MiniZed: Boot from QSPI using FSBL

Overview

Thus far, we have relied on the tools to configure the Zynq PS properly. Although it wasn't explicitly pointed out previously, performing a *Run As* or *Debug As* operation in SDK first sources the `ps7_init.tcl` file that was created during the export to SDK. In a true, embedded application, you will not have a JTAG cable connection that can transfer these settings. Your code must be able to do this before transferring control to an application. The code that sets up the Zynq PS is called the First Stage Boot Loader (FSBL).

Objectives

When this tutorial is complete, you will be able to:

- Create the FSBL
- Prepare the boot image
- Write and boot from QSPI

Experiment Setup

Software

The software used to test this reference design is:

- Windows-7 64-bit
- Xilinx SDK 2017.1
- FTDI FT2232H device driver

Hardware

The hardware setup used to test this reference design includes:

- Win-7 PC with the following recommended memory[1]:
  - 1.6 GB RAM available for the Xilinx tools to complete a XC7Z010 design
  - 2.3 GB RAM available for the Xilinx tools to complete a XC7Z015 design
  - 1.9 GB RAM available for the Xilinx tools to complete a XC7Z020 design
  - 2.7 GB RAM available for the Xilinx tools to complete a XC7Z030 design
- MiniZed
- USB cable (Type A to Micro-USB Type B)

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

LIT#

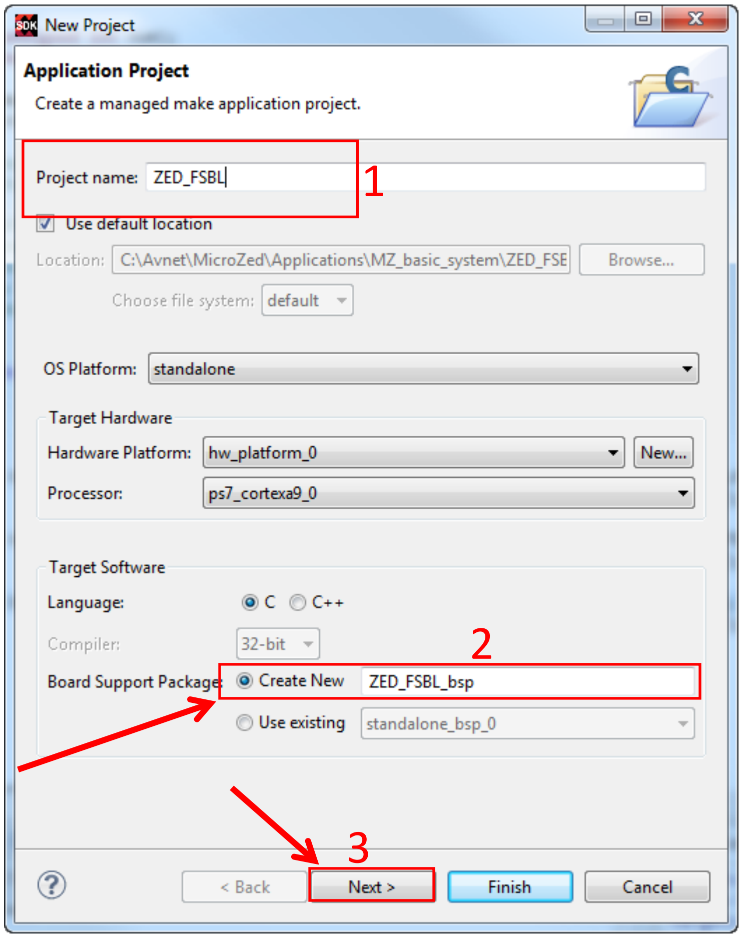
[1] Refer to www.xilinx.com/design-tools/vivado/memory.htm

Experiment 1: Create the FSBL

The first step is to create the FSBL application. This is a C program that embeds all the Zynq internal register settings that were established during the Vivado Block Design.

Similar to the flow for creating the Hello_World application, in SDK create a First Stage Bootloader application.

1. Launch SDK and open the workspace from the Hello World project.
2. **File → New → New Application Project**
3. Name it something like ZED_FSBL and **Create New** BSP. The reason for creating a new BSP is that the FSBL BSP requires a library for the Flash, and the tools will automatically include this for us if we allow it to create the BSP. Click **Next**


New Project

Application Project

Create a managed make application project.

Project name: ZED_FSBL

[✓] Use default location

Location: C:\Avnet\MicroZed\Applications\MZ_basic_system\ZED_FSE

Choose file system: default

OS Platform: standalone

Target Hardware

Hardware Platform: hw_platform_0

Processor: ps7_cortexa9_0

Target Software

Language: [✓ C] C++

Compiler: 32-bit

Board Support Package: [✓ Create New] ZED_FSBL_bsp

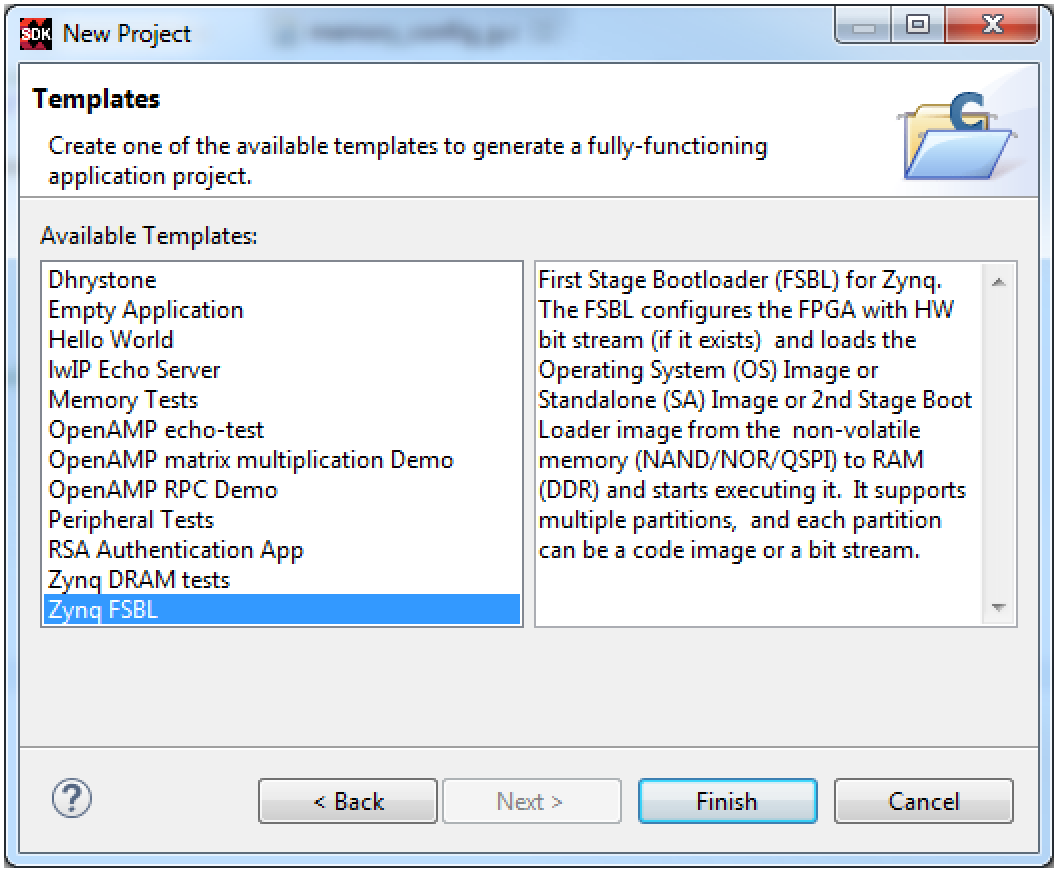
Use existing standalone_bsp_0

Figure 1 – FSBL Application

Copyright © 2017 Avnet, Inc. AVNET, “Reach Further,” and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

4. Select **Zynq FSBL**

5. Click **Finish**



Templates

Create one of the available templates to generate a fully-functioning application project.

Available Templates:

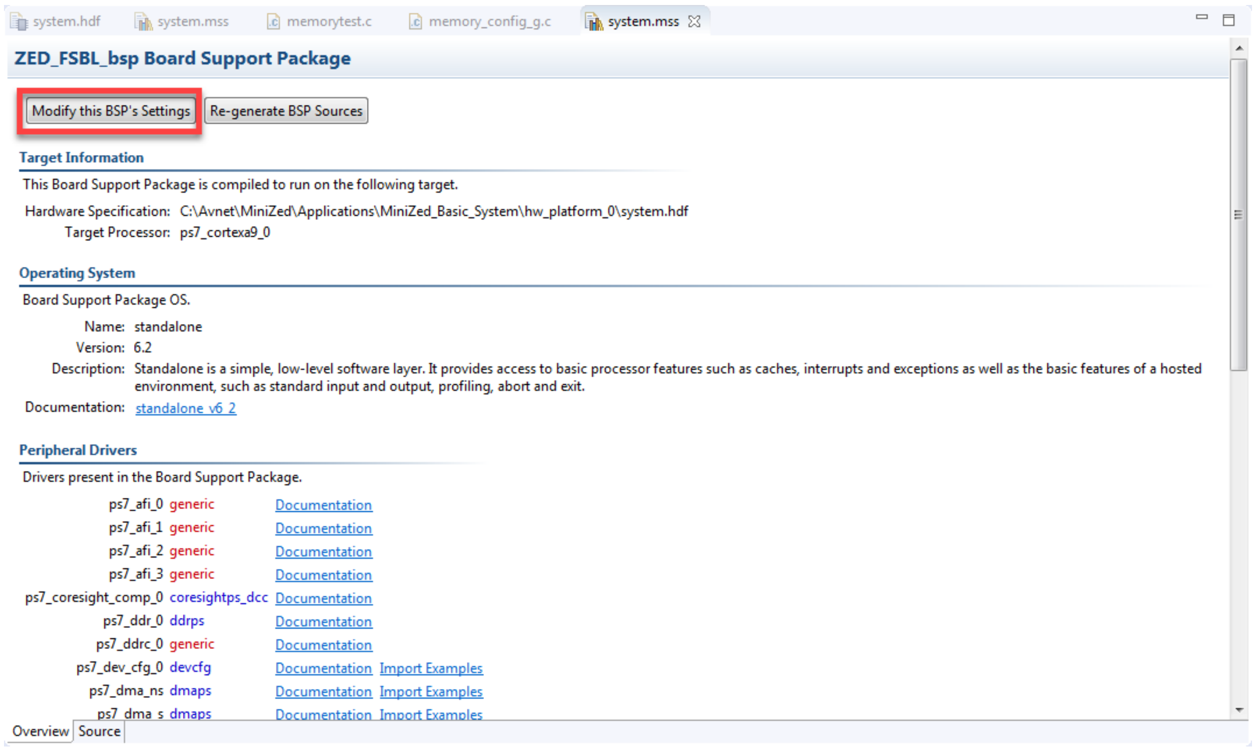
Dhrystone
Empty Application
Hello World
lwIP Echo Server
Memory Tests
OpenAMP echo-test
OpenAMP matrix multiplication Demo
OpenAMP RPC Demo
Peripheral Tests
RSA Authentication App
Zynq DRAM tests
Zynq FSBL

First Stage Bootloader (FSBL) for Zynq. The FSBL configures the FPGA with HW bit stream (if it exists) and loads the Operating System (OS) Image or Standalone (SA) Image or 2nd Stage Boot Loader image from the non-volatile memory (NAND/NOR/QSPI) to RAM (DDR) and starts executing it. It supports multiple partitions, and each partition can be a code image or a bit stream.

Figure 2 – Zynq FSBL Template for Application

6. Similar to the standalone_bsp_0 we created in lab 2, we must modify the BSP's UART settings to properly output our results through the serial port. In the ZED_FSBL_bsp system.mss file **click** on "Modify this BSP's Settings".

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.



ZED_FSBL_bsp Board Support Package

Target Information

This Board Support Package is compiled to run on the following target.

Hardware Specification: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\hw_platform_0\system.hdf
Target Processor: ps7_cortexa9_0

Operating System

Board Support Package OS.

Name: standalone
Version: 6.2
Description: Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
Documentation: standalone_v6_2

Peripheral Drivers

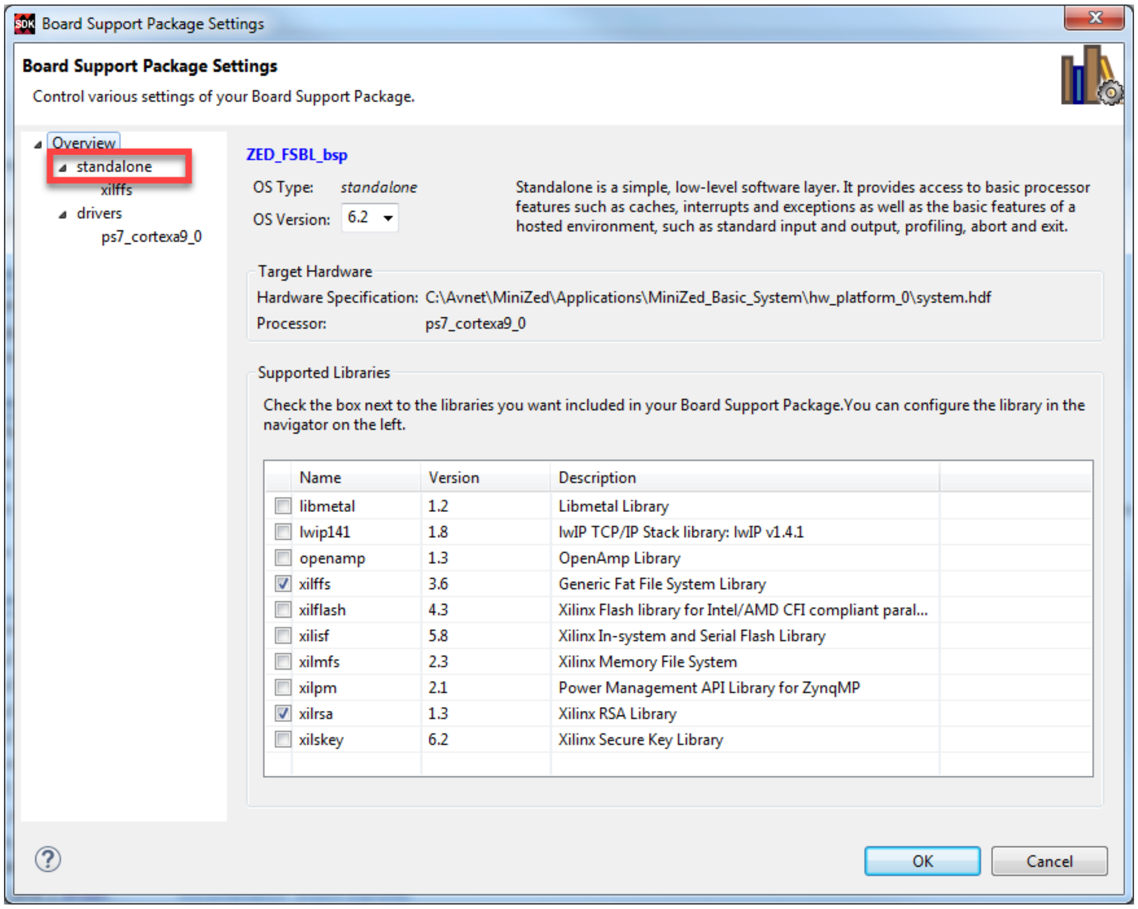
Drivers present in the Board Support Package.

ps7_afi_0	generic	Documentation	
ps7_afi_1	generic	Documentation	
ps7_afi_2	generic	Documentation	
ps7_afi_3	generic	Documentation	
ps7_coresight_comp_0	coresightps_dcc	Documentation	
ps7_ddr_0	ddrps	Documentation	
ps7_ddrc_0	generic	Documentation	
ps7_dev_cfg_0	devcfg	Documentation	Import Examples
ps7_dma_ns	dmaps	Documentation	Import Examples
ps7_dma_s	dmaps	Documentation	Import Examples

Figure 3 -- Modify BSP

7. In the *Board Support Package Settings*, select standalone

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.



Board Support Package Settings
Board Support Package Settings
Control various settings of your Board Support Package.
Overview
standalone
xilffs
drivers
ps7_cortexa9_0
OK
Cancel

ZED_FSBL_bsp

OS Type: *standalone*

OS Version: 6.2

Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Target Hardware

Hardware Specification: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\hw_platform_0\system.hdf

Processor: ps7_cortexa9_0

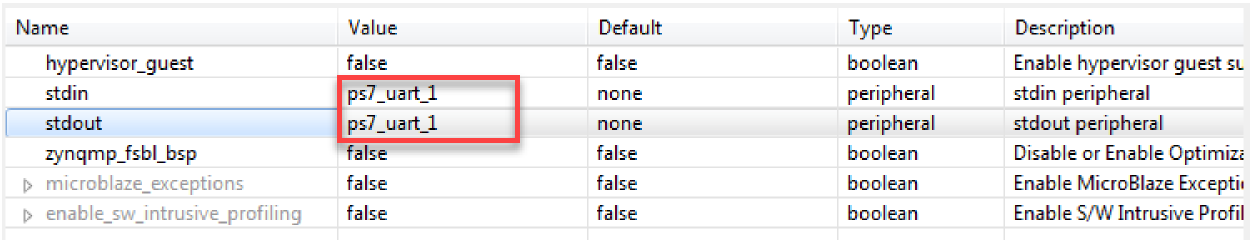
Supported Libraries

Check the box next to the libraries you want included in your Board Support Package.You can configure the library in the navigator on the left.

	Name	Version	Description
☐	libmetal	1.2	Libmetal Library
☐	lwip141	1.8	lwIP TCP/IP Stack library: lwIP v1.4.1
☐	openamp	1.3	OpenAmp Library
✓	xilffs	3.6	Generic Fat File System Library
☐	xilflash	4.3	Xilinx Flash library for Intel/AMD CFI compliant paral...
☐	xilisf	5.8	Xilinx In-system and Serial Flash Library
☐	xilmfs	2.3	Xilinx Memory File System
☐	xilpm	2.1	Power Management API Library for ZynqMP
✓	xilrsa	1.3	Xilinx RSA Library
☐	xilskey	6.2	Xilinx Secure Key Library

Figure 4 – Board Support Package Settings

8. Change the stdin and stdout Value to ps7_uart. This is done to align with the MiniZed's UART Serial connection. Select OK.



Name	Value	Default	Type	Description
hypervisor_guest	false	false	boolean	Enable hypervisor guest su
stdin	ps7_uart_1	none	peripheral	stdin peripheral
stdout	ps7_uart_1	none	peripheral	stdout peripheral
zynqmp_fsbl_bsp	false	false	boolean	Disable or Enable Optimiz
▷ microblaze_exceptions	false	false	boolean	Enable MicroBlaze Excepti
▷ enable_sw_intrusive_profiling	false	false	boolean	Enable S/W Intrusive Profil

Figure 5 – Modify stdin/stdout

Based on the modified settings in SDK, the BSP will automatically be built once it is added to the project. This may take a minute to compile the new BSP. The progress may be seen in the *Console* tab.

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

LIT#

Experiment 2: Prepare the Boot Image

The next step is to create a non-volatile boot image. MiniZed has one non-volatile, primary bootable source, QSPI.

1. In SDK, select Periph_Test.

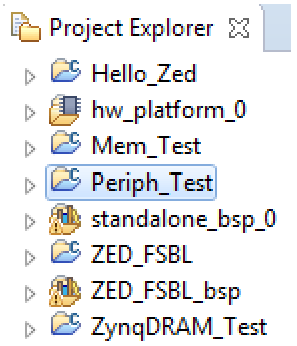


Figure 6 – Select Application to Boot

2. Select **Xilinx Tools → Create Boot Image**.

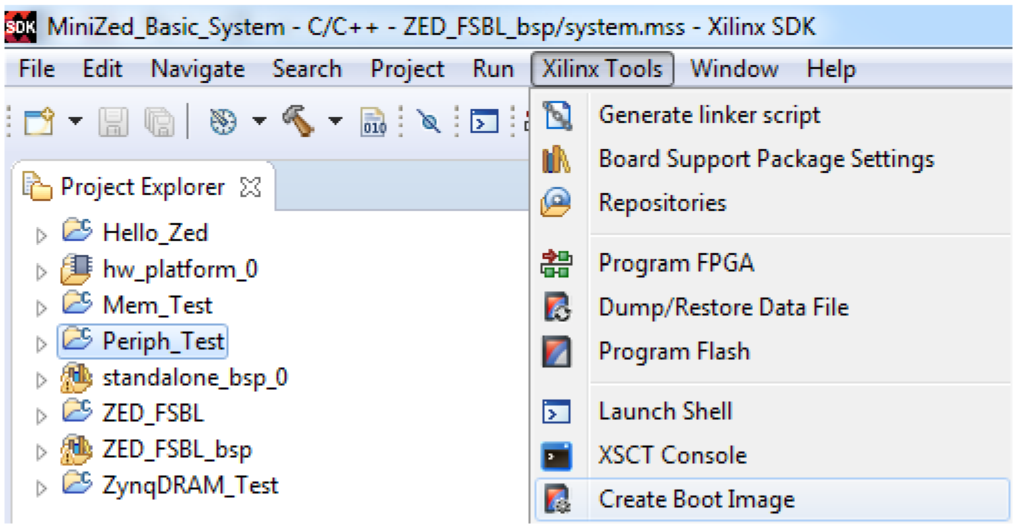
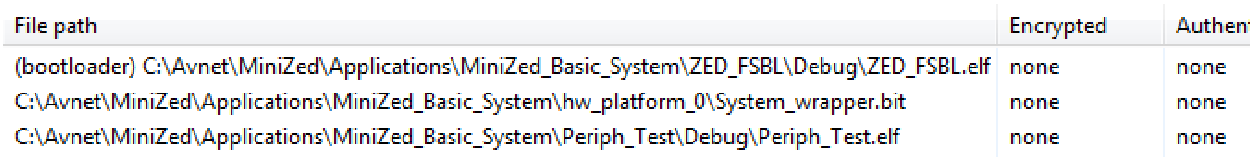


Figure 7 - Create Zynq Boot Image

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

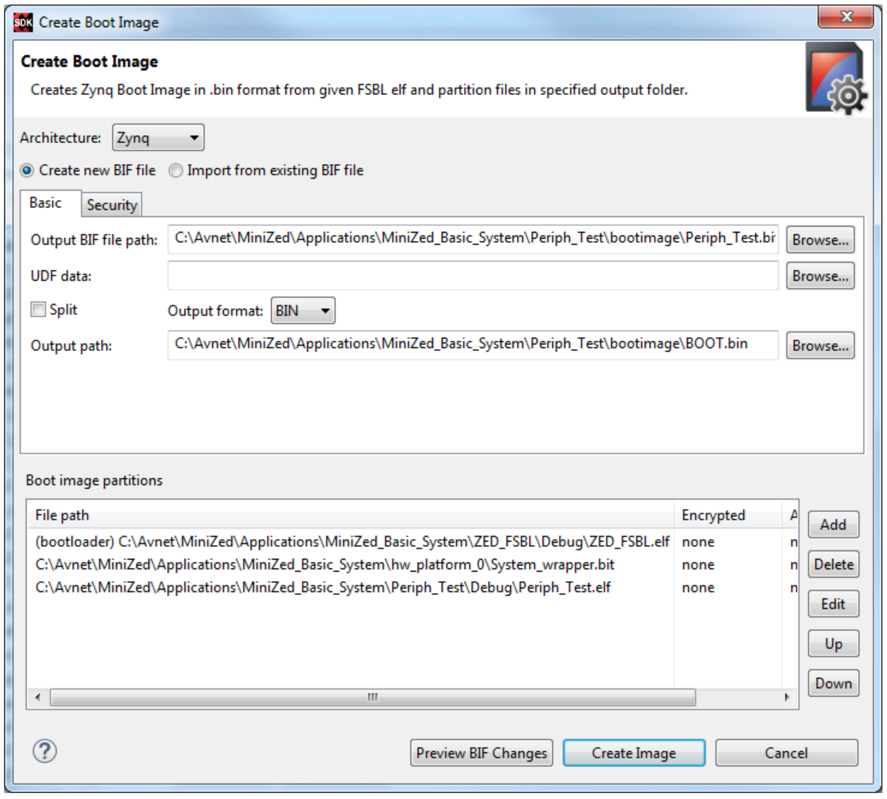
This will preload the FSBL ELF, bitstream, and Application ELF images. The order of the files is important. The FSBL is first, followed by the bitstream, followed by the Application. One function of the FSBL is to program the PL. After the PL is configured, the application is loaded.



File path	Encrypted	Authen
(bootloader) C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\ZED_FSBL\Debug\ZED_FSBL.elf	none	none
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\hw_platform_0\System_wrapper.bit	none	none
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\Debug\Periph_Test.elf	none	none

Figure 8 – Boot Image Partitions

Notice that by default, the output path points to BOOT.bin (highlighted), which is the bootimage required for SD boot. Unfortunately the MiniZed by default is unable to boot from SD Card, the output image must be changed to generate a QSPI bootimage (mcs).



Create Boot Image

Create Boot Image

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

Architecture: Zynq

(●) Create new BIF file ( ) Import from existing BIF file

Basic | Security

Output BIF file path: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\bootimage\Periph_Test.bi

UDF data:

☐ Split Output format: BIN

Output path: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\bootimage\BOOT.bin

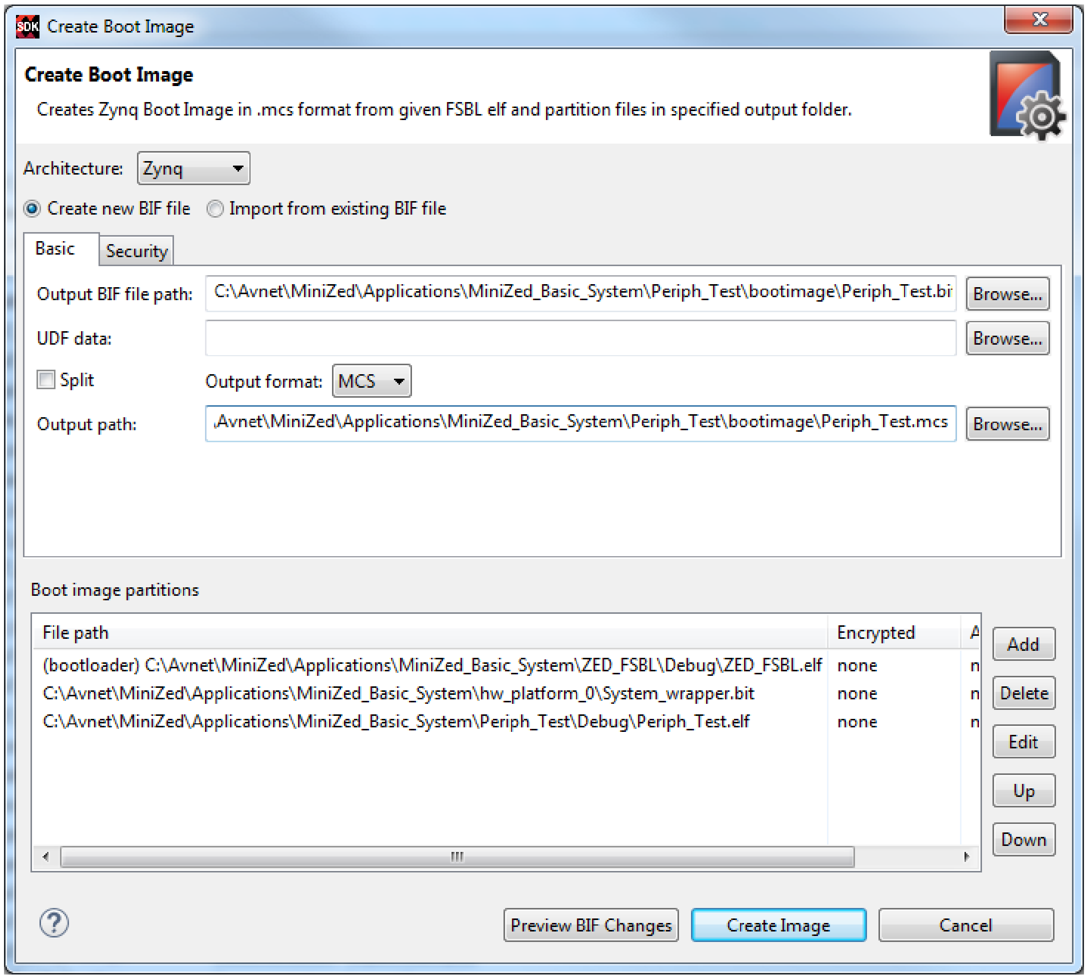
Boot image partitions

File path	Encrypted	A
(bootloader) C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\ZED_FSBL\Debug\ZED_FSBL.elf	none	n
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\hw_platform_0\System_wrapper.bit	none	n
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\Debug\Periph_Test.elf	none	n

Figure 9 – Create Image

3. Change the *Output format* to ***.mcs**. Change Boot.mcs to **`Periph_Test.mcs`** for the *File name*.

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.
LIT#


Create Boot Image

Create Boot Image

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

Architecture: Zynq

(●) Create new BIF file ( ) Import from existing BIF file

Basic | Security

Output BIF file path: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\bootimage\Periph_Test.bi

UDF data:

[ ] Split   Output format: MCS

Output path: Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\bootimage\Periph_Test.mcs

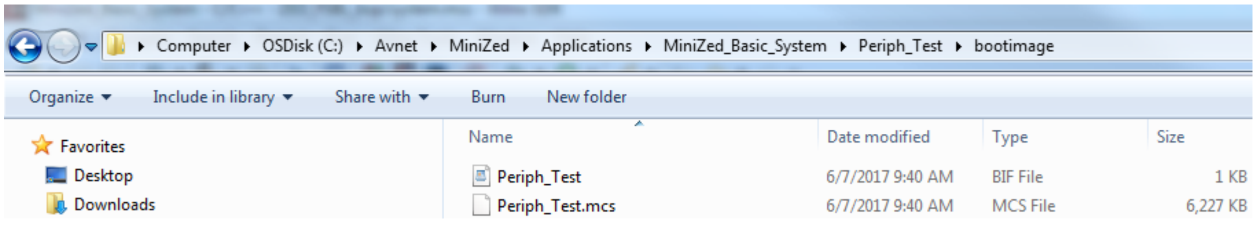
Boot image partitions

File path	Encrypted	A
(bootloader) C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\ZED_FSBL\Debug\ZED_FSBL.elf	none	n
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\hw_platform_0\System_wrapper.bit	none	n
C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\Debug\Periph_Test.elf	none	n

Figure 10 – Creating the QSPI Bootimage

4. Click **Create Image**.

5. Using Windows Explorer, navigate to the application directory, then into `Periph_Test`, then into the newly created **bootimage** directory. Notice that two files have been created: .bif, .mcs. These are all the required files to program either the QSPI or SD Card.


Computer ▸ OSDisk (C:) ▸ Avnet ▸ MiniZed ▸ Applications ▸ MiniZed_Basic_System ▸ Periph_Test ▸ bootimage

Name	Date modified	Type	Size
Periph_Test	6/7/2017 9:40 AM	BIF File	1 KB
Periph_Test.mcs	6/7/2017 9:40 AM	MCS File	6,227 KB

Figure 11 - Bootimage Directory

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

Experiment 3: Write and boot from QSPI

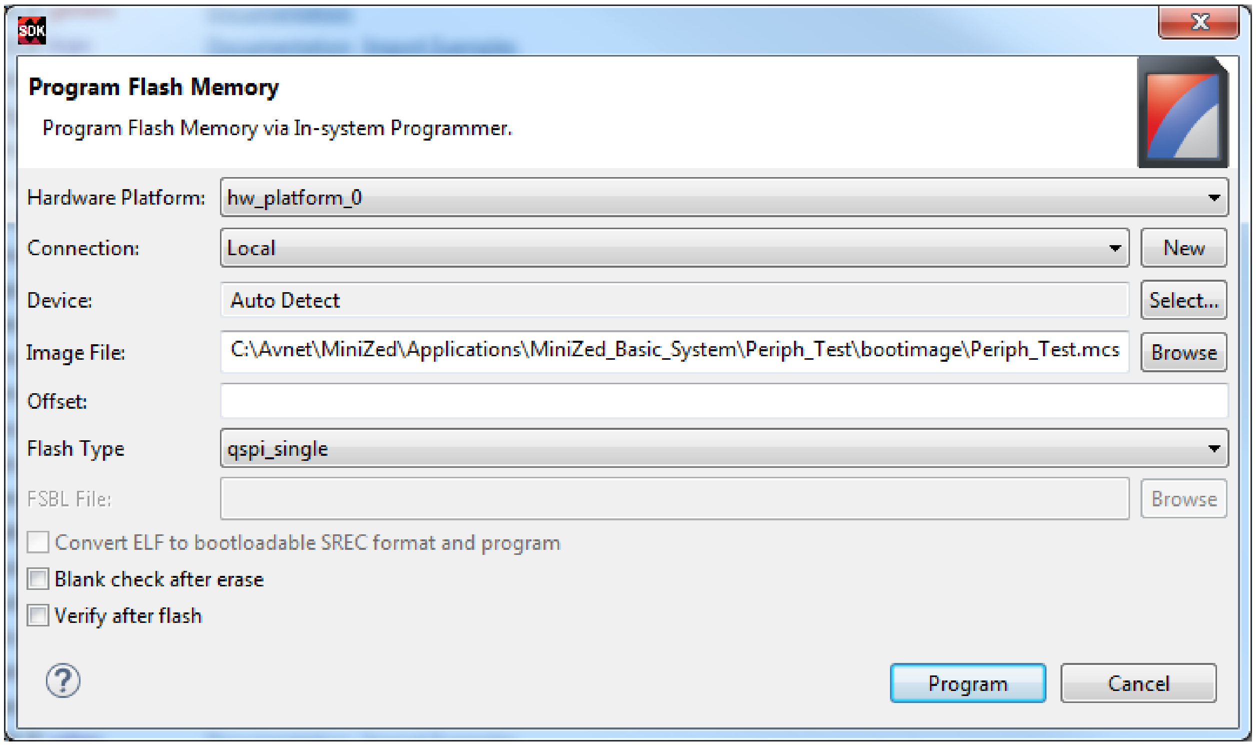
First, we will program the QSPI with the MCS file from within SDK.

1. Set up the MiniZed for Cascaded JTAG mode as before. Attach the micro-USB to J2 and your computer.

2. In SDK, select **Xilinx Tools → Program Flash**.

3. Click the **Browse** button, and browse to the MCS image for the application you wish to program. Select it and click **Open**. You can selectively enable the *Blank check after erase* and/or the *Verify after flash* functions although each of these will add time to the procedure. Neither is required to program the Flash.

4. Click **Program**.


Program Flash Memory

Program Flash Memory via In-system Programmer.

Hardware Platform: hw_platform_0

Connection: Local

Device: Auto Detect

Image File: C:\Avnet\MiniZed\Applications\MiniZed_Basic_System\Periph_Test\bootimage\Periph_Test.mcs

Offset:

Flash Type: qspi_single

FSBL File:

☐ Convert ELF to bootloadable SREC format and program

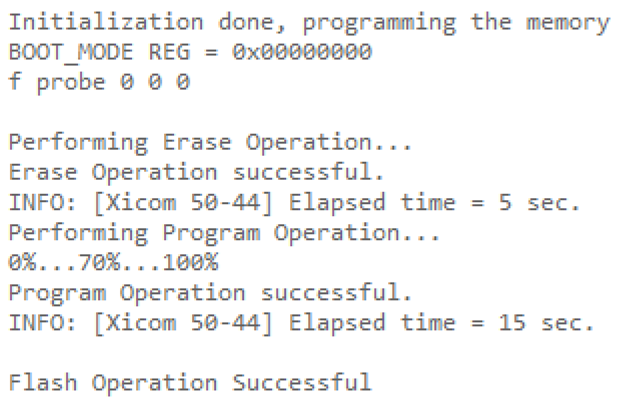
☐ Blank check after erase

☐ Verify after flash

Figure 12 – Program QSPI Flash Memory

5. The operation took approximately 40 seconds on a Win7 PC. You should see the following in the Program Flash Console window (if you see nothing in the console window, click the console pull-down and select Program Flash).

Copyright © 2017 Avnet, Inc. AVNET, “Reach Further,” and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.
LIT#



```
Initialization done, programming the memory
BOOT_MODE REG = 0x00000000
f probe 0 0 0

Performing Erase Operation...
Erase Operation successful.
INFO: [Xicom 50-44] Elapsed time = 5 sec.
Performing Program Operation...
0%...70%...100%
Program Operation successful.
INFO: [Xicom 50-44] Elapsed time = 15 sec.

Flash Operation Successful
```

Figure 13 – QSPI Programmed Successfully

6. Power off the board by unplugging J2 micro-USB connector.
7. Set the MiniZed boot mode switch SW1 to QSPI mode (‘F’ for Flash) as shown below.

Copyright © 2017 Avnet, Inc. AVNET, “Reach Further,” and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.
LIT#

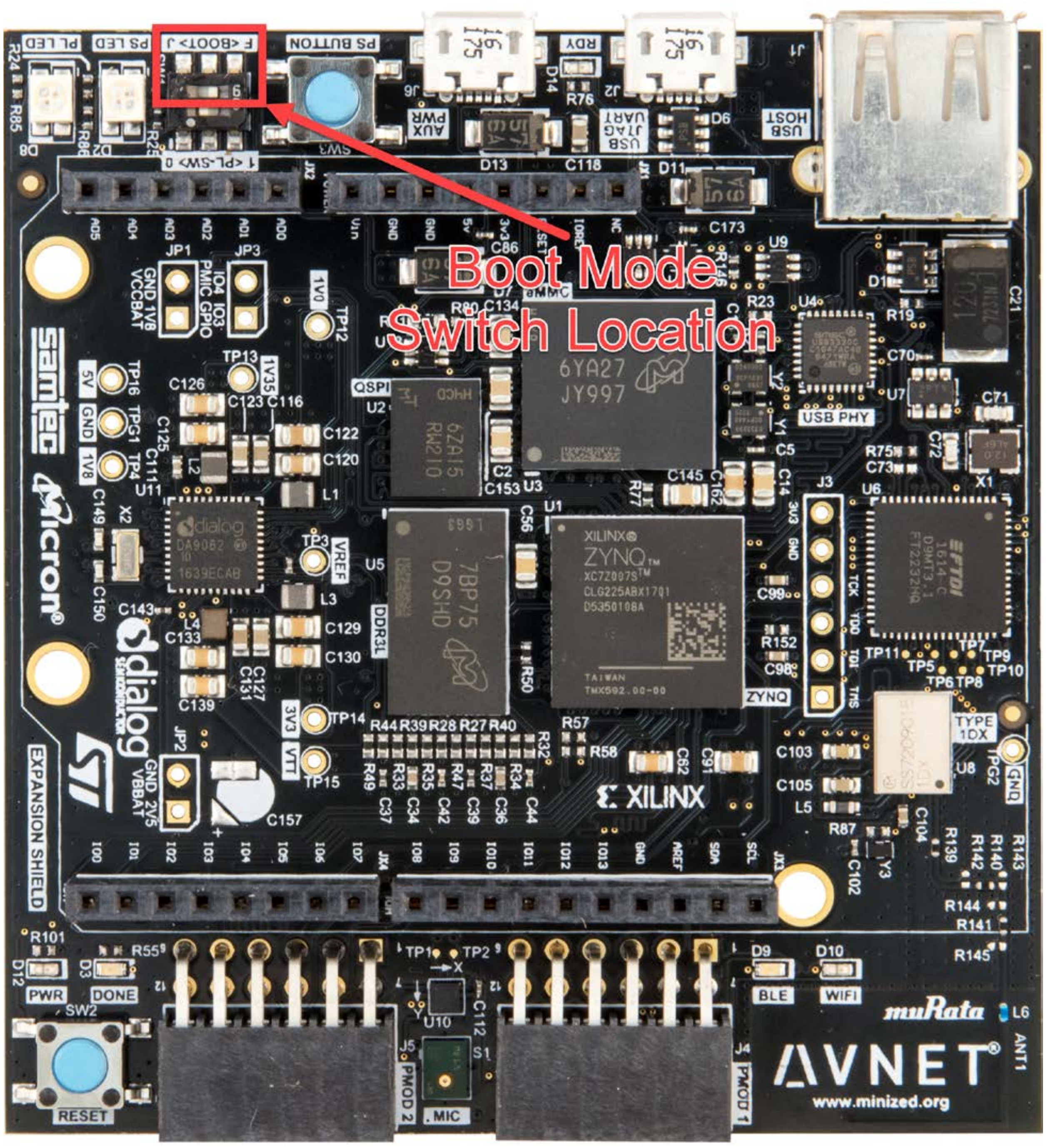


Figure 14 – MiniZed Switch Location

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.
LIT#

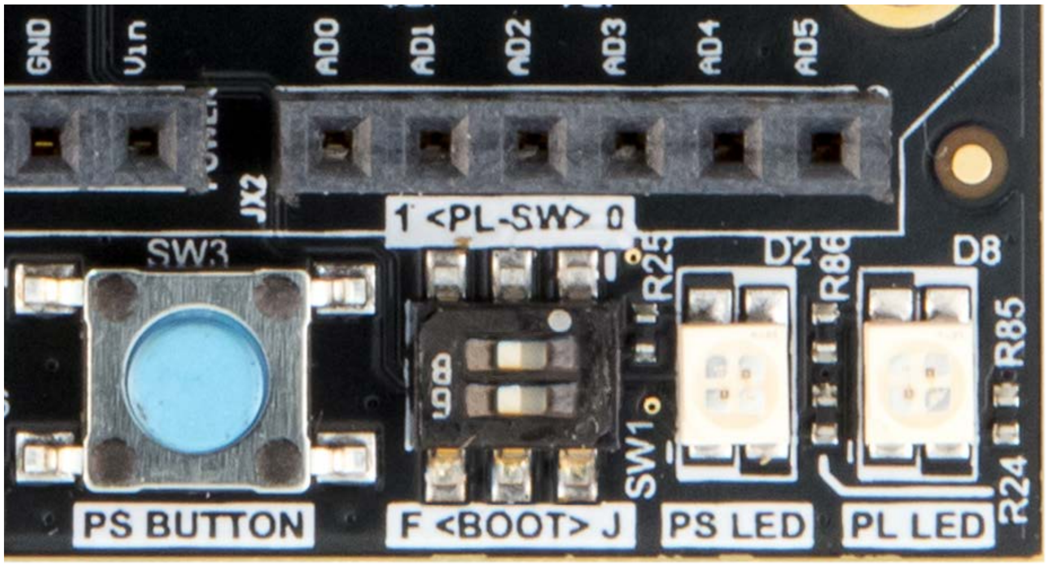
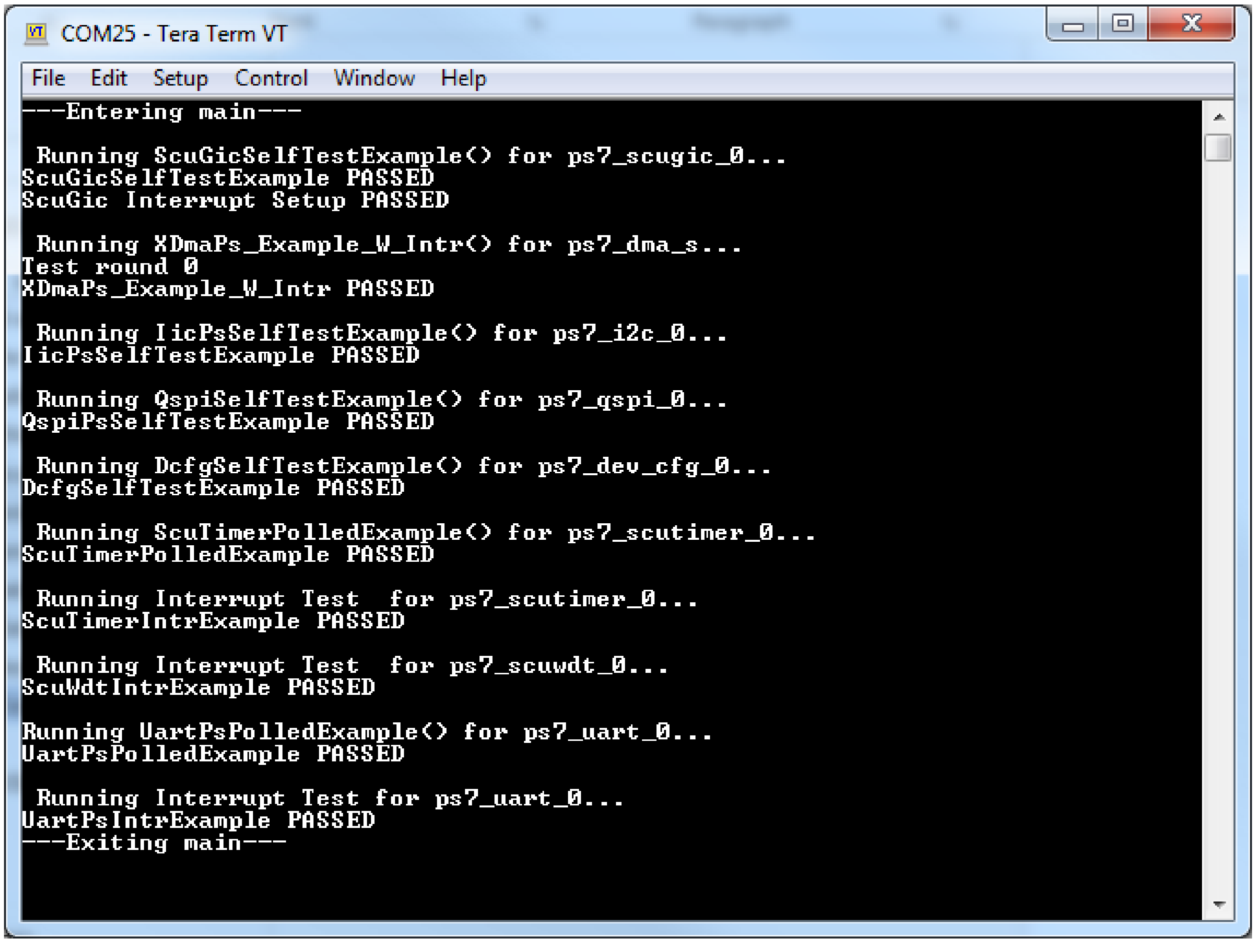


Figure 15 – QSPI/Flash Boot Mode

8. Close or disconnect the terminal that may have previously been open on your PC.
9. Power on the board by reconnecting the USB-JTAG-UART J2 connection. A blue LED should illuminate meaning the bitstream was loaded.
10. Launch a terminal program (TeraTerm) with the 115200/8/n/1/n settings.
11. Push the RESET button (SW2). You should see the results in the terminal.

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.
LIT#



```
---Entering main---

 Running ScuGicSelfTestExample() for ps7_scugic_0...
ScuGicSelfTestExample PASSED
ScuGic Interrupt Setup PASSED

 Running XDmaPs_Example_W_Intr() for ps7_dma_s...
Test round 0
XDmaPs_Example_W_Intr PASSED

 Running IicPsSelfTestExample() for ps7_i2c_0...
IicPsSelfTestExample PASSED

 Running QspiSelfTestExample() for ps7_qspi_0...
QspiPsSelfTestExample PASSED

 Running DcfgSelfTestExample() for ps7_dev_cfg_0...
DcfgSelfTestExample PASSED

 Running ScuTimerPolledExample() for ps7_scutimer_0...
ScuTimerPolledExample PASSED

 Running Interrupt Test  for ps7_scutimer_0...
ScuTimerIntrExample PASSED

 Running Interrupt Test  for ps7_scuwdt_0...
ScuWdtIntrExample PASSED

Running UartPsPolledExample() for ps7_uart_0...
UartPsPolledExample PASSED

 Running Interrupt Test for ps7_uart_0...
UartPsIntrExample PASSED
---Exiting main---
```

Figure 16 – Results from QSPI boot of Periph_Test

12. Close the terminal. Power off the board.

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.

Revision History

Date	Version	Revision
26 Aug 2013	2013_2.01	Initial Avnet release for Vivado 2013.2
06 Sep 2013	2013_2.02	Switching to Release build rather than Debug
10 Jun 2014	2014_1.01	Update to 2014.1. Added instructions that .bin and .mcs must each be generated individually now.
11 Jun 2014	2014_2.01	Update to 2014.2. Bitstream must manually be added to the Boot Image dialog now.
29 Jun 2015	2015_1.01	Update to 2015.1. Bitstream is no longer manual. Add support for PicoZed.
15 Jul 2015	2015_2.01	Update for 2015.2
06 Apr 2016	2015_4.01	Update to 2015.4. Add support for PZCC-FMC-V2.
01 Jun 2016	2015_4.02	Update to 2015.4. Clarified USB to Uart cable connection.
15 Sept 2016	2016_2.01	Updated to 2016.2
20 Jan 2017	2016_4.01	Updated to 2016.4
07 Jun 2017	2017_1.01	Updated to 2017.1 for MiniZed

Copyright © 2017 Avnet, Inc. AVNET, "Reach Further," and the AV logo are registered trademarks of Avnet, Inc. All other brands are the property of their respective owners.