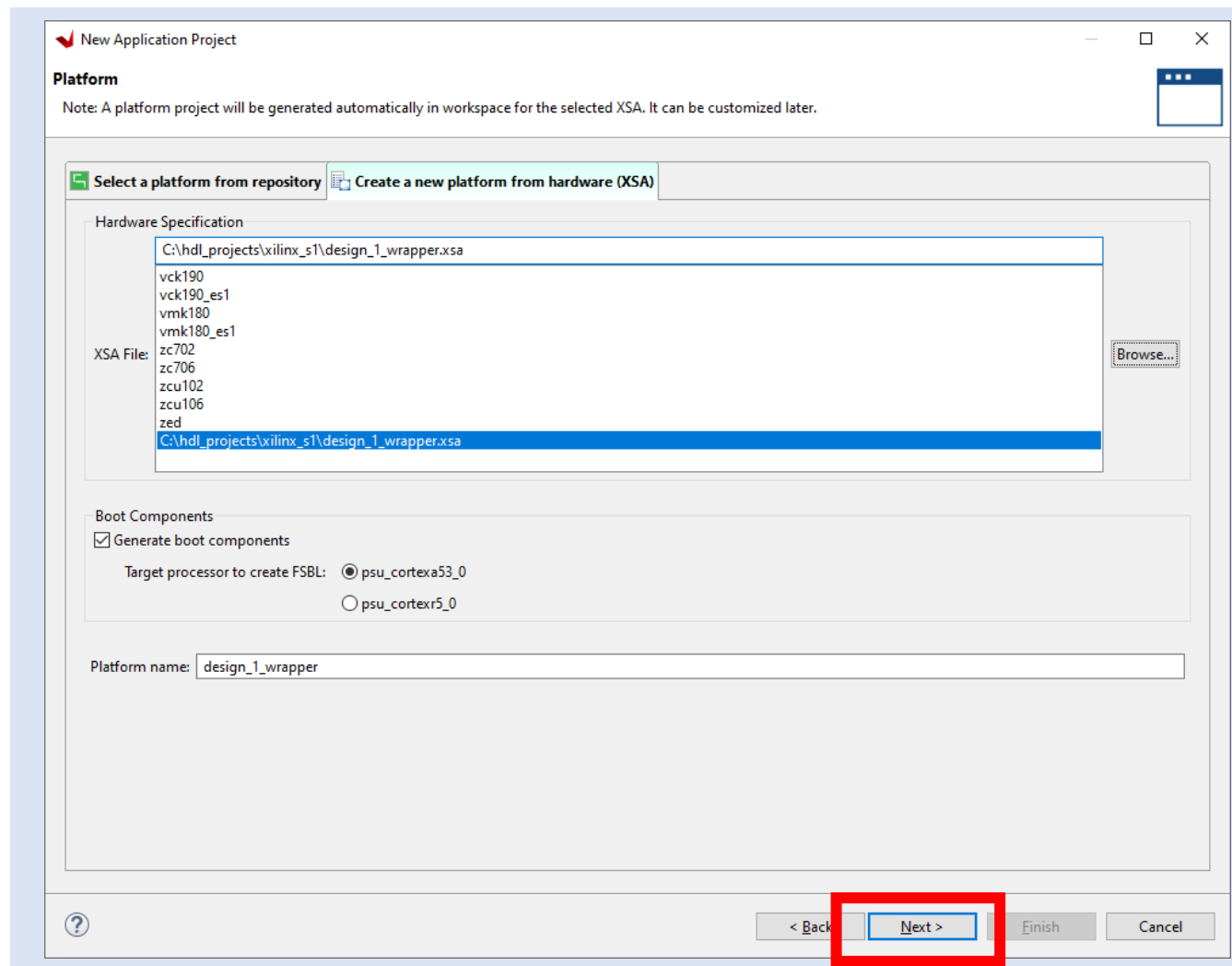
Lab 1: Understanding Vitis Project creation & Flow

Step 48 – Browse to the project directory, select the exported XSA and Click Open



Lab 1: Understanding Vitis Project creation & Flow

Step 49 – Click on OK



Lab 1: Understanding Vitis Project creation & Flow

Step 50 – Enter a project name and select Next

New Application Project

Application Project Details

Specify the application project name and its system project properties

Application project name:

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:

Target processor

Select target processor for the Application project.

Processor	Associated applications
ps7_cortexa9_0	ArtyZ7
ps7_cortexa9_1	
ps7_cortexa9 SMP	

Show all processors in the hardware specification

< Back **Next >** Finish Cancel

Lab 1: Understanding Vitis Project creation & Flow

Step 51 – 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: standalone_ps7_cortexa9_0

Display Name: standalone_ps7_cortexa9_0

Operating System: standalone

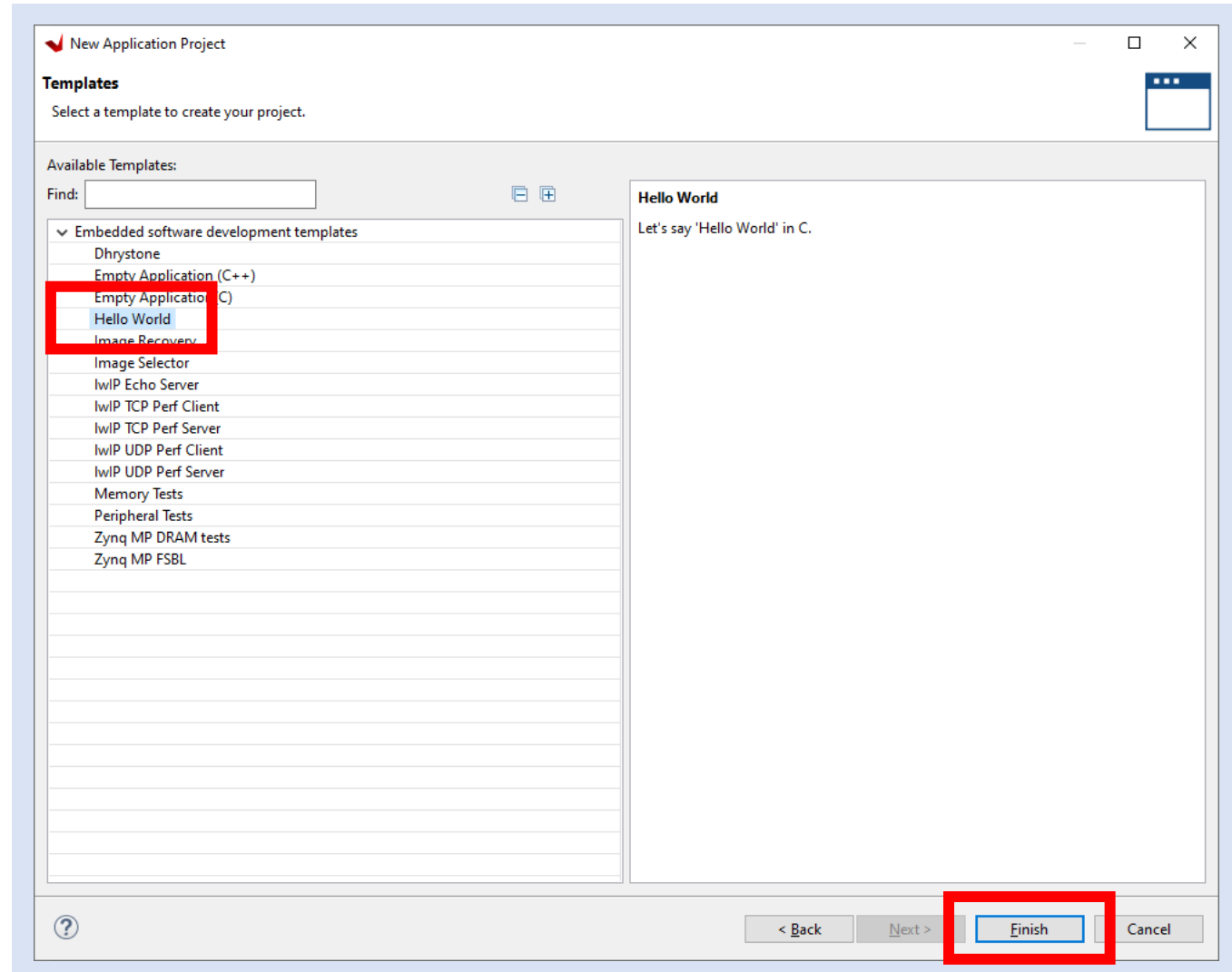
Processor: ps7_cortexa9_0

Architecture: 32-bit

< Back Next > Finish Cancel

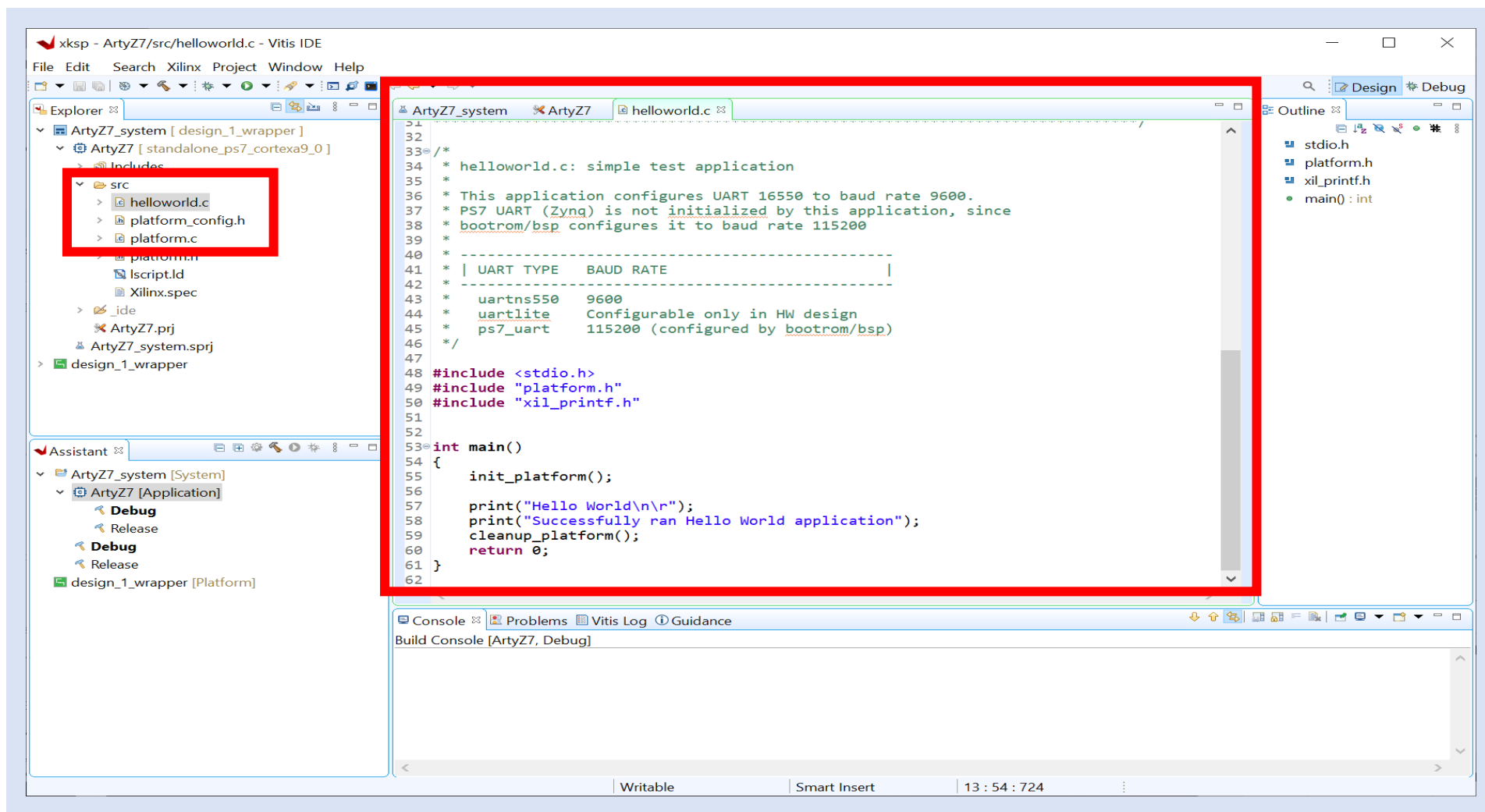
Lab 1: Understanding Vitis Project creation & Flow

Step 52 - Select Hello World and click finish



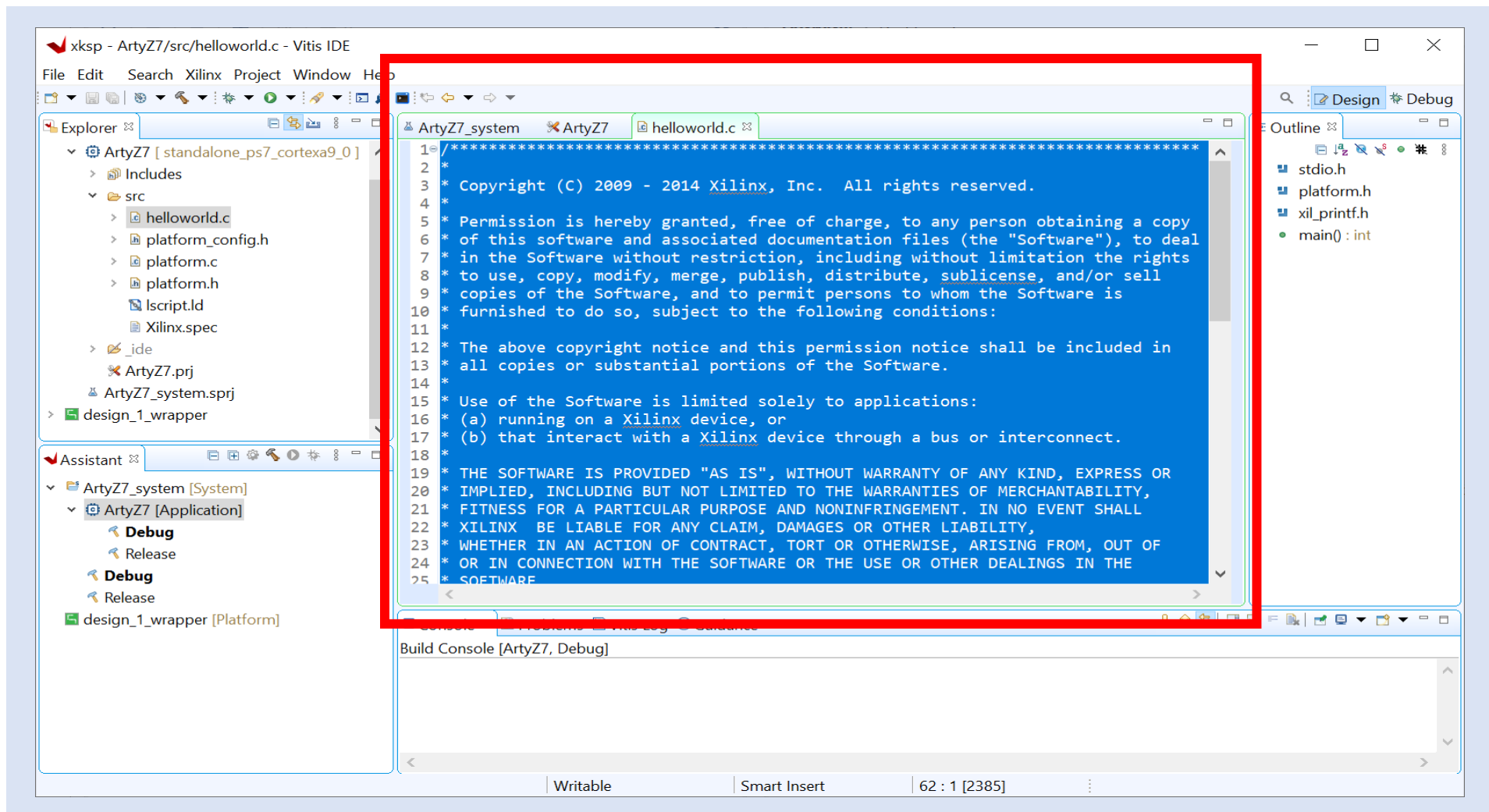
Lab 1: Understanding Vitis Project creation & Flow

Step 53 - From the application / src folder double click and open the helloworld.c



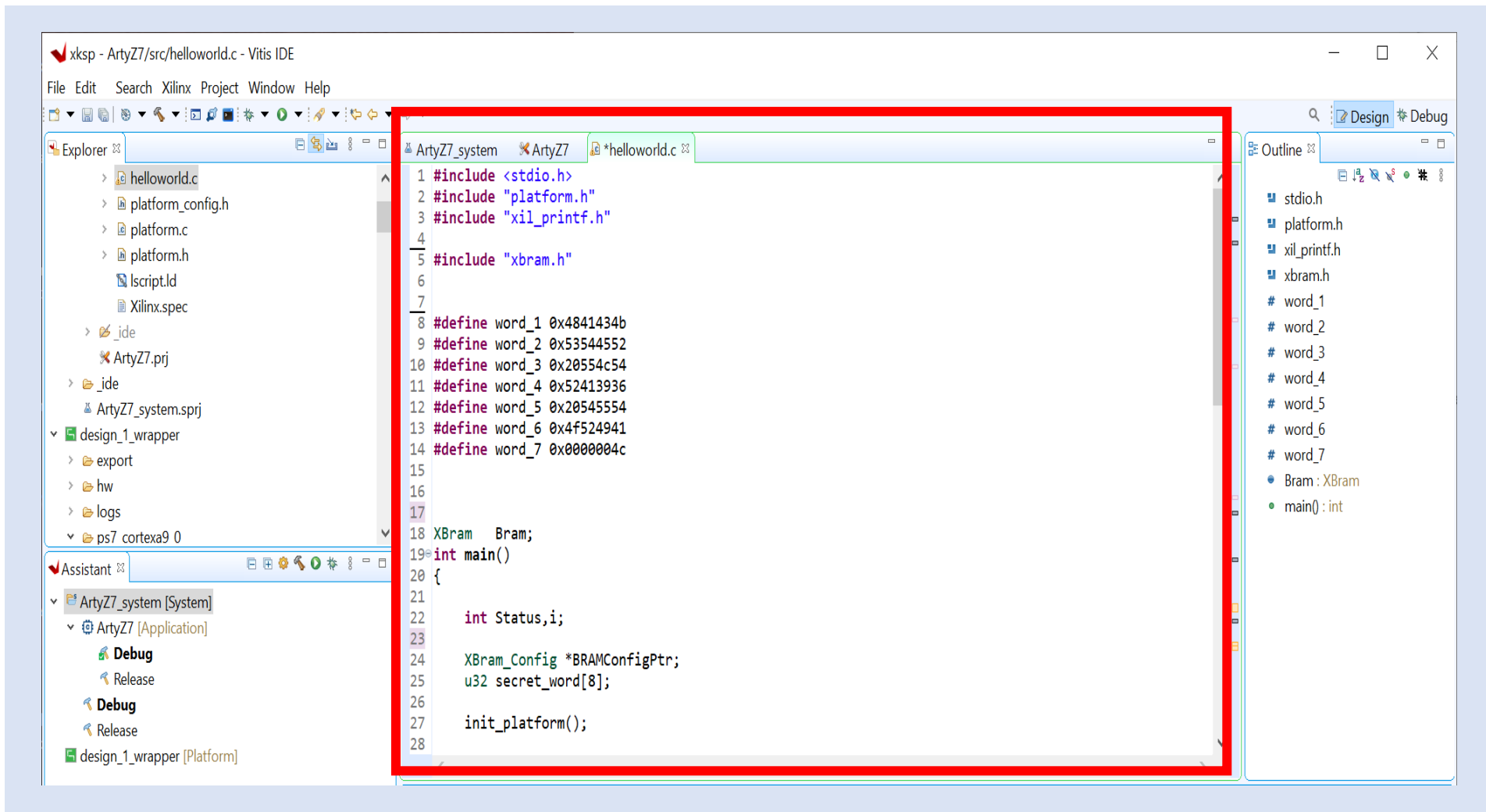
Lab 1: Understanding Vitis Project creation & Flow

Step 54 - Click CNTRL-A to select all the code in the file and delete it



Lab 1: Understanding Vitis Project creation & Flow

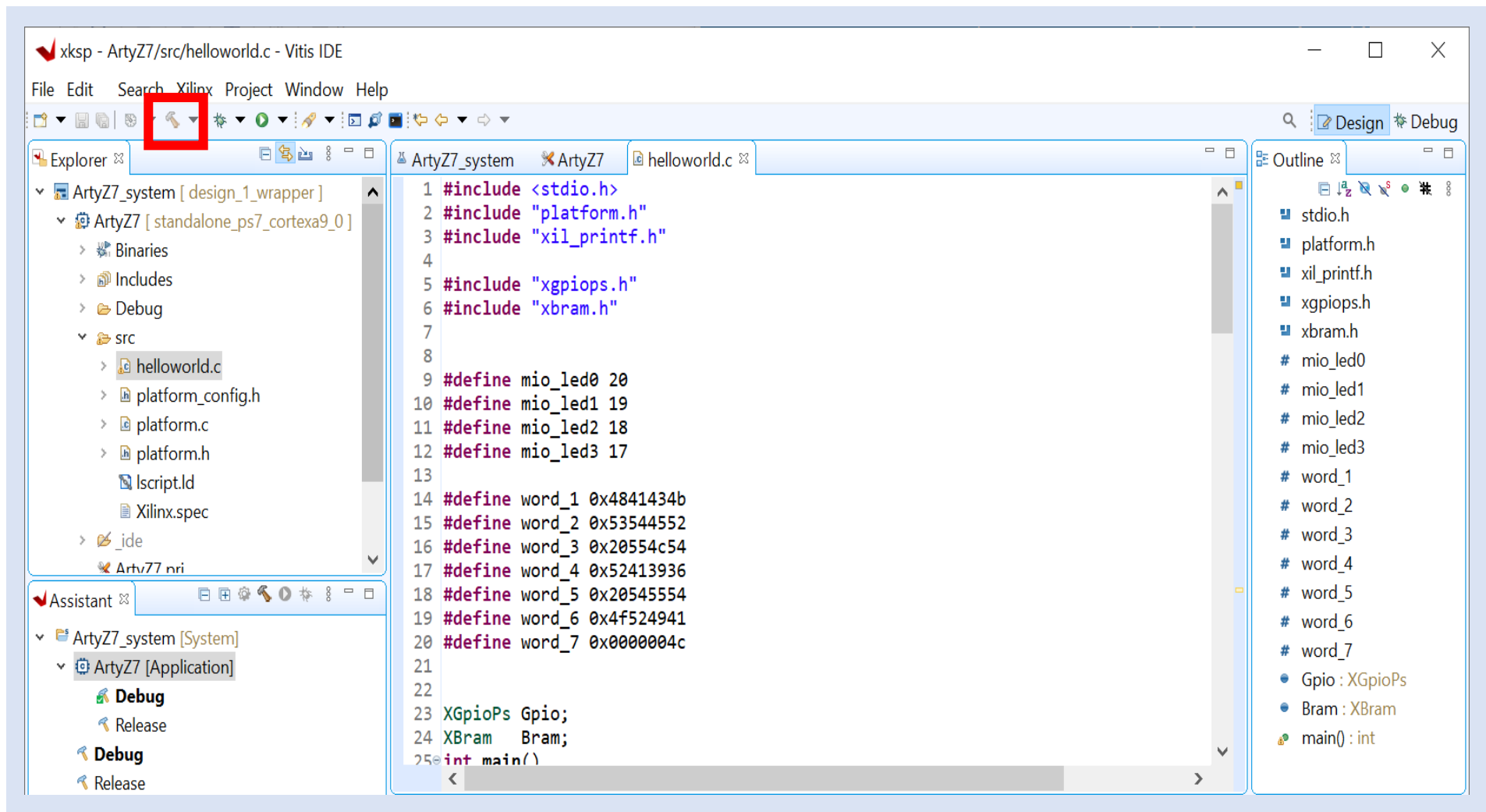
Step 55 - Copy and paste in the code from the Github repo session one lab



```
1 #include <stdio.h>
2 #include "platform.h"
3 #include "xil_printf.h"
4
5 #include "xbram.h"
6
7
8 #define word_1 0x4841434b
9 #define word_2 0x53544552
10 #define word_3 0x20554c54
11 #define word_4 0x52413936
12 #define word_5 0x20545554
13 #define word_6 0x4f524941
14 #define word_7 0x0000004c
15
16
17
18 XBram Bram;
19 int main()
20 {
21
22     int Status,i;
23
24     XBram_Config *BRAMConfigPtr;
25     u32 secret_word[8];
26
27     init_platform();
28
```

Lab 1: Understanding Vitis Project creation & Flow

Step 56 - Click on the Hammer to Build the project



Lab 1: Understanding Vitis Project creation & Flow

Step 57 - It will take a few minutes to compile. Successful completion will show as below

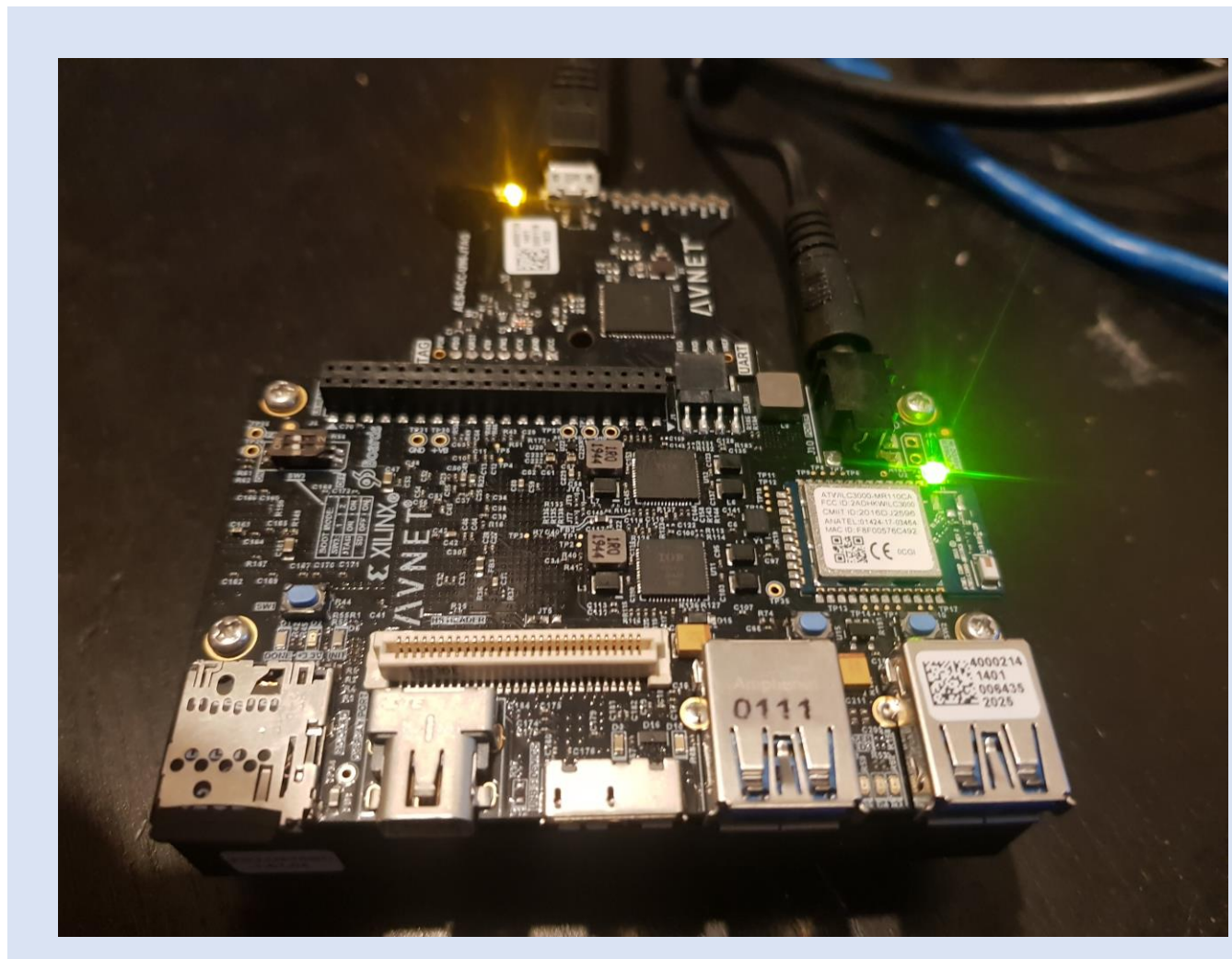
The screenshot displays the Vitis IDE interface. The main editor shows the source code for 'helloworld.c' in the 'ArtyZ7_system' project. The code includes a function to initialize BRAM, a loop to read secret words, and a while loop that checks if the secret words match. The console at the bottom shows the build process for 'ArtyZ7.elf' and reports 'Build Finished (took 1s.265ms)'. A red box highlights the console output.

```
30  
31 Status = XBram_CfgInitialize(&Bram, BRAMConfigPtr, BRAMConfigPtr->CtrlBaseAddress);  
32 if (Status != XST_SUCCESS) {  
33     return XST_FAILURE;  
34 }  
35  
36 for(i=0;i<8;i++){  
37     secret_word[i] = XBram_ReadReg(XPAR_BRAM_0_BASEADDR, i*4);  
38 }  
39  
40  
41  
42 if (Status != XST_SUCCESS) {  
43     return XST_FAILURE;  
44 }  
45  
46  
47  
48 while(1){  
49  
50     if( (secret_word[1]==word_1) && (secret_word[2]==word_2) && (secret_word[3]==word_3)  
51         && (secret_word[4]==word_4) && (secret_word[5]==word_5) && (secret_word[6]==word_6)  
52         && (secret_word[7]==word_7))  
53     {  
54         printf("Secret Word Detected Correctly \n\n");  
55         usleep(1000000);  
56     }  
57     else  
58     {  
59         printf("Secret Word Incorrect \n\n");  
60         usleep(1000000);  
61     }  
62  
63 }  
64  
65  
66 cleanup_platform();  
67 return 0;  
68  
69 }
```

```
Build Console [ArtyZ7, Debug]  
'Building target: ArtyZ7.elf'  
'Invoking: ARM v7 gcc linker'  
arm-none-eabi-gcc -mcpu=cortex-a9 -mfpv3 -mfloat-abi=hard -Wl,-build-id=none -specs=Xilinx.spec -Wl,-T -Wl,..../src/lscrip.td -LC:/hdl_projects/xilinx_s1/xksp/design_1_wrapper/export/de  
'Finished building target: ArtyZ7.elf'  
'  
'Invoking: ARM v7 Print Size'  
arm-none-eabi-size ArtyZ7.elf |tee "ArtyZ7.elf.size"  
text data bss dec hex filename  
51343 2620 22720 76683 12bb ArtyZ7.elf  
'Finished building: ArtyZ7.elf.size'  
'  
09:52:12 Build Finished (took 1s.265ms)
```

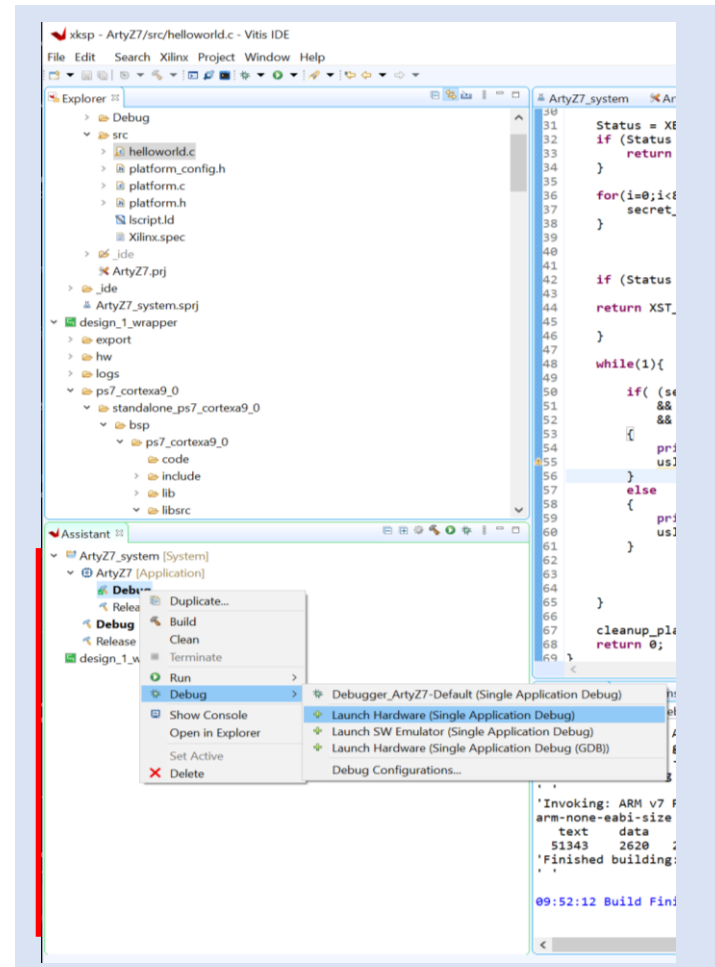

Lab 1: Understanding Vitis Project creation & Flow

Step 58 - Connect the USB to the ArtyZ7, Check the boot Mode is JTAG. Power on the board.



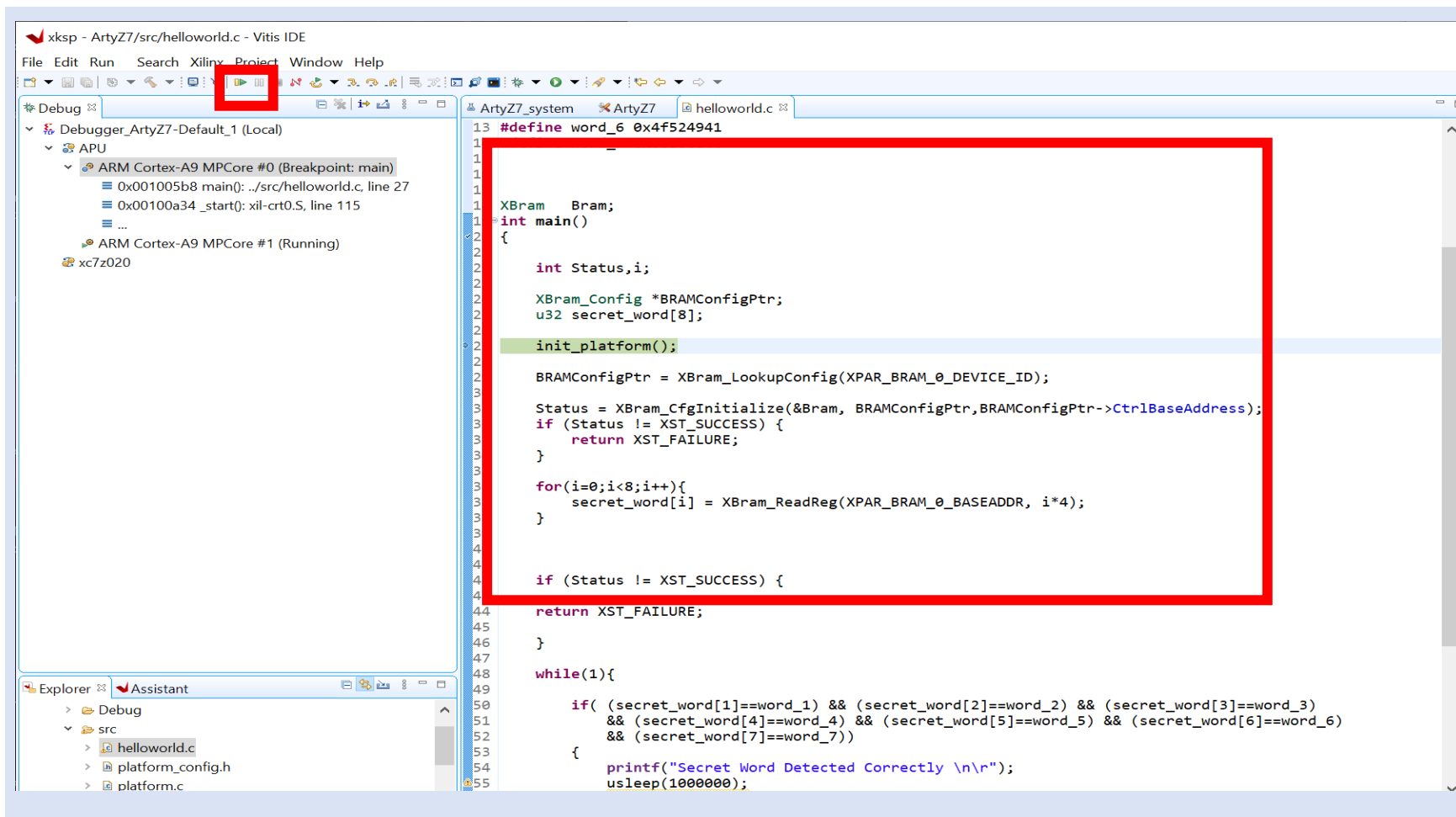
Lab 1: Understanding Vitis Project creation & Flow

Step 59 - From the assistant view, right click on debug and select debug-> launch on Hardware. This will configure the ArtyZ7, download the PL and the application.



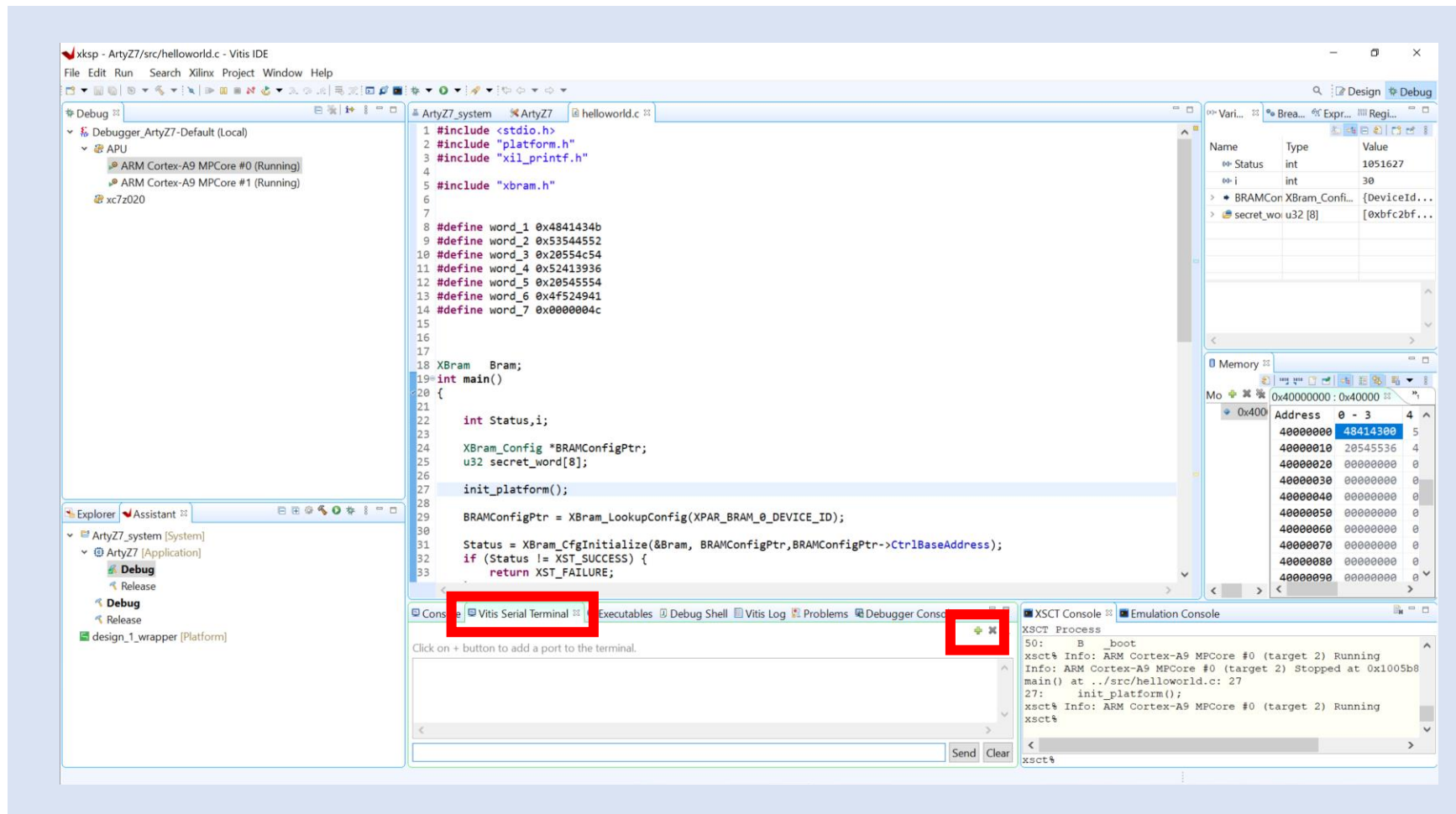
Lab 1: Understanding Vitis Project creation & Flow

Step 60 - The application will pause at the first instruction on the ArtyZ7. Click the Run Arrow



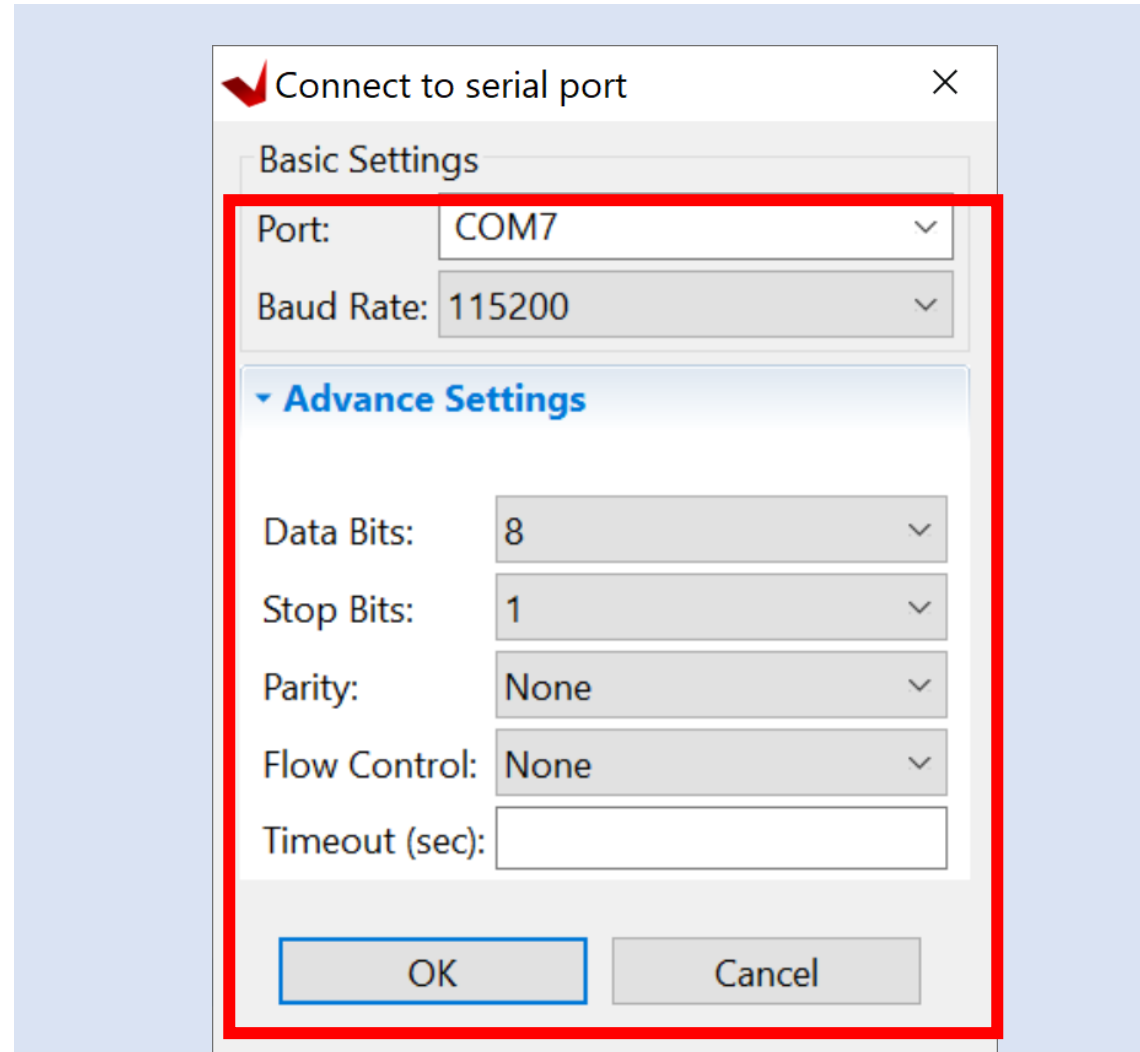
Lab 1: Understanding Vitis Project creation & Flow

Step 61 – Project will run, Click on Vitis Serial Terminal and Click +



Lab 1: Understanding Vitis Project creation & Flow

Step 62 – Add the following settings to the pop up and click ok



Lab 1: Understanding Vitis Project creation & Flow

Step 63 – Check the terminal window, secret word should be detected correctly

