



A Practical Guide to Getting Started with Xilinx SDSoC

Lab 1

Tools:	2017.4
Training Version:	v2
Date:	19 April 2018

© 2018 Avnet. All rights reserved. All trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Avnet is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Avnet makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Avnet expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Introduction

Using the instructions contained herein, you shall learn how to install custom platforms as well as what an output of SDSoC is using the provided example project. Using this base knowledge, you will be able to better understand the needs of SDSoC while leveraging this platform for your own projects. There are some places where we will accept some acceleration (use of Board Presets), however there is no reason that you would be required to do such. For instance, if you were generating the SDSoC platform for a custom board.

MiniZed Overview

The MiniZed™ Starter Kit from Avnet Electronics Marketing provides engineers with a complete system for prototyping and evaluating systems based on the Xilinx Zynq® 7Z000S device family.

MiniZed is a Zynq 7Z007S single-core development board. With the advent of the latest cost-optimized portfolio from Xilinx, this board targets entry-level Zynq developers with a low-cost prototyping platform. Please contact your local Avnet FAE for further details.

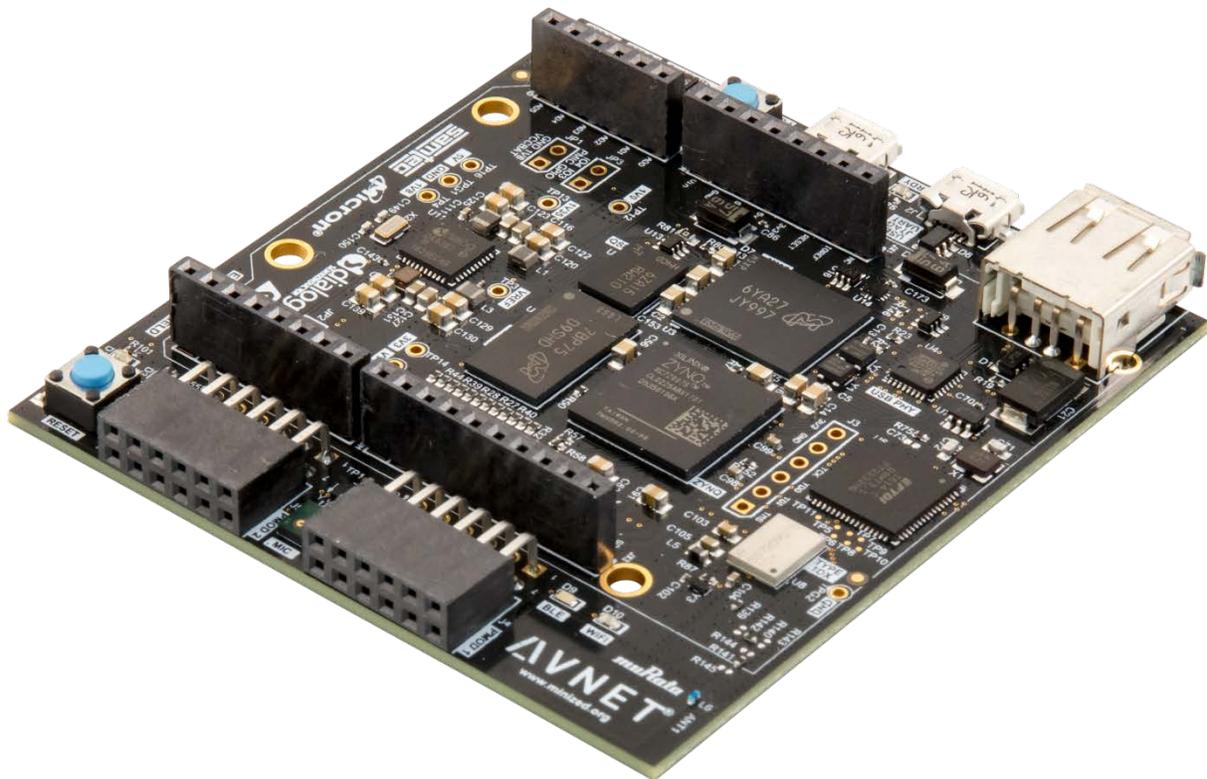


Figure 1 - MiniZed

Lab 1 Design Objectives

Lab 1 offers system developers an example of how to:

- Work through and become familiar with the SDSoC 2017.4 tool flow, from a designer's perspective
- Demonstrate a Matrix Multiple Example output on a MiniZed using a pre-built SDSoC platform (you will learn in Lab 2 how to create that platform)
- Learn how to install custom SDSoC Platforms

In order to install a custom platform, you will first need to generate the platform. In this case, Xilinx and Avnet has multiple resources available. The main reference you should use will be the Xilinx User Guide 1146. The SDSoC Environment Platform Development Guide v2017.1 document has all the proper references to the details one would need to create their own platform. Avnet and Xilinx also both offer trainings. Avnet has a training guide which steps the user through creation of a custom platform for the MiniZed development board. This SpeedWay is called "A Practical Guide to Getting Started with Xilinx SDSoC".

Example Design Requirements

Software

The software used to test this reference design is:

- Xilinx SDx / SDSoC 2017.4 (SDSoC License Required)
- Platform archive
- MiniZed Board Definition for Vivado

Hardware

The hardware setup used to test this reference design includes:

- Lenovo ThinkPad T420 Laptop
 - Intel® Core i5-2540M CPU - 2.60 GHz
 - 4GB DDR3 Memory
 - SD card slot on PC or external USB-based SD card reader
- Avnet MiniZed (AES-MINIZED-7Z007-G)
- 1 - USB cable (Type A to Micro-USB Type B)

Experiment Set Up

You must have installed the Xilinx tools and properly licensed them. The MiniZed Board Definition should be installed into both your standard Vivado and your SDx Vivado installations. Instructions for installing board definitions are located on the MiniZed.org website.

You will also need an unzip tool, such as 7-zip.

Experiment 1: Install the Pre-Built MiniZed (mz_avnet) SDx Hardware Platform

SDSoC comes pre-installed with many platforms that allow you to immediately use many Xilinx boards. MiniZed is not one of them. For this experiment, you will use a pre-built hardware platform called **mz_avnet** so that you can quickly begin using SDx. You will later learn how to build that hardware platform.

Installation of an SDx hardware platform is a two-step process:

- Copy the hardware platform files to a repository. For our purposes today, we will use `C:\Avnet\platforms` as our repository location.
- Setup SDx to use a repository, pointed at the repository location

We'll start this experiment by creating a new project so that you can see the pre-installed platforms. Then you will add the `mz_avnet` platform.

1. Launch the SDx IDE by clicking **Start → Xilinx Design Tools → SDx 2017.4 → SDx IDE 2017.4**

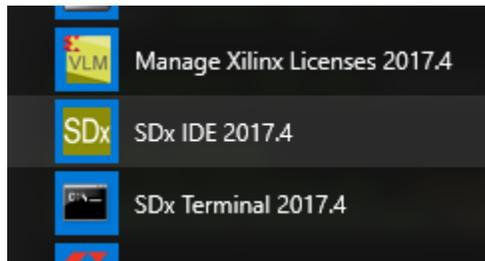
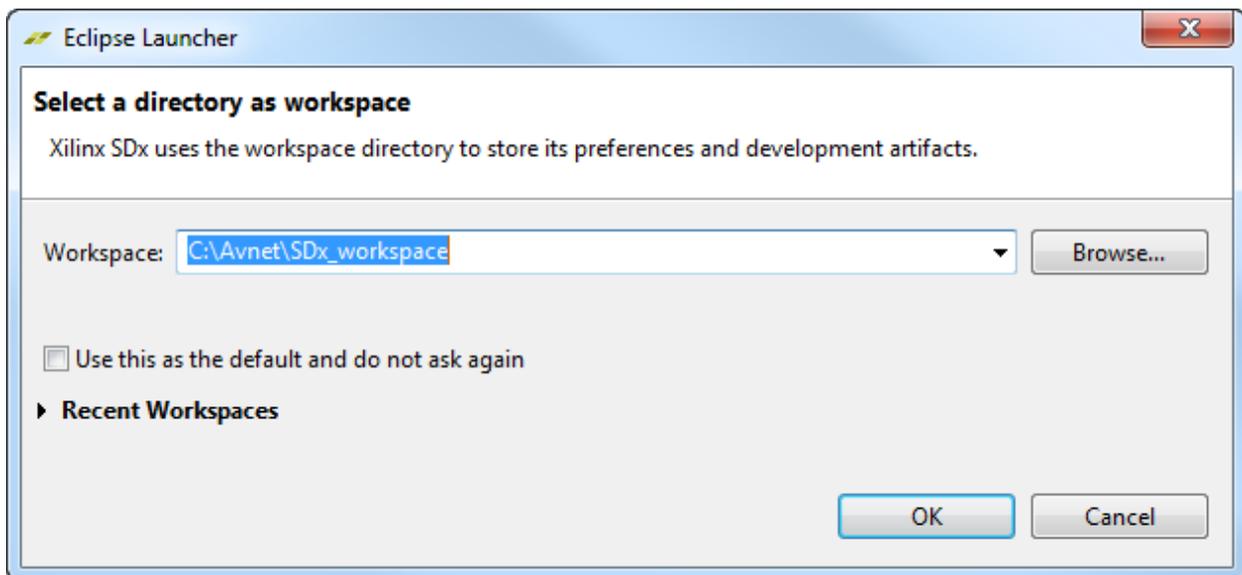


Figure 2 – Launch SDx IDE

2. SDSoc requires a workspace. Due to Windows path length constraints. It is important that this path be short. Please enter this specific path for consistency through these labs and then click **OK**.

C:\Avnet\SDx_workspace



3. If you see a security alert, select **Allow Access**.

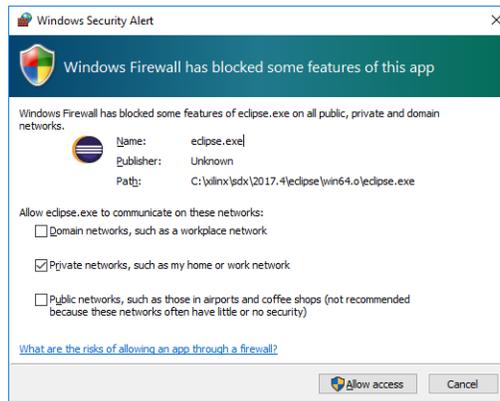


Figure 3 – Security Alert

The SDx IDE will launch and validate your license. You should see the Welcome dashboard as shown below.

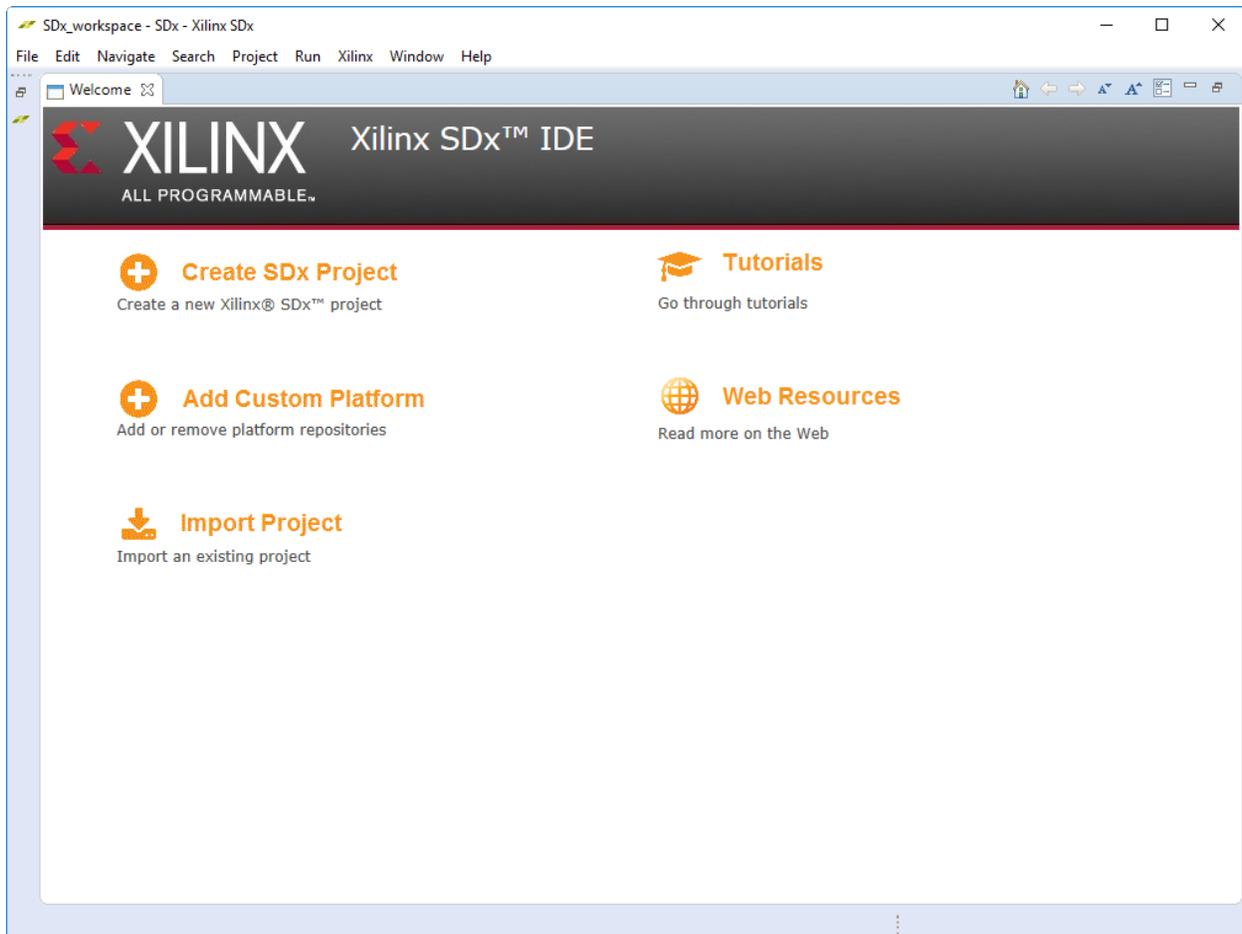


Figure 4 – SDx IDE Dashboard

4. Begin by creating a new Xilinx SDx project
 - a. Select **File** → **New** → **Xilinx SDx Project...** or click **Create SDx Project** on Welcome screen

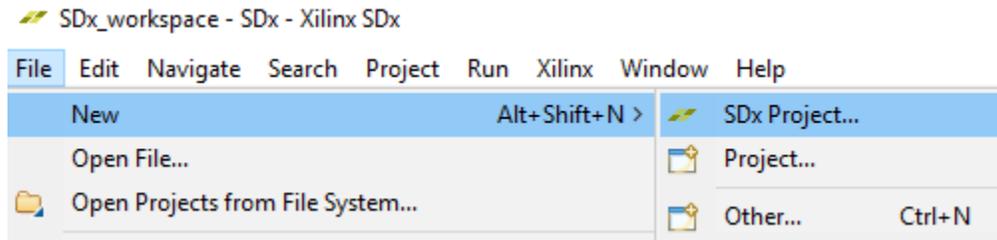


Figure 5 – Xilinx SDx Project Creation

5. For Project Type, leave the default Application Project Selection, click **NEXT>**

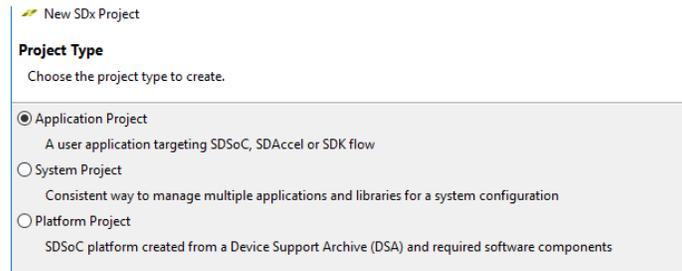


Figure 6 – Project Type Selection

6. For Project Name, enter **MiniZedMM**
 - b. NOTE: the location of the project

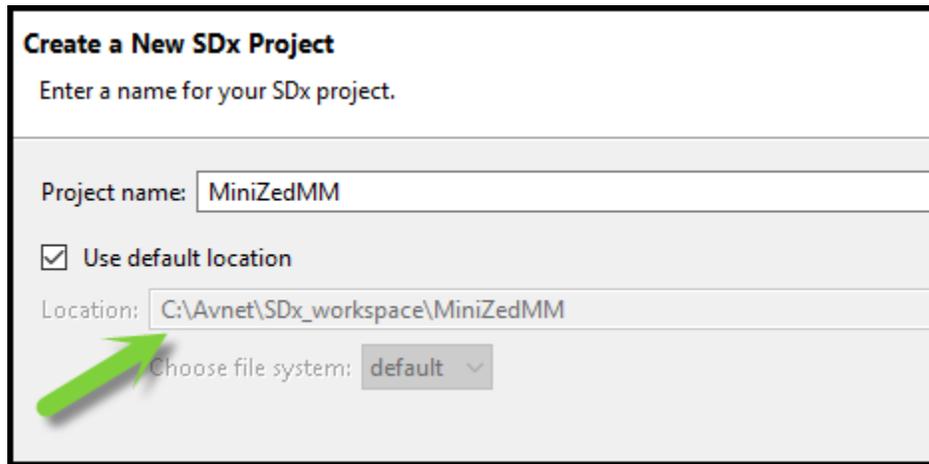


Figure 7 – Project Location

7. Click **Next >**

Here you will see the list of pre-installed Hardware Platforms. These are all Zynq-based boards. Notice that MiniZed is not in the list.

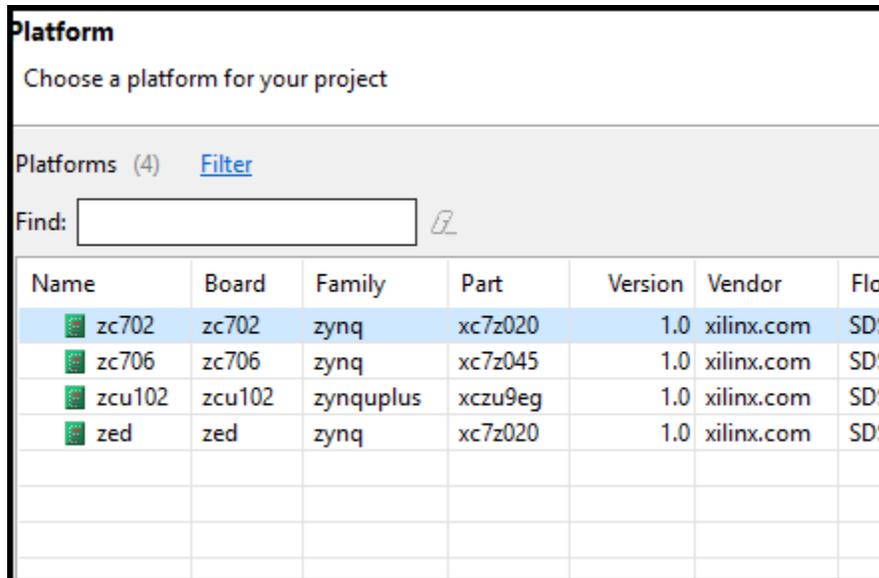


Figure 8 – Pre-installed Hardware Platforms in SDx 2017.4

8. Open Windows Explorer and create the directory `C:\Avnet\platforms`
9. Now unzip the `mz_avnet.zip` archive located at `C:\Speedway\SDSoC\2017_4\Support_documents` into the repository at `C:\Avnet\platforms`

When complete you should have a directory structure as shown below:

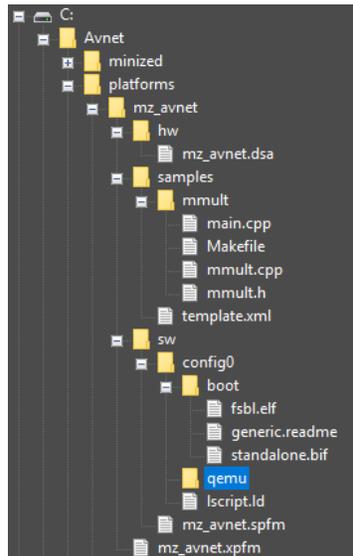


Figure 9 – SDx mz_avnet Installed to Repository

10. In the SDx New Project dialog, click on **Manage Repositories...**

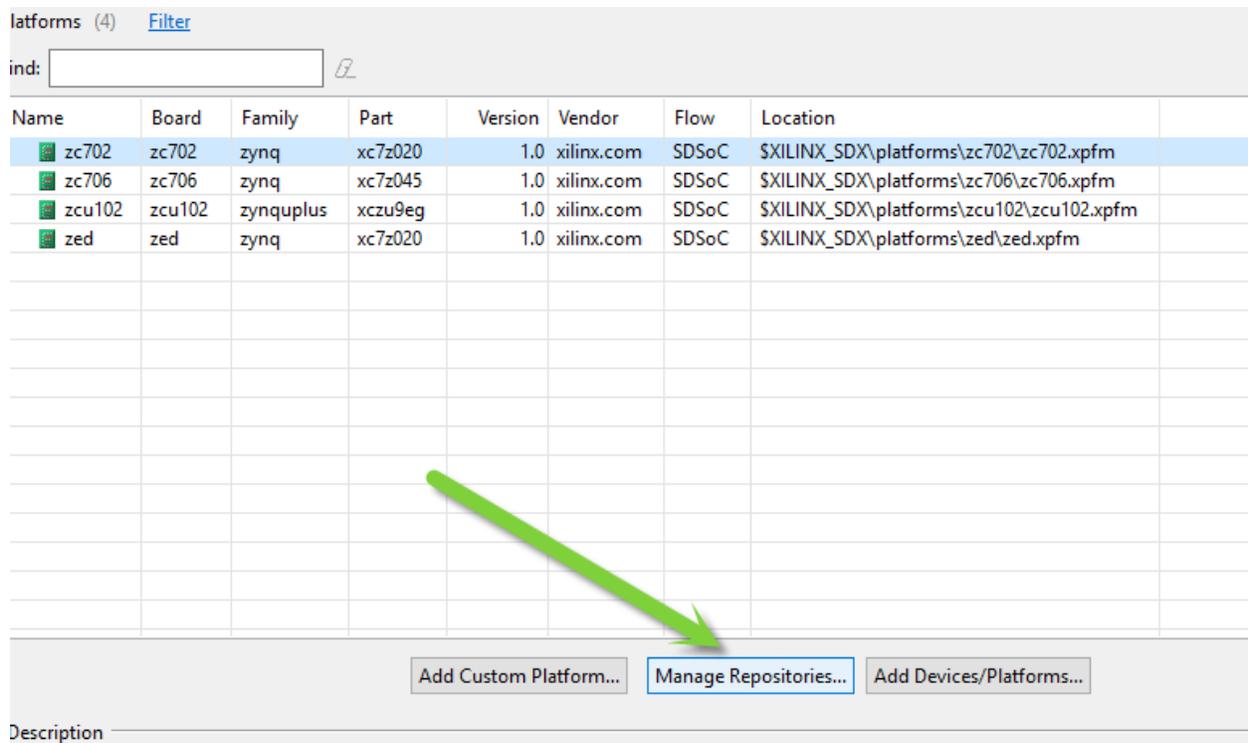


Figure 10 – Manage Repositories

11. In the next dialog, click on the green plus sign on the left pane.

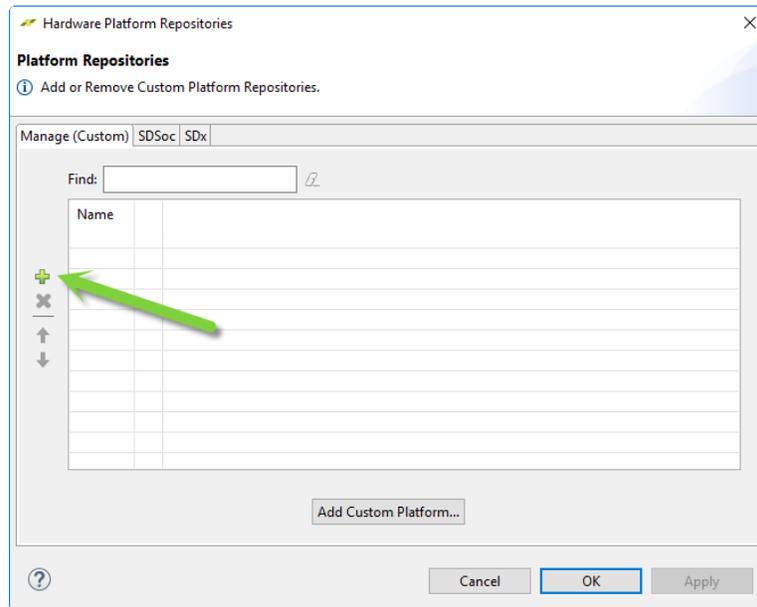


Figure 11 – Add Custom Platform Location

12. Navigate to C:\Avnet\platforms. Select **platforms** and click **OK**.

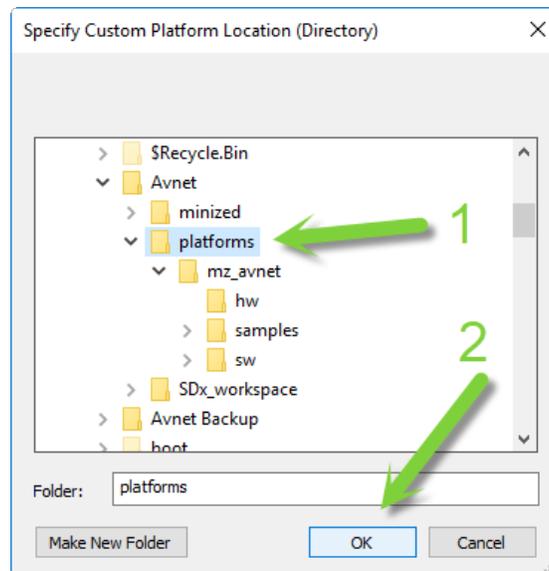


Figure 12 – Specify Custom Platform Location

13. Next Click on Apply. Click OK to dismiss the Hardware Platform Repositories Dialog.

Notice that in the **Choose Hardware Platform** is updated with the **mz_avnet (custom)** Platform as a selection, targeting Part **xc7z007s**.

Platforms (5) [Filter](#)

Find: *g*

Name	Board	Family	Part	Version	Vendor	Flow	Location
 mz_avnet [custom]	av	zynq	xc7z007s	1.0	em.avnet.com	SDSoC	C:\Avnet\platforms\
 zc702	zc702	zynq	xc7z020	1.0	xilinx.com	SDSoC	\$XILINX_SDX\platfor
 zc706	zc706	zynq	xc7z045	1.0	xilinx.com	SDSoC	\$XILINX_SDX\platfor
 zcu102	zcu102	zynqplus	xczu9eg	1.0	xilinx.com	SDSoC	\$XILINX_SDX\platfor
 zed	zed	zynq	xc7z020	1.0	xilinx.com	SDSoC	\$XILINX_SDX\platfor

Figure 13 – mz_avnet (custom) Now an Available Platform

You have now installed a custom platform.

Experiment 2: Create the mz_avnet Matrix Multiply Project

14. Now, we can choose the mz_avnet platform by selecting **mz_avnet (custom)** then click **Next >** to continue

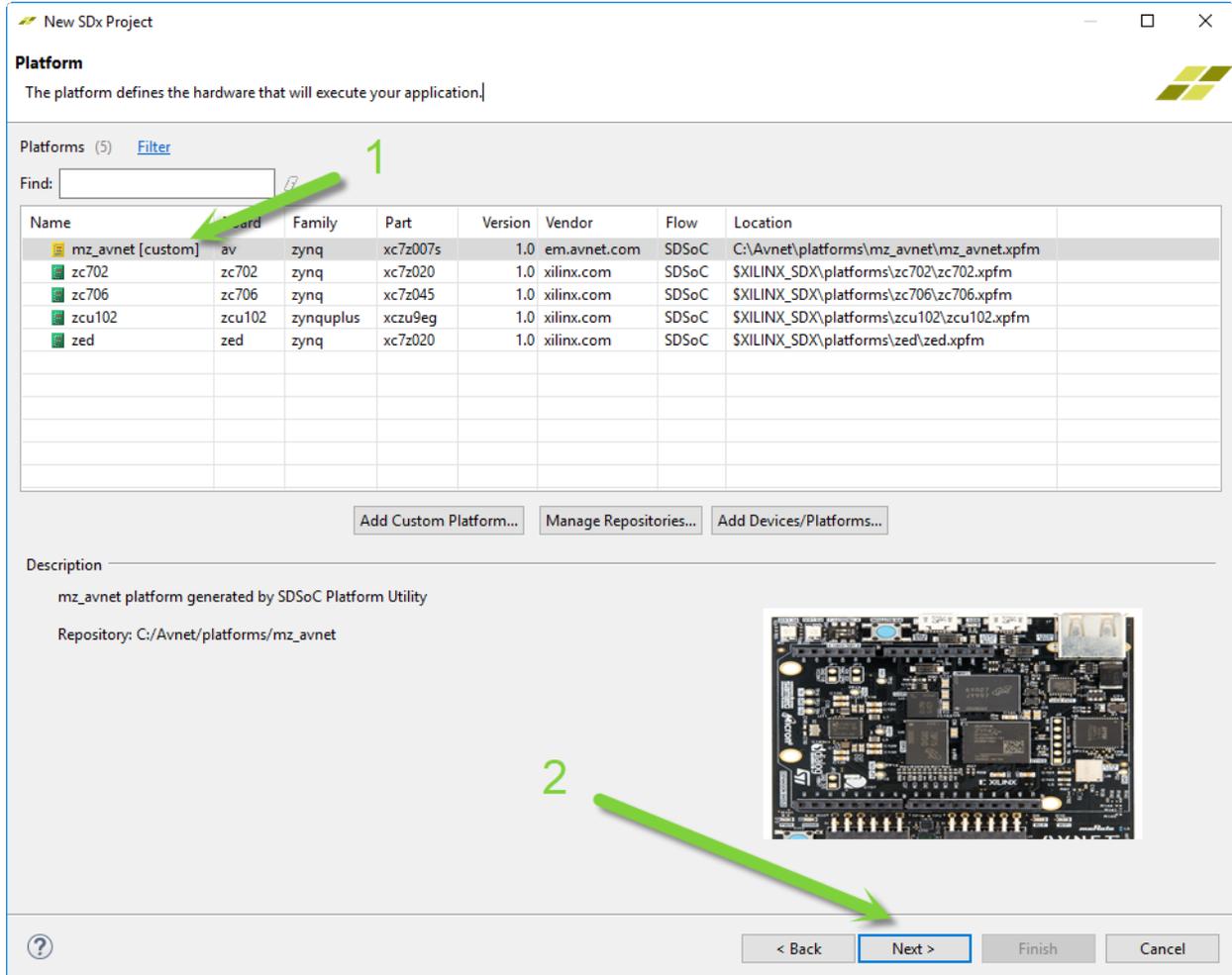
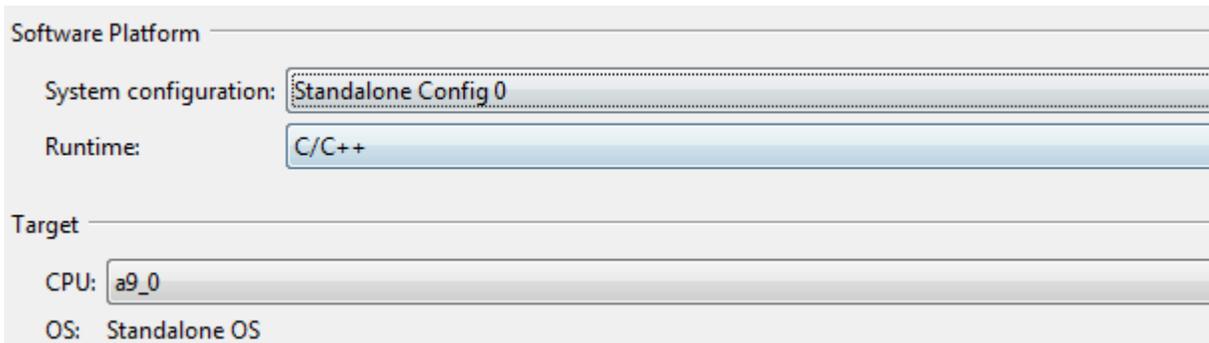


Figure 14 – Choose mz_avnet Hardware Platform

15. You can leave the defaults on the next screen where you need to choose the software platform



Software Platform

System configuration: Standalone Config 0

Runtime: C/C++

Target

CPU: a9_0

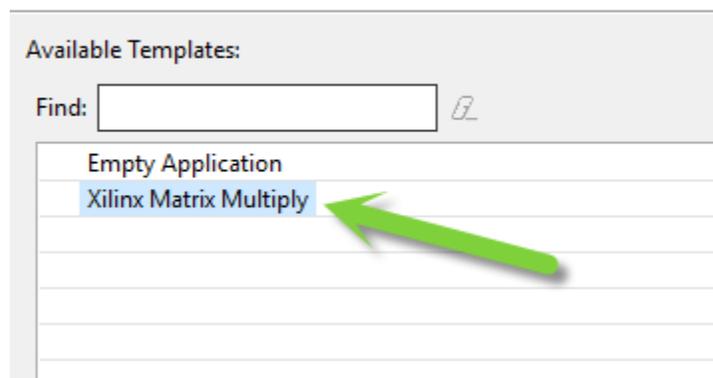
OS: Standalone OS

Figure 15 – Software Platform

16. Click **Next >** to continue.

Templates

Select a template to create your project.



Available Templates:

Find: 

Empty Application
Xilinx Matrix Multiply

Figure 16 – Example Selection

17. Select the provided Xilinx Matrix Multiply example as shown in Figure 16

18. Click **Finish** to create a new Empty application.

Now the SDx Project Explorer will have the MiniZedMM from which we can generate SDSoc projects.

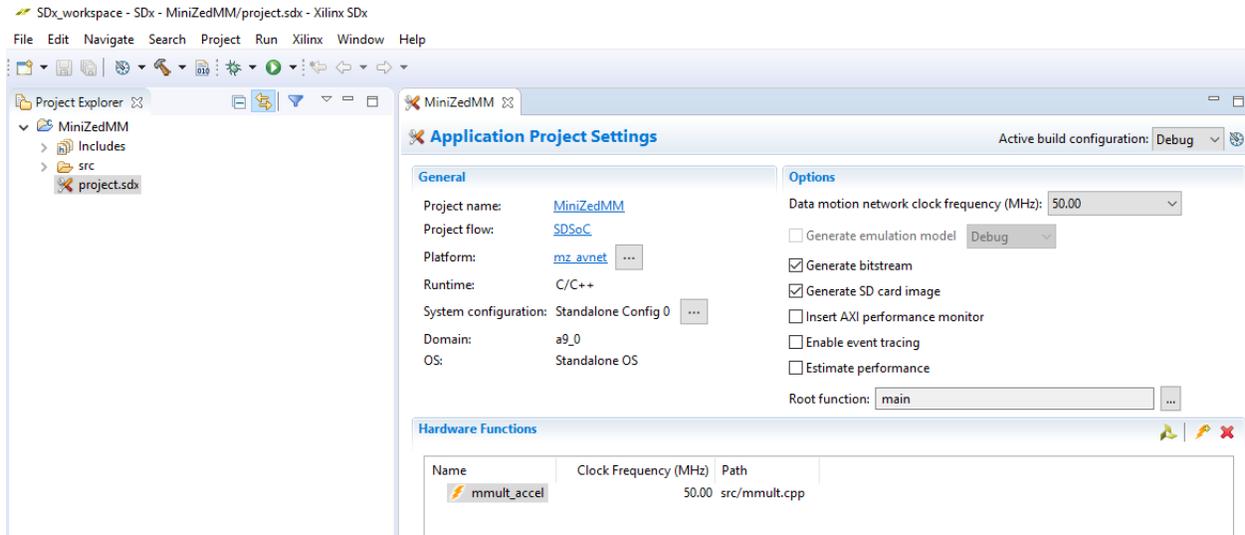


Figure 17 – MiniZedMM added to SDx Project Explorer

For the purposes of this experiment, we will use Matrix Multiply Example code.

The example code that you choose here is not really relevant as we are interested in the outputs of SDSoC as well as the platform itself. This should be seen as a black box where any accelerator (ex. video processing, LTE engine, AES engine, etc) could be inserted.

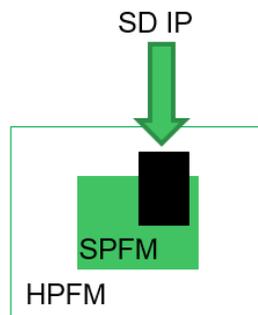


Figure 18 – Target Code is Black Box

This is one of the most powerful abilities of SDSoC! Since we are using a provided platform, simply imported the example template straight from the platform folder.

Instructions on how to create your own Sample Template is located in UG1146.

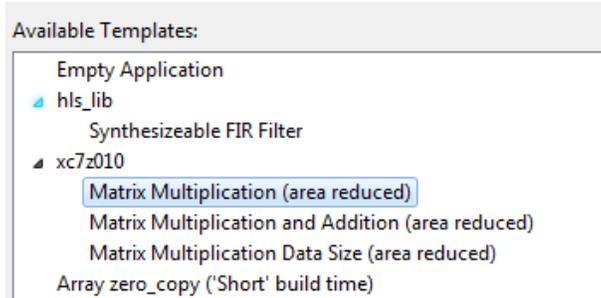


Figure 19 – Code Template for Pre-installed Platform MicroZed

If you need to import files, the process is the same as with Xilinx SDK. To learn how to import the matrix multiply files from within the SDSoc installation see the Appendix.

Experiment 3: Build the mz_avnet Matrix Multiply Project

19. Next, back in SDx Project Explorer right click the MiniZedMM project name, click on **Build Project**
20. Note that if you see the below Critical Warning, this is OK at this time. It is a warning that has been put there to indicate that the DSA has been tampered with. It has not been tampered with but at this time does not pass the SDx DSA integrity check. You will see later on that there is a validation step we perform in Vivado. This check is the most important check at this time. In future versions of SDx, this will stop the tools from building. At this time, we should take note of it and proceed.

```
INFO: [VPL 60-895] Target platform: C:/Avnet/platforms/mz_avnet/mz_avnet.xpfm  
CRITICAL WARNING: [VPL 60-724] 'C:/Avnet/platforms/mz_avnet/hw/mz_avnet.dsa' : failed DSA integrity check: digest mismatch.  
INFO: [VPL 60-423] Target device: mz_avnet
```

Figure 20 – Critical Warning

21. While this is building, notice the SDx project Settings. This is a great place to see the overview of the settings that are going into building this project

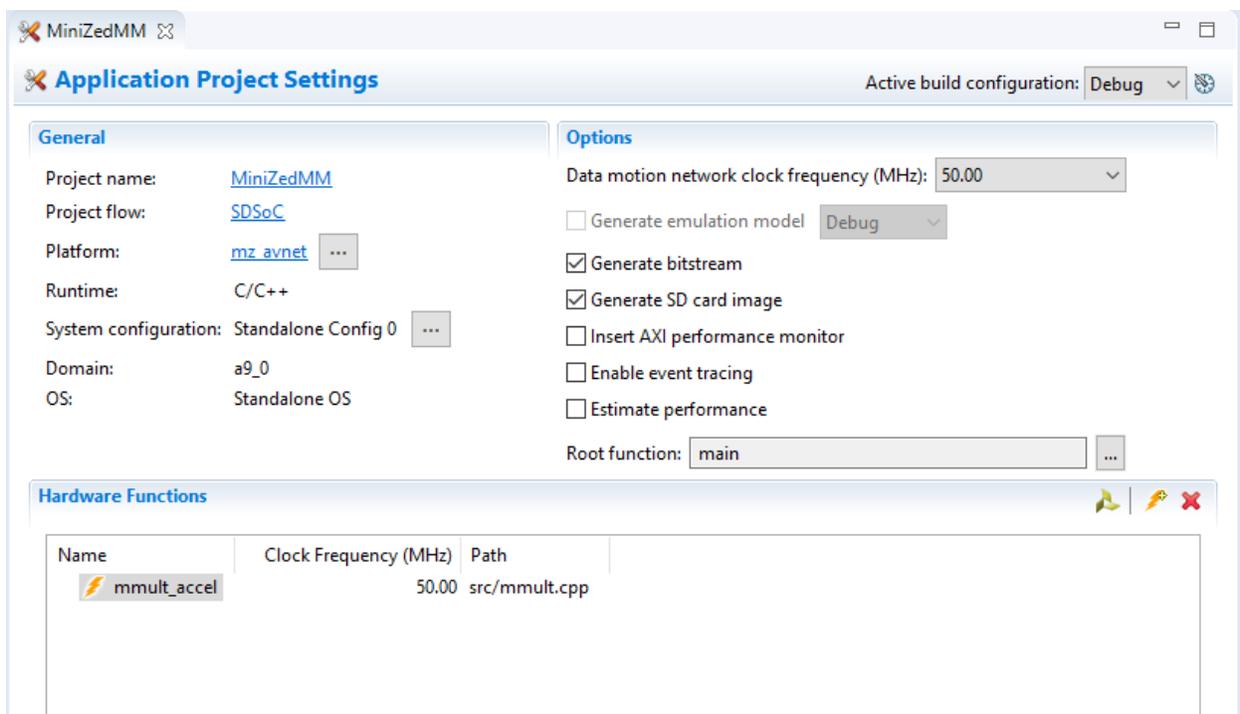


Figure 21 – Project Settings

We left the project as Debug and did not select Release. If we had selected Release, the tool would have performed additional runtime optimizations, which would have increased our build time.

The above steps can be seen in more detail in UG1028 – SDSoC Intro Tutorial v2017.4, pages 9-14, with additional information we will see useful later on in future labs on pages 14-19.

NOTE: Use DocNav or search directly from Xilinx.com as there is a good chance you can end up with an older version of the documentation.

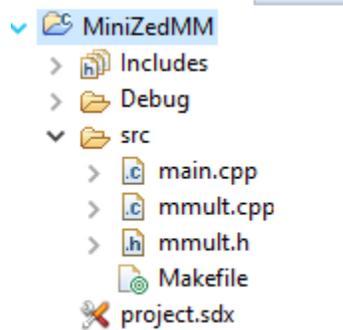


Figure 22 – MiniZedMM Sources

22. Notice that SDSoC generates a layout that looks very similar to the Xilinx SDK. This was a design decision to ensure a familiarity to the tools including the process flow.

While we wait for the build, let's explore. What is happening while this is building?

- The tool copies the Vivado project from the platform into your local build area
- The tool analyzes the provided C code, including pragmas, and builds an internal data motion graph – what connects to what, how, etc. It will make decisions at this stage based on your memory configuration, buffer sizes (if known), etc. to determine interfaces, data movers, etc.
- The C code moving to hardware is synthesized by HLS
- SDSoC updates the BD to incorporate the data motion infrastructure and the generated HLS IPs
- The tool updates your C code to seamlessly call the accelerator instead of the C function (you can see the results of this in the `_sds` directory in the project build area)
- Generate a bitstream
- Combine the bitstream into BOOT.BIN using the FSBL, BIF, etc. that you provide as part of the platform

23. Navigate to C:\Avnet\SDx_workspace\MiniZedMM\src. Notice the 3 source files there.

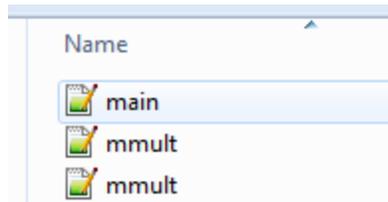


Figure 23 – MiniZedMM Source files

More notes regarding the files and the file structure

- Main.cpp runs the functions
- Mmult.cpp has pragmas listed
- Mmult.h has pragmas listed

24. While this is building, open the Xilinx Software Command Line Tool

- a. We will need this to jtag the BOOT.BIN file onto our MiniZed.

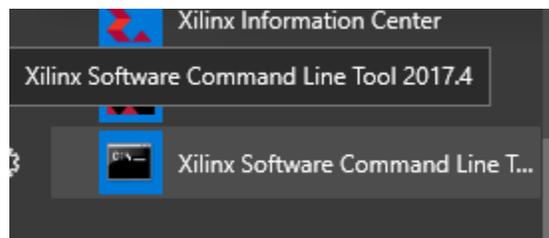


Figure 24 – Open Xilinx Software Command Line Tool

Experiment 4: Set Up Your MiniZed

While the project is building, you can also set up your MiniZed.

25. First configure the boot jumper. You can select between FLASH and JTAG booting. We want to ensure switch 1 is set towards the F or PS_Button.
 - a. Note: From the Factory the switch's protective film should be removed and already set to F. If it is not, the switch will look similar to Figure 25.

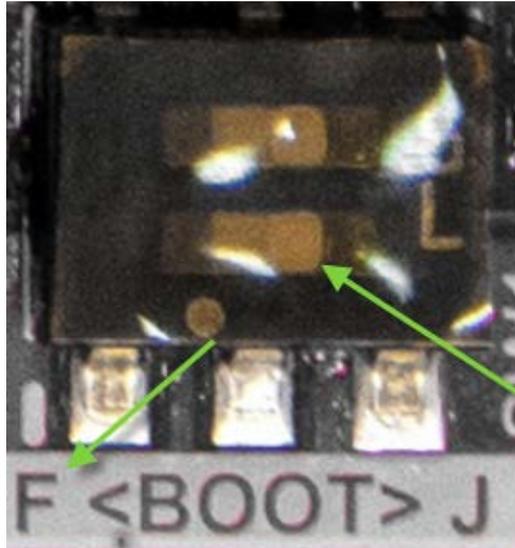


Figure 25 - Untouched Boot Switch

- b. If your MiniZed boot configuration switch is similar to the above, remove the protective film and slide switch 1 (indicated by the silkscreen DASH above the F) to be toggled to FLASH Booting (F).
26. Next plug the MiniZed into your PC in order to register the board with a COM port
 - a. Note: Windows 10 has been known to create two COM ports when plugging the MiniZed into the PC.
 - b. With a factory fresh board, open two instances of Tera Term, one for each COM port; 8,N,1,115200
 - c. Reboot your MiniZed using the Reset button (see Figure 29 – Reset Button)
 - d. In the Windows Device Manager (see Experiment 5, step 25) remove the COM port that did NOT have text appear

Experiment 5: Run the Design

27. Insert one USB cable into the MiniZed USB JTAG/UART MicroUSB connector.
28. Plug the other end into your PC.
29. If this is the first time you have connected your MiniZed, locate the COM Port assigned.
 - a. Right click on “My Computer” and choose Manage.
 - b. From here, select Device Manager
 - c. Under the Ports section, locate the USB Serial Port Assignment

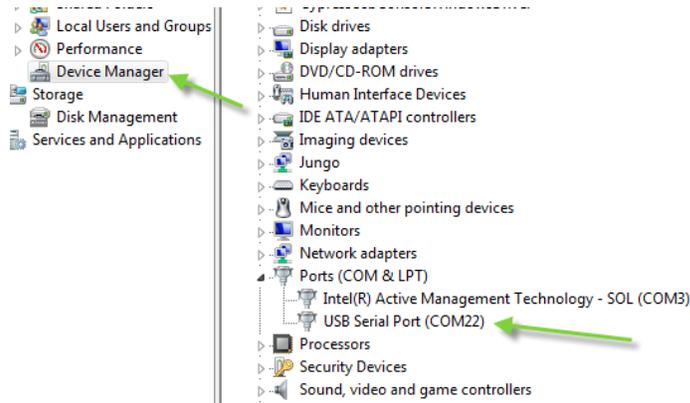


Figure 26 - Device Manager Used to Find USB Serial Port COM Port

30. Open a terminal program such as TeraTerm. Configure it to connect to the COM Port we found in the previous step using 115200/8/n/1/n as settings.
 - a. NOTE: if you open C:\Program Files (x86)\teraterm\TERATERM.INI you can change the defaults so you do not have to continuously modify the settings each time you open a serial port

```
; Serial port parameters
; Port number
ComPort=1
; Baud rate
BaudRate=9600
; Parity (even/odd/none/mark/space)
Parity=none

; Serial port parameters
; Port number
ComPort=4
; Baud rate
BaudRate=115200
; Parity (even/odd/none/mark/space)
Parity=none
```

Figure 27 – Changing TeraTerm Defaults

If this is the first time you have powered up your MiniZed, you will see a LOT of text as the default factory boot image is PetaLinux.

SDSoC builds BOOT.BIN files. While the intention is to be able to use these with SDCARD boot, such as with devices like UltraZed, this is just a standard BOOT.BIN file and can be treated as such for devices like MiniZed, which do not have an SDCARD cage.

31. Using explorer, navigate to the below and inspect for files. They will appear once the build is complete

```
C:\Avnet\SDx_workspace\MiniZedMM\Debug\sd_card
```

32. Once the build is complete, in the SDK Command Line Tool 2017.4, change the working directory to the location of BOOT.BIN

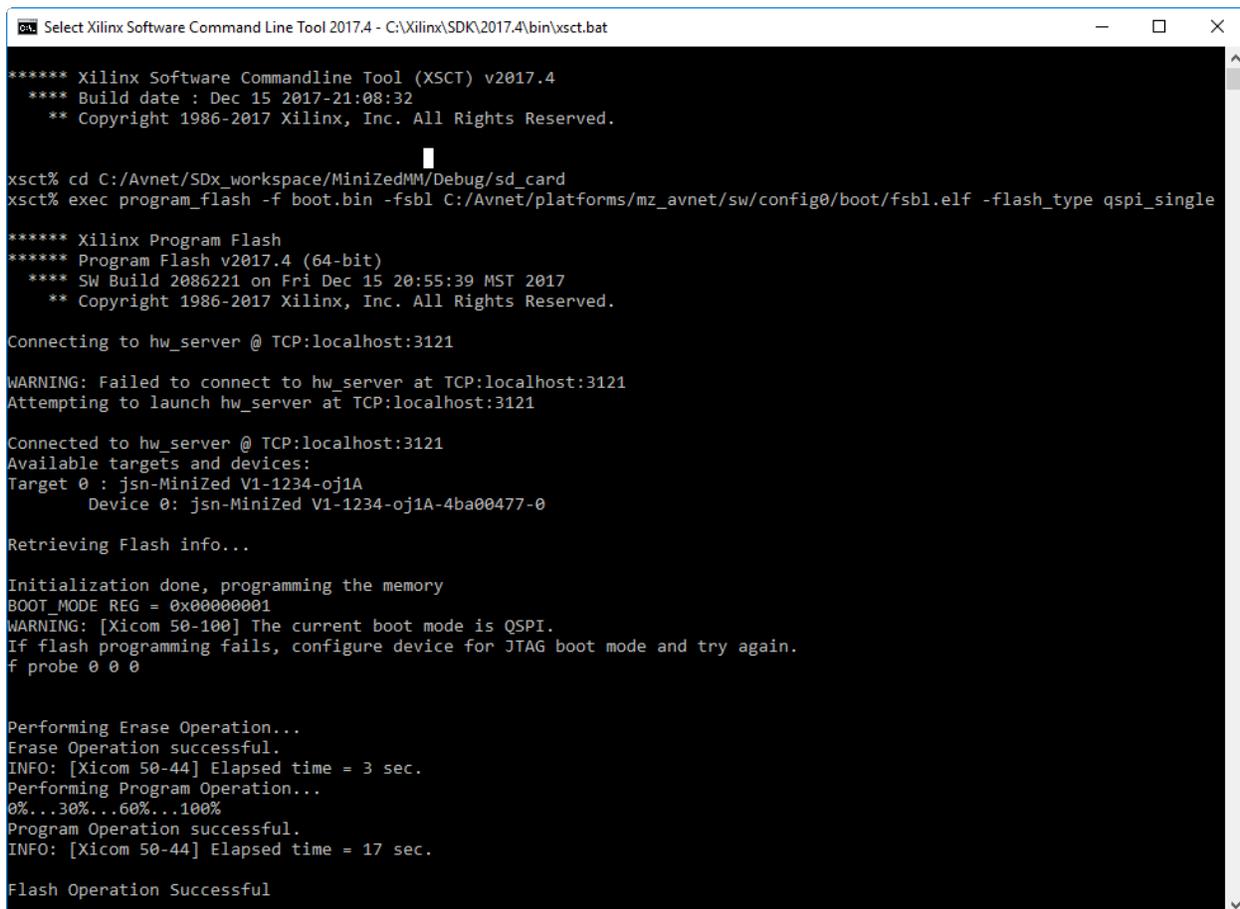
```
cd C:/Avnet/SDx_workspace/MiniZedMM/Debug/sd_card
```

33. Once there, execute the command to program the the QSPI over JTAG in the MiniZed.

```
exec program_flash -f boot.bin -fsbl  
C:/Avnet/platforms/mz_avnet/sw/config0/boot/fsbl.elf -flash_type  
qspi_single
```

a. Notice the blue DONE LED turns off. This indicates the image is being programmed, do not interrupt this process!

b. When complete, you should see the message, "Flash Operation Successful"



```
Select Xilinx Software Command Line Tool 2017.4 - C:\Xilinx\SDK\2017.4\bin\xsct.bat  
***** Xilinx Software Commandline Tool (XSCT) v2017.4  
**** Build date : Dec 15 2017-21:08:32  
** Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.  
  
xsct% cd C:/Avnet/SDx_workspace/MiniZedMM/Debug/sd_card  
xsct% exec program_flash -f boot.bin -fsbl C:/Avnet/platforms/mz_avnet/sw/config0/boot/fsbl.elf -flash_type qspi_single  
  
***** Xilinx Program Flash  
***** Program Flash v2017.4 (64-bit)  
**** SW Build 2086221 on Fri Dec 15 20:55:39 MST 2017  
** Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.  
  
Connecting to hw_server @ TCP:localhost:3121  
  
WARNING: Failed to connect to hw_server at TCP:localhost:3121  
Attempting to launch hw_server at TCP:localhost:3121  
  
Connected to hw_server @ TCP:localhost:3121  
Available targets and devices:  
Target 0 : jsn-MiniZed V1-1234-oj1A  
Device 0: jsn-MiniZed V1-1234-oj1A-4ba00477-0  
  
Retrieving Flash info...  
  
Initialization done, programming the memory  
BOOT_MODE REG = 0x00000001  
WARNING: [Xicom 50-100] The current boot mode is QSPI.  
If flash programming fails, configure device for JTAG boot mode and try again.  
f probe 0 0 0  
  
Performing Erase Operation...  
Erase Operation successful.  
INFO: [Xicom 50-44] Elapsed time = 3 sec.  
Performing Program Operation...  
0%..30%..60%..100%  
Program Operation successful.  
INFO: [Xicom 50-44] Elapsed time = 17 sec.  
  
Flash Operation Successful
```

Figure 28 – Programming MiniZed QSPI

34. Close XSCT and the Explorer window. Note that if you were to rebuild and the two windows were left open, the SDx tools will COMPLETE all building, BIT FILE, HLS IPI generation, etc., and JUST as it is ready to combine into a boot.bin, it will fail the build as XCST% will hold the directory while the SDx tool tried to delete – thus failing after the 20 minute+ build. Therefore, closing XSCT is critical.
35. Once you get the successful message, press the RESET button on the MiniZed

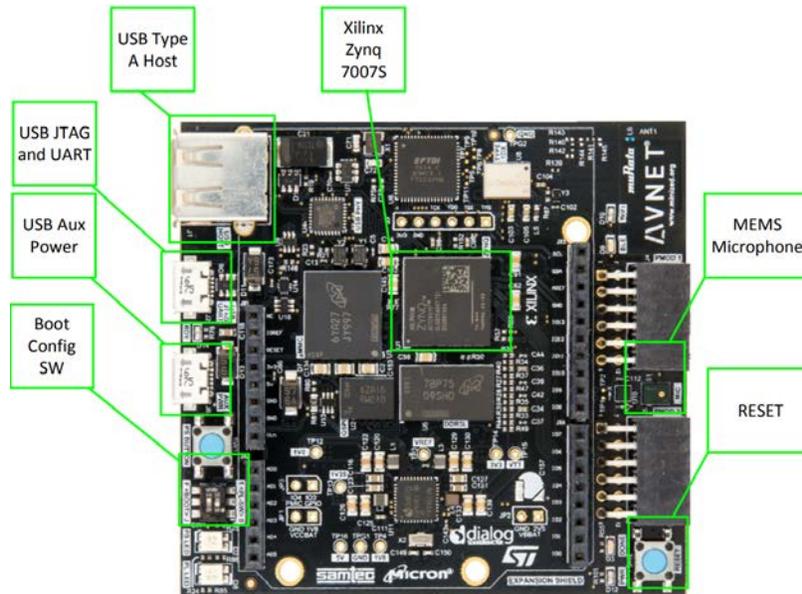


Figure 29 – Reset Button

36. The BLUE Done should light up again and you will see after each press of the reset, the Matrix Multiply will run.
37. You can now CLOSE the SDx IDE.

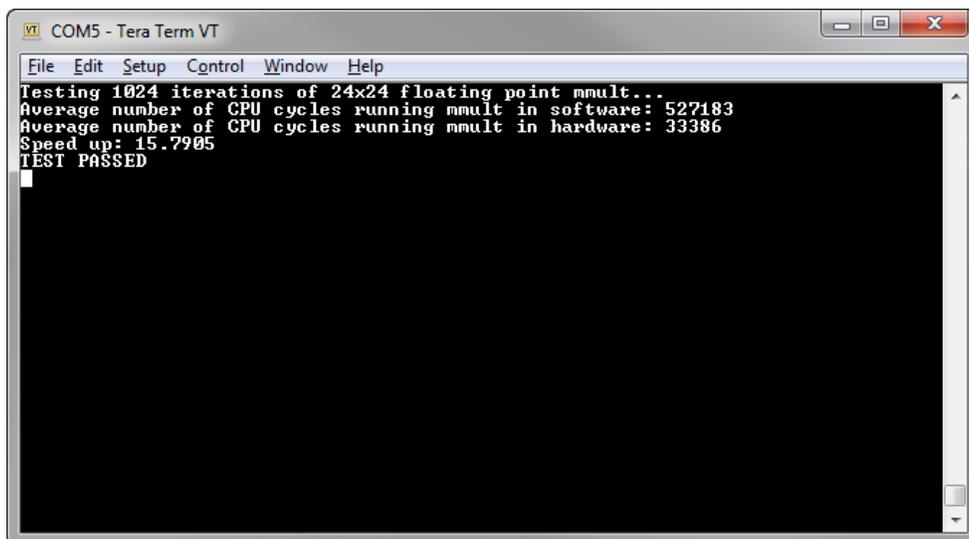


Figure 30 – Matrix Multiply Running in Both Software and Hardware

Appendix A: Importing Files

In order to import source files into our project, the operational flow is the same as with Vivado SDK. These basic steps are laid out in the following procedure. This is only necessary if you are importing files instead of using the example built into the provided platform.

1. Right click on the MiniZedMM src folder and select **Import...**

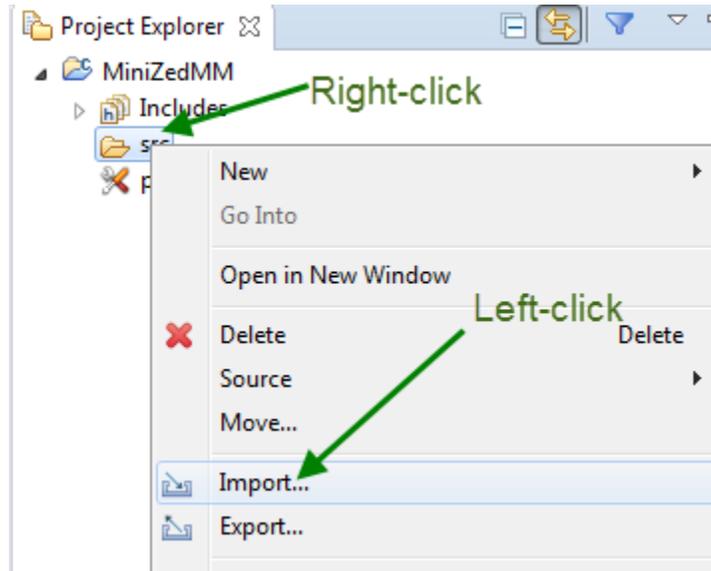


Figure 31 – Import Source Files

2. Select **General**, then **File System** and **Next >**.

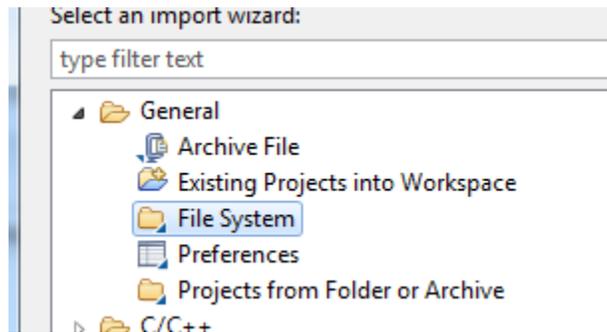


Figure 32 – Import from the File System

3. For the “From Directory” field, either copy/paste the location below or click on **Browse** then browse to the following example location:

```
C:\Xilinx\SDx\2017.4\samples\mmult
```

- Next select three checkboxes next to the .cpp and .h files. Ensure that the **Into folder** box shows MiniZedMM/src as well as the checkboxes matches the screen in the below figure.

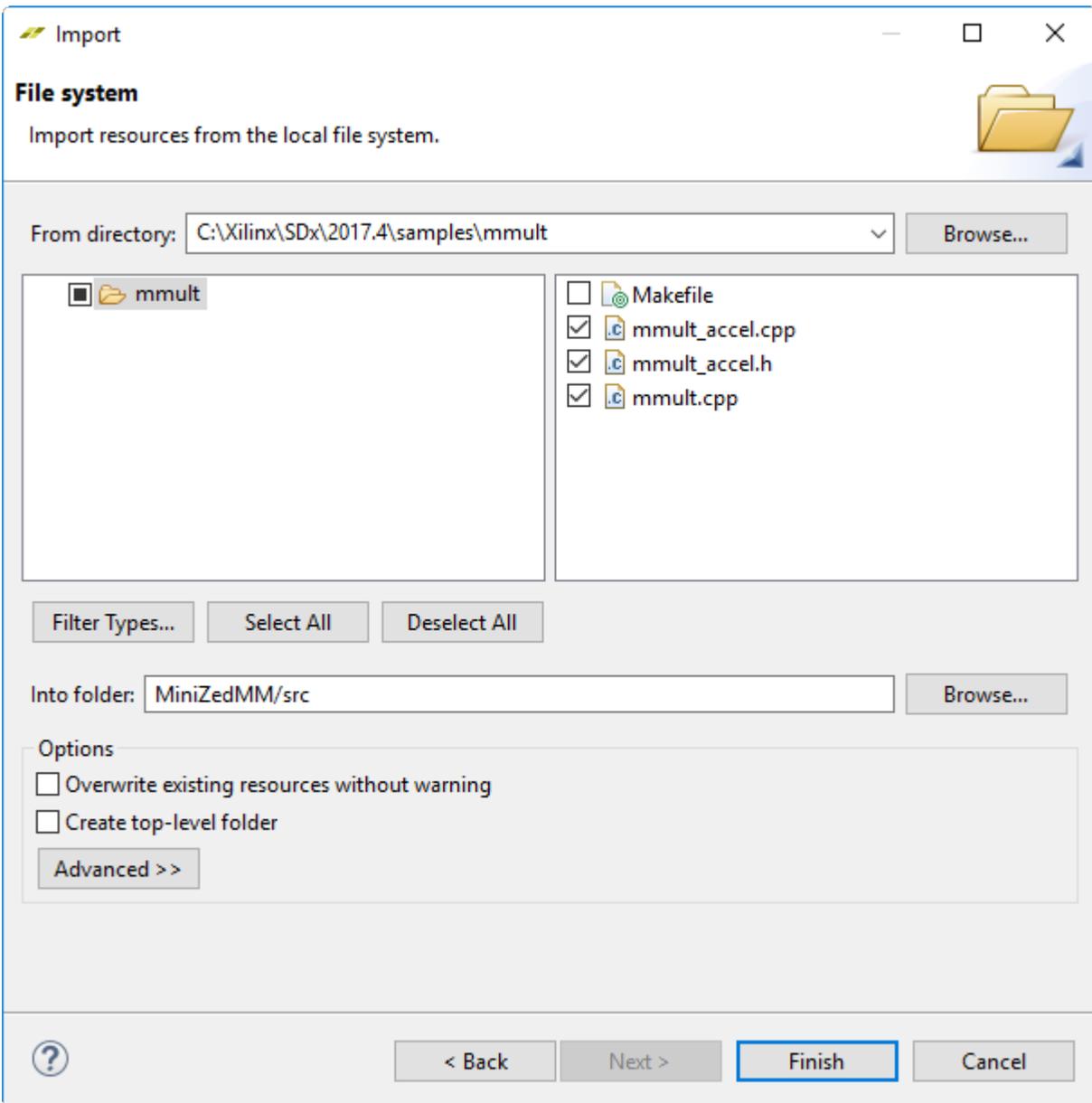


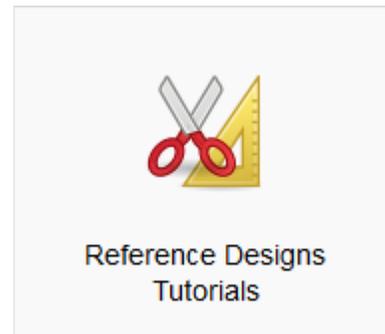
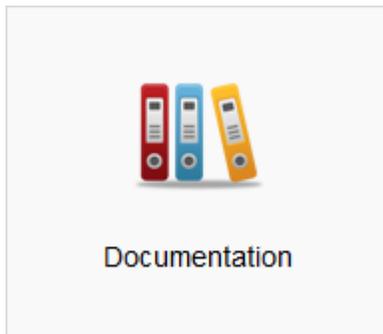
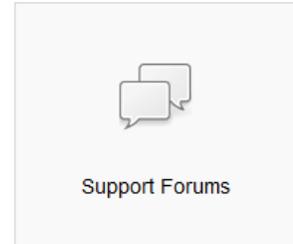
Figure 33 – Import Example Files

- Click **Finish**. You should now see the three sources added in Project Explorer.

Appendix B: Getting Support

Avnet Support

- Technical support is offered online through the minized.org website support forums. MiniZed users are encouraged to participate in the forums and offer help to others when possible.
- To access the most current collateral for the MiniZed, visit the community support page (www.minized.org/content/support) and click one of the icons shown below:



- MiniZed Documentation
<http://minized.org/support/documentation/18891>
- MiniZed Reference Designs
<http://minized.org/support/design/18891/146>

Xilinx Support

The following technical support options are available to Xilinx customers:

- Technical information is available online 24 hours a day from the [Support website](#)
- Technical Support staff are available to respond to your questions in the [Community Forums](#)
- Individual assistance from Xilinx Technical Support **may** be available through [Service Portal](#)
- Phone support is **only** available with an active open case number

Global Phone Number

Region	Language	Phone**	Support Hours*
North America	EN	1 800-255-7778 or +1 408-879-5199	M-F 7:00 -17:00 PST
Europe, Middle East and Africa	EN, DE, FR	00 800-5152-5152 or +353 1-461-5700	M-F 8:00 -17:00 GMT
China	CH (Mandarin), EN	+86 800 988 0218 +86 400 880 0218 (Mobile Phone)	M-F* 9:00 -18:00 CST
Taiwan	CH (Mandarin), EN	+886 2-8176-1060	M-F 9:00 -18:00 CST
Hong Kong	CH (Mandarin), EN	+852 3187-3855	M-F 9:00 -18:00 CST

* Support hours listed apply for both standard and daylight savings (summer) time. Please check the [Technical Support Holiday Calendar 2016](#) for support availability during holidays in your region.

** 00 800-5152-5152 is a international free phone (toll free) number available in the following countries: Austria, Belgium, Denmark, Finland, France, Germany, Ireland Israel, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, and United Kingdom. All other countries must use +353 1-461-5700.

** For the numbers listed, '+' represents the International Direct Dialing (IDD) prefix of the country from which you are calling. Please consult your local telephone service provider for more information on specific IDD instructions.

Revision History

Date	Version	Revision
25 Aug 17	00	Preliminary release
28 Aug 17	00	Edits per Author
17 Sep 17	1	First public release
19 Apr 18	2	Updated to SDSoc 2017.4