Gen 3 Space Grade Serial Dual QSPI P-SRAM™ Development Kit for Xilinx User Guide

AK3XXG208XILCC

Table 1: Revision History

Revision No.	Date	History
1.0	11/22/2022	Initial Release
1.1	12/15/2022 1/17/2023 3/28/2023 05/22/2023	Updated Development Kit Ordering Info Added information on Direct Load and Boot Using Avalanche ID Updated Avalanche ID Removed Dev. Kit's Non-Socketed Ordering Option Added 96-Ball FBGA Socket Ordering Info

Contents

1. Overview

The Avalanche Gen 3 Space Grade Serial Dual QSPI P-SRAM™ Development Kit for Xilinx enables users to configure an Avalanche Serial Dual QSPI P-SRAM™ to be a bootable device for a Xilinx Versal ACAP VCK190 FPGA platform. Intended to enable rapid evaluation and prototyping, the Development Kit connects easily to the VCK190 platform via Samtec connectors.

Xilinx's Versal ACAP/FPGA platform requires a complex high density code configuration and storage memory architecture including multiple memory types such as Boot ROM, SRAM and Flash. Each of these memory technologies faces technical challenges in achievable densities, scalability, endurance/reliability and radiation resilience. Avalanche's 3rd Generation (Gen 3) Space Grade Serial Dual Quad Serial P-SRAM™ offers scalable density migration from 1Gb to 8Gb in a common package available today and is designed specifically to address these challenges. As such, this single device replaces ROM, SRAM, Flash and associated support circuitry and is ideal for radiation environments. Leveraging our 22nm high reliability pMTJ STT-MRAM, data is always non-volatile with 10^{16} write cycles endurance and greater than 20-year retention @ 85°C.

The Versal ACAP platform provides six different boot modes options: JTAG, Octal SPI, Quad SPI, SD, eMMC1 and SelectMAP. The boot modes are categorized into master or slave boot modes. For the purpose of this user guide, it will focus only on the Quad SPI (QSPI32 option, 4-Byte Address mode) master boot mode function. This Avalanche Development Kit is effectively a simple daughter board containing a custom socket for any of the Serial Dual Quad Serial P-SRAM densities (the MRAMs themselves are orderable separately), which leverage the 96-Ball FBGA device and connects to a Xilinx VCK190 evaluation board via Samtec high speed connector (240-pin, 8x20). Other boot options are outside of the scope of this user guide.

Writing to the Avalanche Serial Dual QSPI P-SRAM™ can be done through Linux U-Boot sf program or other means. This user guide will provide example and steps to follow on how to write to the device through Linux U-Boot.

2. Development Kit Ordering

Table 2: Development Kit Socketed Ordering

Part #	Description
AK30X208XILCCSOC	**Gen 3 Dual QSPI Plug in Board for Xilinx VCK190** – 96-Ball FBGA socketed daughter board (for MRAM*)

Note: * MRAM devices orderable separately

3. Ordering Options

3.1 Development Kit

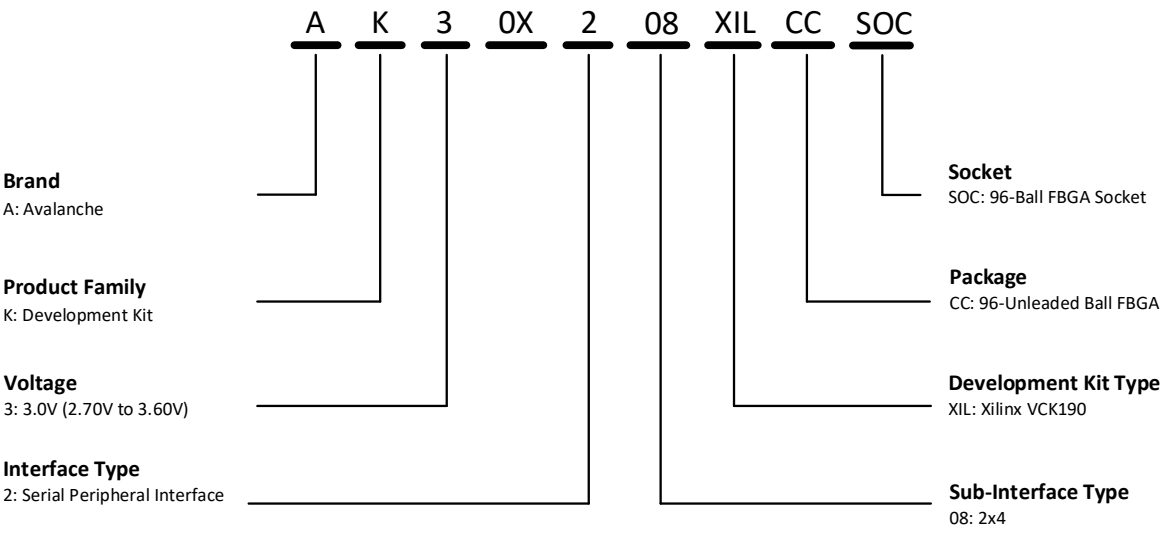


Figure 1: Development Kit Ordering Information

3.2 96-Ball FBGA Socket

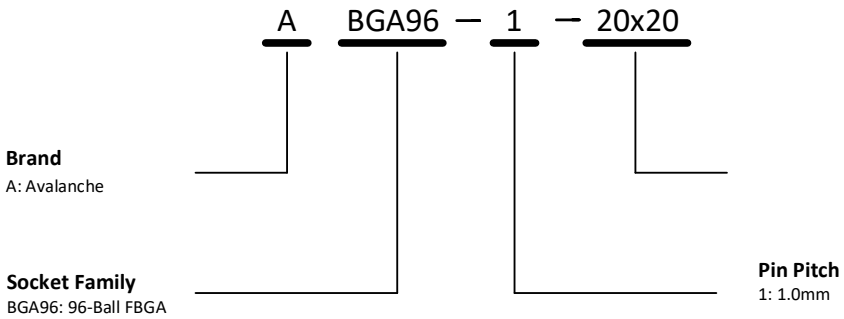


Figure 2: 96-Ball FBGA Socket Ordering Information

4. Development Kit Package Contents

1. An Avalanche daughter board (3 x 2.5 inches) with a 96-Ball FBGA socket
2. A 3.3V 1.8A AC/DC power supply cord

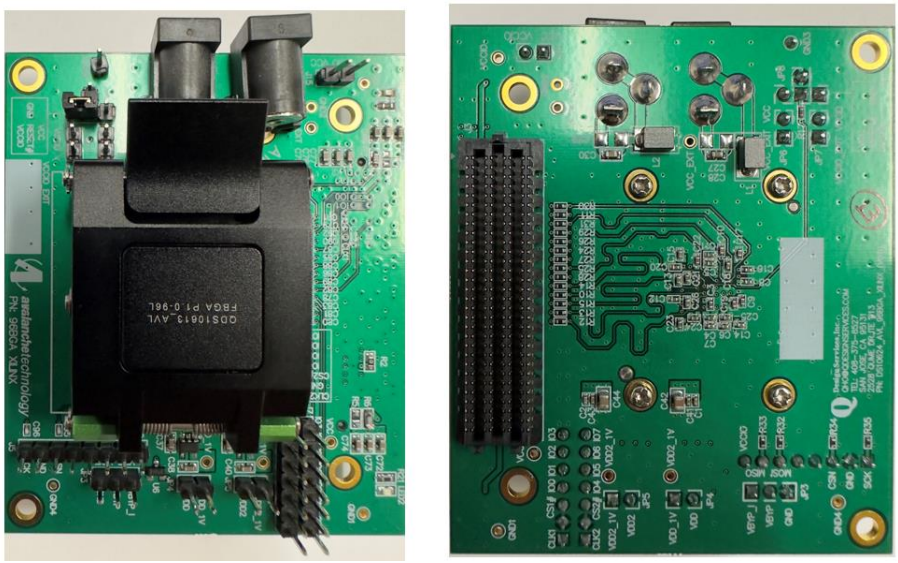


Figure 3: Serial Dual QSPI P-SRAM™ 96-Ball FPGA Socket Daughter Board (Front & Back)

Prerequisites

This user guide assumes the following prerequisite hardware, software and understanding:

- User is familiar with the Xilinx Vivado Design Suite & the Vitis software development platform
- Xilinx Vivado Design Suite has been installed
- Xilinx Vitis Software Platform has been installed
- Xilinx Vivado tools working environment is fully setup
- User has exported the hardware platform and .xsa image file is generated
- PetaLinux software tool has been installed
- User has created a new PetaLinux project or has an existing PetaLinux project

Requirements

- A PC system with one available USB 2.0/3.0 port
- Windows 10 or higher with 32/64-bit Operation System
- Linux OS to install PetaLinux tools
- Prerequires specified above
- Gen 3 Space Grade Serial Dual QSPI P-SRAM™ Development Kit for Xilinx:
    - Avalanche daughter board with a 96-Ball FBGA socket
    - 3.3V 1.8A AC/DC power supply cord

- Xilinx Versal ACAP VCK190 Development Kit

5. QSPI32 Boot Commands Supported by Avalanche Serial Dual QSPI P-SRAM™ AS3XXG208 and Xilinx Versal ACAP VCK190

Table 3 below shows read commands in both 4-Byte Address (QSPI32) and 3-Byte Address (QSPI24) supported by Xilinx RCU. Avalanche AS3XXG208 Dual Quad SPI P-SRAM device fully supports these commands. Refer to Avalanche 1Gb-8Gb Dual Quad SPI P-SRAM™ datasheet for more information. Click here to download datasheet.

Table 3: Quad SPI Read/Write Commands Supported by Xilinx RCU (ROM Code Unit)

Boot Mode	Data Bus Width	SPI Command	Command Op-Code	Address Byte	Latency/Dummy Cycles Required	Avalanche AS3XXG208 Command Support
QSPI32	1	Normal Read	13h	4	-	✓
QSPI32	1	Fast Read	0Ch	4	8	✓
QSPI32	8	Quad Output Fast Read (Dual Parallel Quad SPI)	6Ch	4	8	✓
QSPI32	1	Normal Write	02h	4	-	✓
QSPI24	1	Fast Read	0Bh	3	-	✓
QSPI24	8	Quad Output Fast Read (Dual Parallel Quad SPI)	6Bh	3	8	✓

Source: Versal ACAP Technical Reference Manual (AM011).

Table 4: Xilinx Versal ACAP VCK190 Supported Quad SPI Boot Mode

Configuration	MRAM Device Count	Chip Select (CS#) Count	Data Bus Width	Avalanche AS3XXG208 Configuration Support
Single (1-bit, 4-bit)	1	1	4	✓
Dual-Parallel (8-bit)	2	2	8	✓

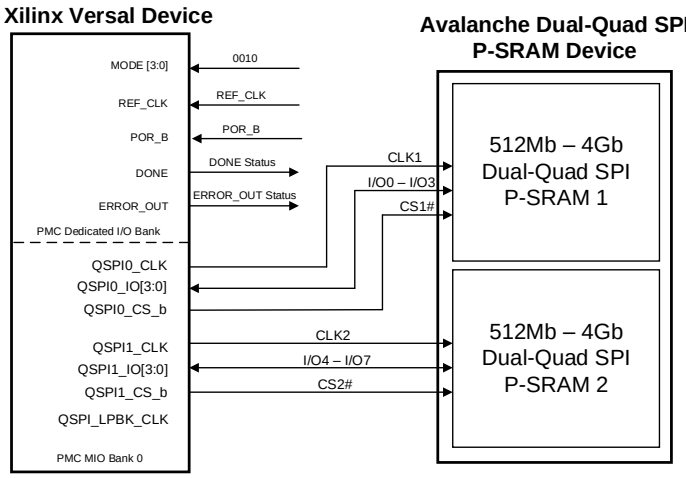


Figure 4: Dual-Parallel Quad SPI Interface Example

Note: For QSPI0_CLK > 37.5 MHz, QSPI_LPBK_CLK must be enabled in the design and unconnected on the board. Source: Versal ACAP Technical Reference Manual (AM011).

6. Development Kit Connection

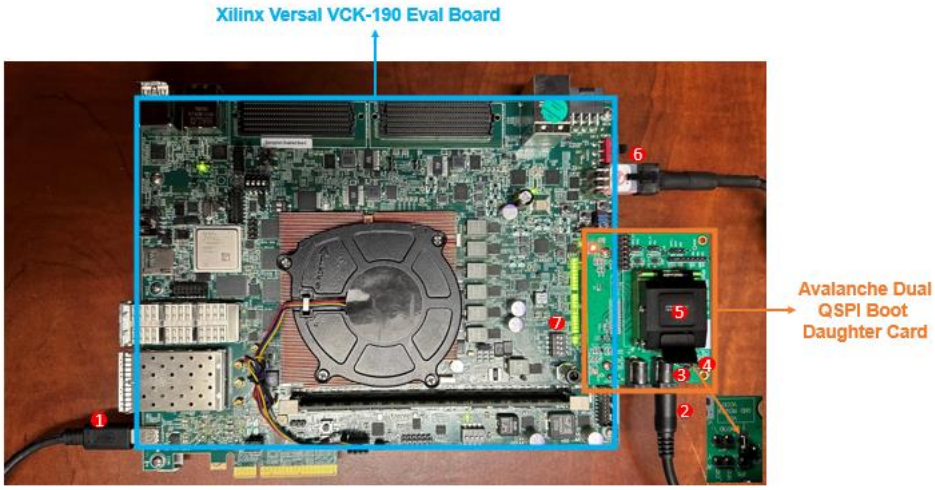


Figure 5: Xilinx Versal VC190 Connecting to Avalanche Serial Dual QSPI P-SRAM™ Daughter Board

Table 5: Development Kit Setup Description

Callout Number	Description
1	USB type C to PC host
2	3.3V 1.8A AC/DC power supply cord*
3	Power Barrel Connector Jack**
4	JP8 jumper setting

Callout Number	Description
5	Avalanche 96-Ball FBGA socket
6	VCK190 power connector
7	VCK190 Switch SW1 Settings

*Note: *Optional external power supply can be used to supply 3.3V 1.8A to the Avalanche Daughter Board*

*** Plug polarity on the Power Barrel Connector Jack: P = Center Positive*

Step 1: Set Configuration Switches on Xilinx Versal VCK190 for QSPI32 Boot Mode

The mode DIP switch SW1 on Xilinx Versal VCK190 must be set at ON, ON, OFF, ON positions as indicated in table 6 below.

Table 6: Xilinx Versal VCK190 Eval Board Quad SPI Boot Mode Settings

Boot Mode	Mode Pins [3:0]	Mode Switch SW1 [4:1]	Data Bus Width	Address Byte
QSPI32	0010	ON, ON, OFF, ON	1-bit, 4-bit	4

Source: Versal ACAP Technical Reference Manual (AM011)

Step 2: Power Sequence - Connect Power

1. Attach the Avalanche daughter board to the Versal VCK190 via Samtec connector
2. Connect USB cable between VCK190 and PC system
3. Connect power cable to VCK190's power connector and power outlet
4. Connect the 3.3V 1.8A power supply cord to power outlet
5. Turn on the VCK190 power using power switch SW13
6. Turn on Avalanche daughter board by connecting the 3.3V power connector to Daughter Board's connector jack
7. Hit the Reset button SW2 on the VCK190

Note: to ensure the Avalanche Serial Dual QSPI P-SRAM™ device to be powered up correctly, a correct order of step 5 and 6 must be followed.

Disconnect Power

1. Turn off Avalanche daughter board by disconnecting the 3.3V power connector to Daughter Board's connector jack
2. Turn off the VCK190 power using power switch SW13

Note: to ensure the Avalanche Serial Dual QSPI P-SRAM™ device to be powered down correctly, a correct order of step 1 and 2 must be followed.

Step 3: Re-building U-Boot to include Avalanche ID

User should download source code using external Kernel and U-Boot with PetaLinux. Click here to download the source code. For more information on how to repackage the U-Boot source, click here

The "SPI-NOR-IDs.c" file is located in "u-boot-xlnx/drivers/mtd/spi/spi-nor-ids.c. User is required to add Avalanche ID information to the file as indicated in the Avalanche Device ID sample code, Figure 5 below.

```
#ifdef CONFIG_SPI_FLASH_MT35XU
    { INFO("mt35xl512aba", 0x2c5a1a, 0,  128 * 1024,  512, USE_FSR | SPI_NOR_OCTAL_READ | SPI_NOR_4B_OPCODES | SPI_NOR_OCTAL_DTR_READ) },
    { INFO("mt35xu512aba", 0x2c5b1a, 0,  128 * 1024,  512, USE_FSR | SPI_NOR_OCTAL_READ | SPI_NOR_4B_OPCODES | SPI_NOR_OCTAL_DTR_READ) },
#endif /* CONFIG_SPI_FLASH_MT35XU */
    { INFO6("mt35xu01g",  0x2c5b1b, 0x104100, 128 * 1024,  1024, USE_FSR | SPI_NOR_OCTAL_READ | SPI_NOR_4B_OPCODES) },
    { INFO("mt35xu02g",  0x2c5b1c, 0, 128 * 1024,  2048, USE_FSR | SPI_NOR_OCTAL_READ | SPI_NOR_4B_OPCODES) },
#endif

#ifdef CONFIG_DUAL_QSPI_MRAM_AVALANCHE
    { INFO("AVALANCHE_512Mb",    0xE6212801, 0, 64 * 1024, 512, SECT_4K | USE_FSR | SPI_NOR_QUAD_READ | NO_CHIP_ERASE) },
    { INFO("AVALANCHE_1Gb",      0xE6212901, 0, 64 * 1024, 1024, SECT_4K | USE_FSR | SPI_NOR_QUAD_READ | NO_CHIP_ERASE) },
    { INFO("AVALANCHE_2Gb",      0xE6212a01, 0, 64 * 1024, 2048, SECT_4K | USE_FSR | SPI_NOR_QUAD_READ | NO_CHIP_ERASE) },
    { INFO("AVALANCHE_4Gb",      0xE6212c01, 0, 64 * 1024, 4096, SECT_4K | USE_FSR | SPI_NOR_QUAD_READ | NO_CHIP_ERASE) },
#endif

#ifdef CONFIG_SPI_FLASH_SPANSION    /* SPANSION */
    /* Spansion/Cypress -- single (large) sector size only, at least
     * for the chips listed here (without boot sectors).
     */
    { INFO("s25sl032p",  0x010215, 0x4d00,  64 * 1024,  64, SPI_NOR_DUAL_READ | SPI_NOR_QUAD_READ) },
    { INFO("s25sl064p",  0x010216, 0x4d00,  64 * 1024, 128, SPI_NOR_DUAL_READ | SPI_NOR_QUAD_READ) },
    { INFO("s25fl256s0", 0x010219, 0x4d00, 256 * 1024, 128, SPI_NOR_DUAL_READ | SPI_NOR_QUAD_READ | USE_CLSR) },
    { INFO("s25fl256s1", 0x010219, 0x4d01,  64 * 1024, 512, SPI_NOR_DUAL_READ | SPI_NOR_QUAD_READ | USE_CLSR) },
```

Figure 6: Adding Avalanche Device ID info to Linux SPI-NOR-IDS.c

Step 4: Rebuilding PetaLinux with updated U-Boot that includes Avalanche ID

The following Linux commands are used:

- Petaliux-build
- Petaliux-package --boot --u-boot –force

```
rlc2 10:10 /home/jun/opt/vck_linux_new2/images/linux % ll
total 697212
     4 drwxr-xr-x. 4 jun      4096 Jan 12 11:35 ./
     4 drwxr-xr-x. 3 jun      4096 Jan 11 18:19 ../
    68 -rw-r--r--. 1 jun     63432 Jan 12 10:17 bl31.bin
   540 -rw-r--r--. 1 jun    548704 Jan 12 10:17 bl31.elf
     4 -rw-rw-r--. 1 jun      3952 Jan 12 11:35 BOOT_bh.bin
  2792 -rw-rw-r--. 1 jun   2844896 Jan 12 11:35 BOOT.BIN
     4 -rw-rw-r--. 1 jun       634 Jan 12 11:35 bootgen.bif
     4 -rw-r--r--. 1 jun      2777 Jan 11 18:13 boot.scr
    12 -rw-r--r--. 1 jun      8474 Jan 12 10:08 config
 21580 -rw-r--r--. 1 jun  22000128 Jan 12 10:21 Image
  9372 -rw-r--r--. 1 jun   9550417 Jan 12 10:21 Image.gz
 33140 -rw-r--r--. 1 jun  33792796 Jan 12 10:21 image.ub
  2580 -rw-r--r--. 1 jun   2627252 Jan 12 10:51 plm.elf
  2580 -rwxr-xr-x. 1 jun   2626676 Jan 11 18:43 plm_linux_dual_200.elf*
  2580 -rwxrwxr-x. 1 jun   2627252 Jan 12 10:51 plm_linux_dual_200_new.elf*
  2304 -rw-r--r--. 1 jun   2346400 Jan 11 18:43 plm_old.elf
    32 -rw-r--r--. 1 jun     26896 Jan 12 10:19 pmc_cdo.bin
   208 -rw-r--r--. 1 jun    207748 Jan 11 18:17 psmfw.elf
     4 drwxr-xr-x. 2 jun      4096 Jan 11 18:13 pxelinux.cfg/
 28604 -rw-------. 1 jun 268435456 Jan 12 11:35 qemu_boot.img
 67612 -rw-r--r--. 1 jun  68955136 Jan 12 10:20 rootfs.cpio
 23744 -rw-r--r--. 1 jun  24209079 Jan 12 10:20 rootfs.cpio.gz
 23744 -rw-r--r--. 1 jun  24209143 Jan 12 10:20 rootfs.cpio.gz.u-boot
 97604 -rw-r--r--. 1 jun  99547136 Jan 12 10:20 rootfs.ext4
 29820 -rw-r--r--. 1 jun  30408704 Jan 12 10:21 rootfs.jffs2
    28 -rw-r--r--. 1 jun     23251 Jan 12 10:20 rootfs.manifest
 23868 -rw-r--r--. 1 jun  24335314 Jan 12 10:20 rootfs.tar.gz
    32 -rw-r--r--. 1 jun     27658 Jan 12 10:18 system.dtb
   944 -rwxr-xr-x. 1 jun    960176 Jan 12 10:18 u-boot.bin*
   972 -rw-r--r--. 1 jun    987834 Jan 12 10:18 u-boot-dtb.bin
  1040 -rw-r--r--. 1 jun   1053952 Jan 12 10:18 u-boot-dtb.elf
  7036 -rwxr-xr-x. 1 jun   7168360 Jan 12 10:18 u-boot.elf*
    36 -rw-r--r--. 1 jun     32657 Jan 11 18:13 versal-qemu-multiarch-pmc.dtb
    92 -rw-r--r--. 1 jun     89063 Jan 11 18:13 versal-qemu-multiarch-ps.dtb
    88 -rw-r--r--. 1 jun     85887 Jan 11 18:13 versal-qemu-ps.dtb
314132 -rw-r--r--. 1 jun 320401640 Jan 12 10:21 vmlinux
     4 drwxrwxr-x. 8 jun      4096 Jan 12 11:35 .Xil/
```

Figure 7: PetaLinux Boot Image Files

Step 5: Modifying Existing QSPI driver and PLM (Platform Loader and Manager) to add Avalanche ID to "Xloader_QSPI.C"

Assuming user has exported the hardware platform, the .xsa image file has been generated, and the PLM has been created, the Vitis software platform will create the PLM application project and edit_versal wrapper platform under the Explorer view as indicated in Figure 7 and Figure 8 below.

User is required to modify xloader_qspi.c to add Avalanche ID to it (refer to Figure 8 and Figure 9 below). The xloader_qspi.c is located in the sub-folder, Figure 7 below.

For more information on how to create the PLM, click here

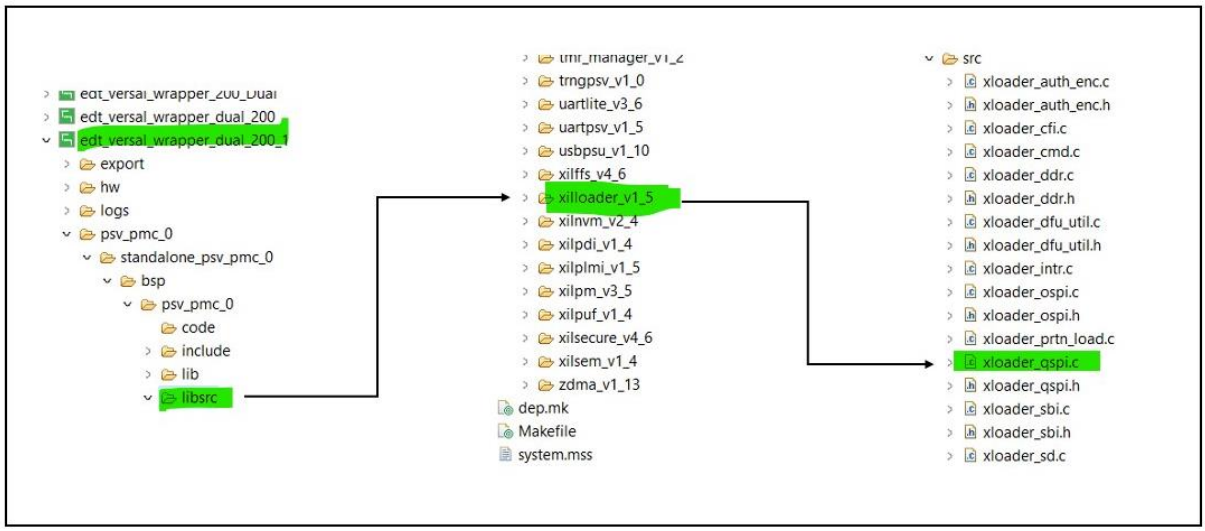


Figure 8: xloader_qspi.c file location

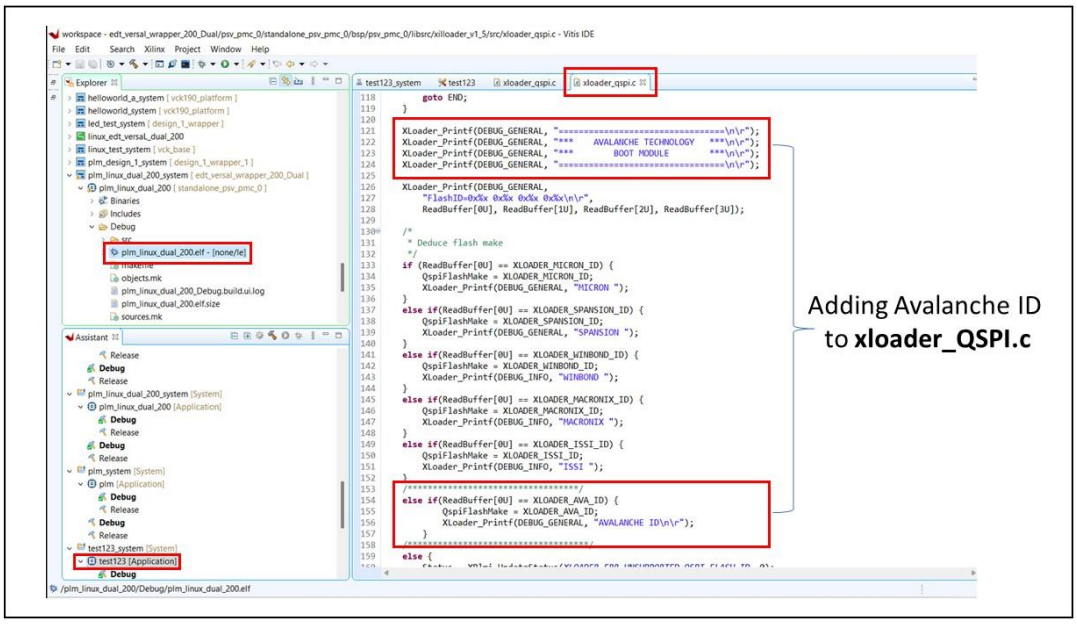


Figure 9: Adding Avalanche ID to Xloader_QSPI.c

```
171         * Deduce flash Size
172         */
173     /*****************************************************************/
174         if(ReadBuffer[0U] == XLOADER_AVA_ID) {
175             if (ReadBuffer[1U] == 0X21U) {
176                 XLoader_Printf(DEBUG_GENERAL, "HP Dual-Quad SPI - 3V\r\n");
177             }
178             else {
179     //          Status = XPlmi_UpdateStatus(XLOADER_ERR_UNSUPPORTED_QSPI_FLASH_ID, 0);
180                 XLoader_Printf(DEBUG_GENERAL, "Unsupported Interface or Voltage\r\n");
181     //          goto END;
182             }
183             if (ReadBuffer[2U] == 0X28U) {
184                 QspiFlashSize = XLOADER_FLASH_SIZE_512Mb;
185                 XLoader_Printf(DEBUG_GENERAL, "40C~125C, 512Mb\r\n");
186             }
187             else if (ReadBuffer[2U] == 0X29U) {
188                     QspiFlashSize = XLOADER_FLASH_SIZE_1G;
189                     XLoader_Printf(DEBUG_GENERAL, "40C~125C, 1Gb\r\n");
190                 }
191             else if (ReadBuffer[2U] == 0X2AU) {
192                     QspiFlashSize = XLOADER_FLASH_SIZE_2G;
193                     XLoader_Printf(DEBUG_GENERAL, "40C~125C, 2Gb\r\n");
194                 }
195             else if (ReadBuffer[2U] == 0X2CU) {
196                     QspiFlashSize = XLOADER_FLASH_SIZE_4G;
197                     XLoader_Printf(DEBUG_GENERAL, "40C~125C, 4Gb\r\n");
198                 }
199             else {
200     //          Status = XPlmi_UpdateStatus(XLOADER_ERR_UNSUPPORTED_QSPI_FLASH_SIZE, 0);
201             QspiFlashSize = XLOADER_FLASH_SIZE_1G;
202             XLoader_Printf(DEBUG_GENERAL, "Unsupported Density\r\n");
203     //      goto END;
```

Figure 10: Adding Avalanche ID Information to xloader_QSPI.c

A fully modified xloader_QSPI.c sample code that includes Avalanche ID and device propriety can be down-loaded from Avalanche website. Click here to download the sample code.

Step 6: Building booting Linux image with Modified PLM.elf

The PLM.elf file needs to be re-built and U-Boot source needs to be repackaged. The following commands are used:

- Cp plm_new.elf plm.elf
- Petaliux-package --boot --u-boot –force

Step 7: Configuring Boot Image Using Vivado Hardware Platform

To configure a bootable image, user can use Vivado hardware design to configure the Avalanche Serial Dual QSPI P-SRAM™ device. Use the following steps to generate the .xsa then use PetaLinux to create Linux boot images:

1. Select "Boot Mode"
2. Select "QSPI" as a Boot/Storage
3. Select "Dual Parallel" as QSPI Mode
4. Select "x4" as Data Mode
5. Enter "80" for Requested QSPI Reference Clock Frequency (MHz)

MRAM QSPI Interface Frequency (MHz) = QSPI Reference Clock Frequency (MHz)/8

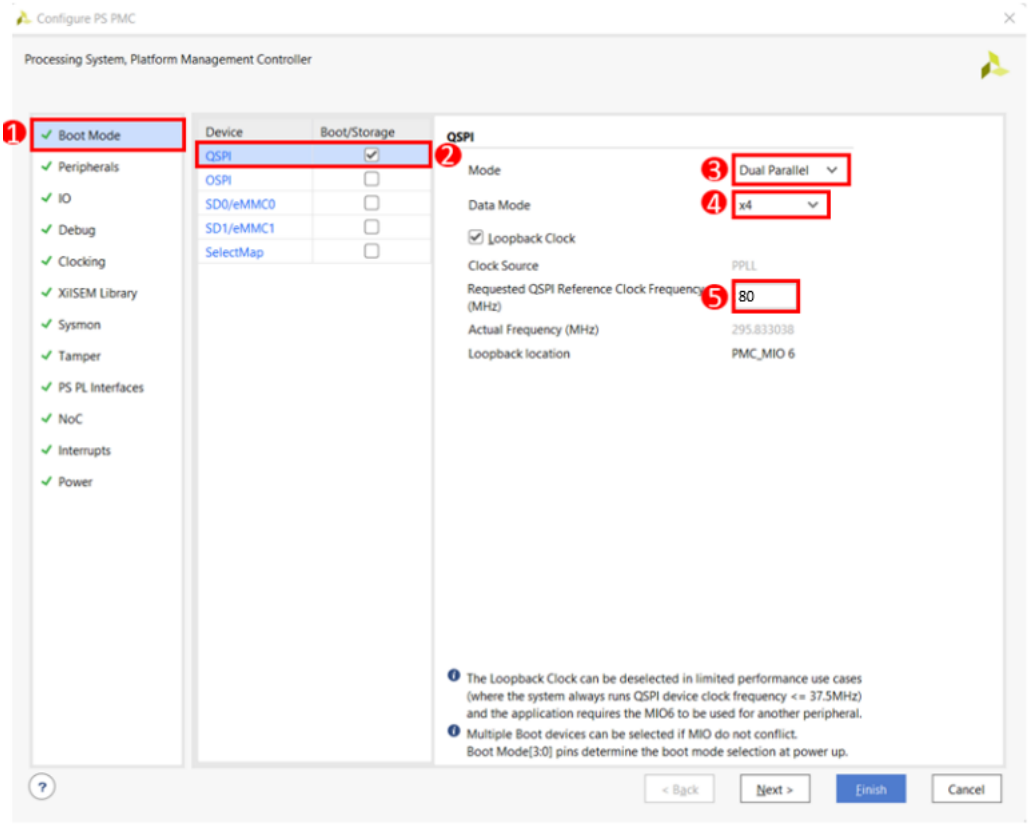


Figure 11: Steps to Configure Boot Images Using Vivado Hardware Platform

Step 8: Adding QSPI Device Tree Setting in "System-user.dtsi" during PetaLinux Build

To access the Avalanche Serial Dual QSPI P-SRAM up to 40MHz frequency, user is required to configure PetaLinux device tree file "system-user.dtsi". This config file is located at *<plnx-proj-root>/project-spec/meta-user/recipes-bsp/device-tree/files/*

Refer to Petalinux Tools Reference Guide for more information. Click here to download Petalinux Tools Reference Guide.

```
&qspi {
    status = "okay";
    spi-tx-bus-width = <0x1>;
    spi-rx-bus-width = <0x4>;
    flash0: flash@0 {
        spi-tx-bus-width = <0x1>;
        spi-rx-bus-width = <0x4>;
        spi-max-frequency = <40000000>;
    };
```

Figure 12: Including QSPI Device Tree Setting in system-user.dtsi

Step 9: Loading Images to DDR Memory and Writing PetaLinux Images to an Avalanche Serial Dual QSPI P-SRAM™ Device using JTAG and U-Boot Commands

Loading PetaLinux Images to DDR Memory

1. Create the Tcl script using this PetaLinux command from the Versal project directory:
   $petalinux-boot --jtag --kernel --tcl versal.tcl
   Note: the versal.tcl file includes commands to select appropriate targets and download application files to required locations in the DDR memory. Below is a sample versal.tcl file

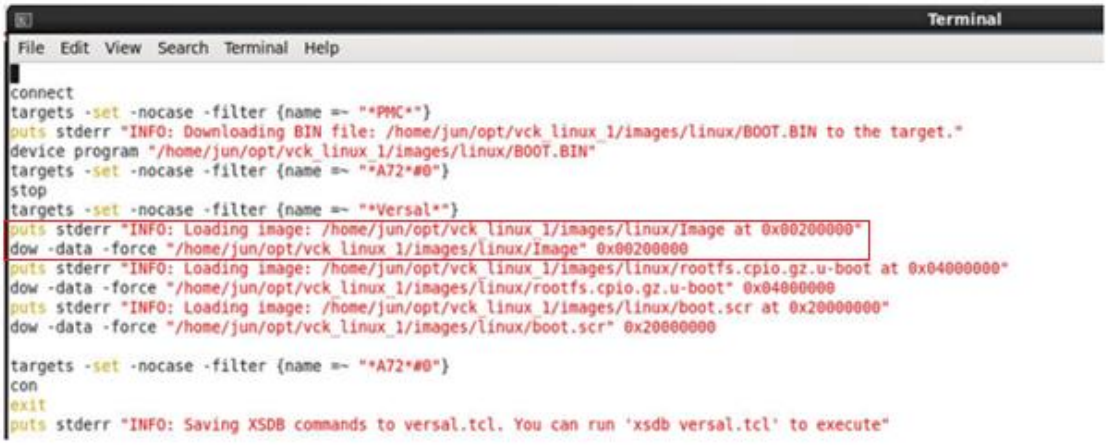


Figure 13: Sample versal.tcl file

2. Modify the generated versal.tcl to include the two commands with parameters: puts stdeer, and down -data -force as shown in the sample versal.tcl file above.
3. Set the boot mode switch SW1 to ON-ON-ON-ON JTAG boot mode.
4. Configure the Tera Term serial application with default serial settings 115200, N8 and open the Tera Term console.
5. In the XSCT console, connect to the target over JTAG using this connect command: xsct% connect
   The connect command returns the channel ID of the connection.
6. Run the following target command to list the available targets and select a target using its ID: xsct% targets.
7. Download the versal.tcl file which will load the following files and images: BOOT.BIN, Image, rootfs.cpio.gz.u-boot, boot.scr on the DDR memory of the VCK190 board using the following commands:
   - xsct% targets 1
   - xsct% rst
   - xsct > source versal.tcl

Writing PetaLinux Images from DDR Memory to the Serial Dual QSPI Device

8. During U-Boot sequence, when the message “hit any key to stop” displays on the Terminal, hit any key to go to Linux prompt.
9. From Linux prompt, use this command to write data from DDR memory to the Serial Dual QSPI Daughter Board: sf write <address> <offset> $filesize
10. Follow **Step 2: Power Sequence – Disconnect Power** to disconnect power to both the Avalanche daughter board and the VCK190. Set the boot mode switch SW1 to ON-ON-OFF-ON QSPI32 boot mode as indicated in table 6.

Step 10: Booting Images from an Avalanche Serial Dual QSPI P-SRAM™ Device

Follow **Step 2: Power Sequence – Connect Power** to power cycle both the Avalanche daughter board and the VCK190 to complete Linux boot from Avalanche Serial Dual QSPI device.

8. Avalanche Serial Dual QSPI P-SRAM™ Daughter Board

The Avalanche Serial Dual QSPI P-SRAM™ daughter board (part # AK30X208XILCCSOC) is an 8-layer board.

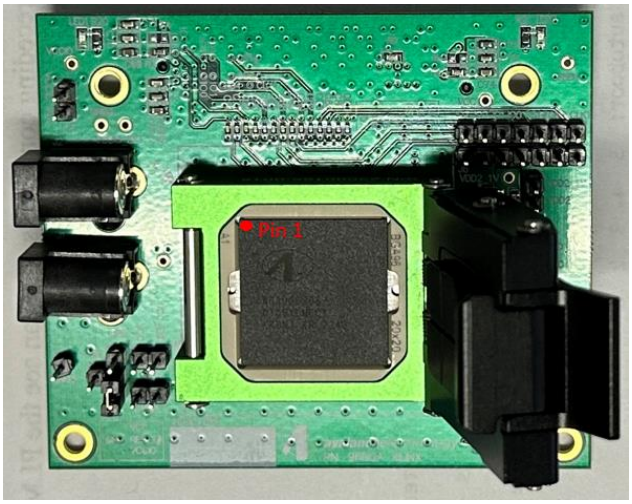
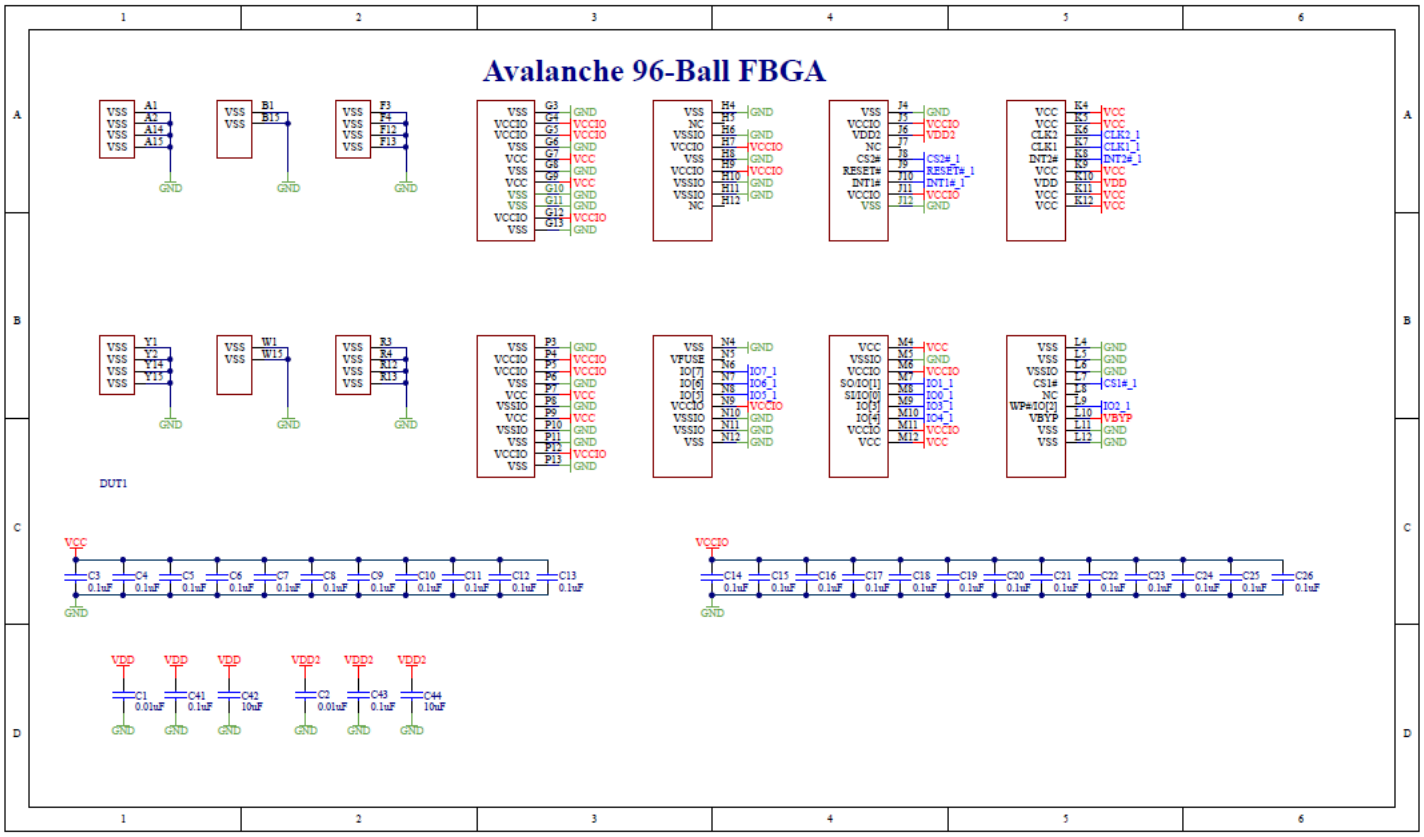


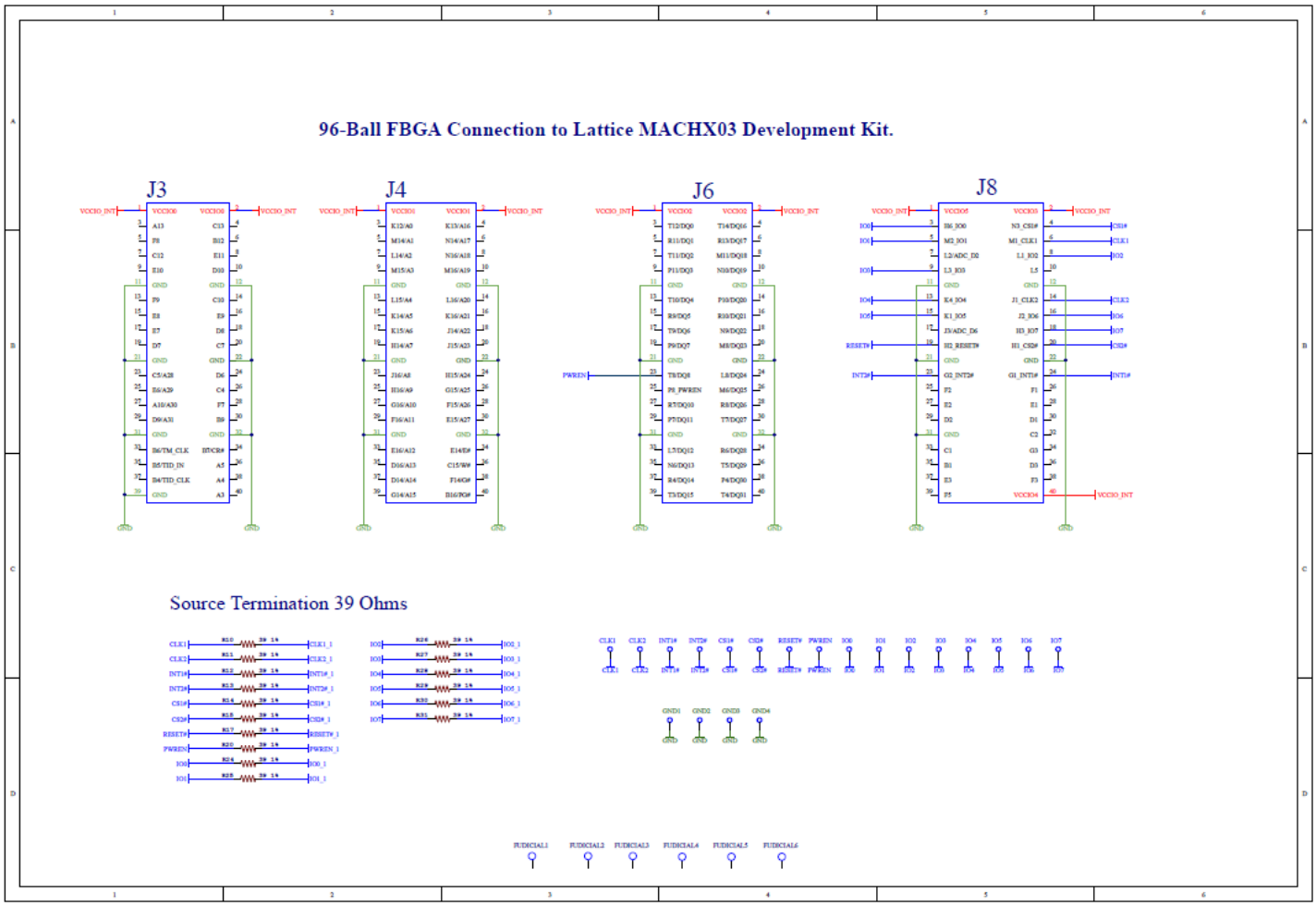
Figure 14: Avalanche Serial Dual QSPI P-SRAM™ Daughter Board with AS30XG208 Device

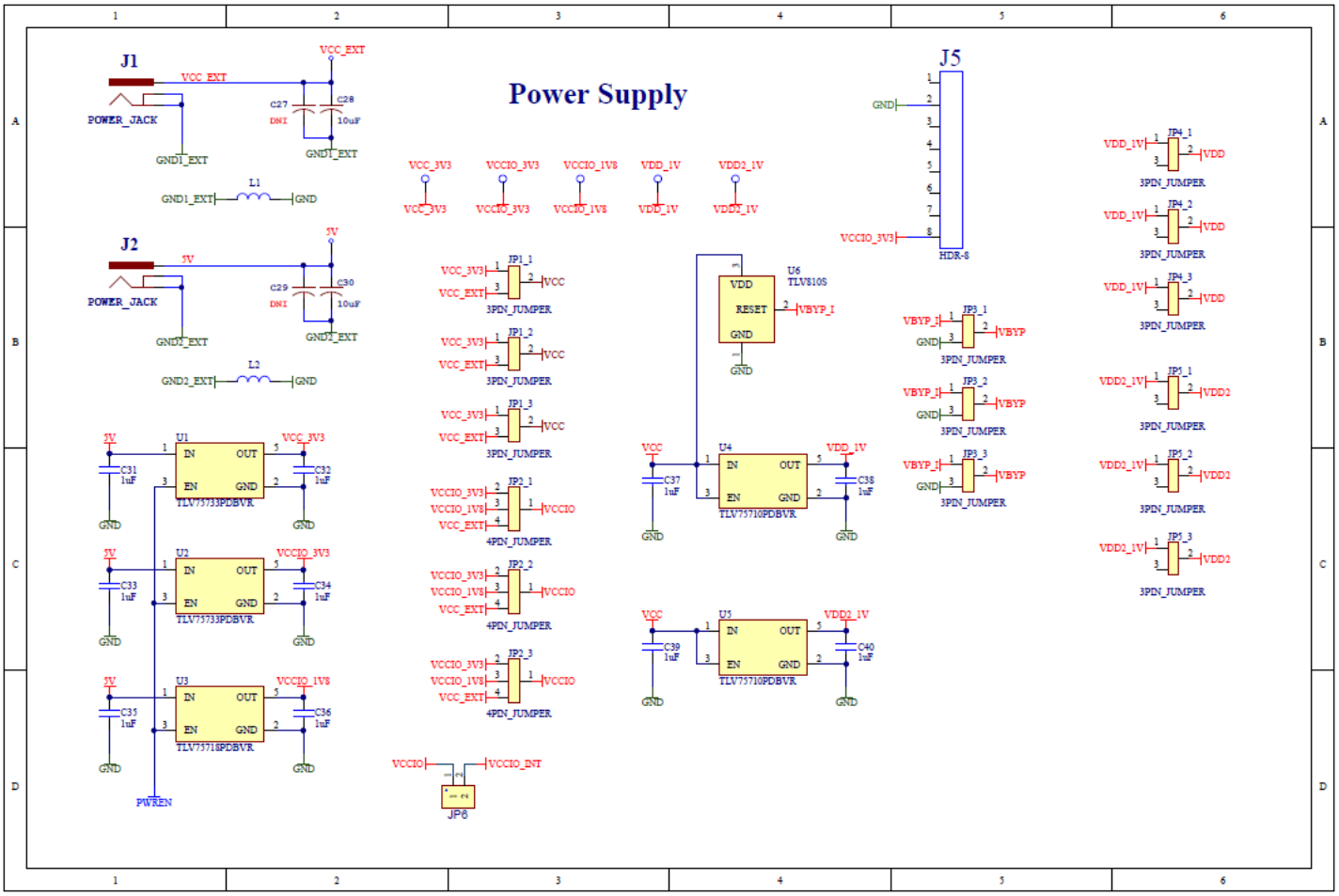
9. Schematic and BOMs

Click here to download gerber file, board layout and schematic in high resolution ALTIUM format.

Schematic




96-Ball FBGA Connection to Lattice MACHX03 Development Kit.
J3
J4
J6
J8
Source Termination 39 Ohms


Power Supply
J1
J2
J5
POWER_JACK
VCC_EXT
5V
GND1_EXT
GND2_EXT
L1
L2
C27
C28
C29
C30
10uF
DNI
U1
U2
U3
U4
U5
U6
TLV75733PDBVR
TLV75718PDBVR
TLV75710PDBVR
TLV810S
IN
OUT
EN
GND
RESET
VBYP_I
PWREN
VCC_3V3
VCCIO_3V3
VCCIO_1V8
VDD_1V
VDD2_1V
VCCIO
VCCIO_INT
VCC
VBYP
VDD
VDD2
JP1_1
JP1_2
JP1_3
JP2_1
JP2_2
JP2_3
JP3_1
JP3_2
JP3_3
JP4_1
JP4_2
JP4_3
JP5_1
JP5_2
JP5_3
JP6
3PIN_JUMPER
4PIN_JUMPER
HDR-8
C31
C32
C33
C34
C35
C36
C37
C38
C39
C40
1uF

BOMs

Item	Quantity	Description	Designator	Footprint
1	2	0.1µF	C1, C2	N_0402
2	26	0.1µF	C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C41, C43	N_0402
3	2	DNI	C27, C29	N_1206
4	5	10µF	C28, C30, C69, C72, C75	N_0603
5	10	1µF	C37, C38, C39, C40, C70, C71, C73, C74, C76, C77	N_0603
6	2	10µF	C42, C44	N_0805
7	18	5pF	C78, C79, C80, C81, C82, C83, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96	N_0402
8	1	96-Ball FBGA footprint/socket	DUT1	Ava 96-Ball FBGA
9	8	TP	GND1, GND2, GND3, GND4, Vcc, Vccio, Vdd2_1V, Vdd_1V	Test Point
10	2	Power Jack	J1, J2	Power Jack
11	1	Samtec Connector (SEAM-30-02.0-L-08-1-A-K-TR)	J3	Samtec
12	1	HDR-8	J5	HDR 1x8
13	2	HDR-8	J6, J7	HDR 1x7
14	1	HDR-2-Pin	J15	HDR-2
15	1	3-Pin Jumper	JP3	3-Pin Jumper
16	4	HDR-TH 2P-P2.54	JP4, JP5, Jp6, JP7	1x2 Header with Shunt 051914
17	1	4-Pin Jumper	JP8	4-Pin Jumper
18	2	FB	L1, L2	FB
19	2	Red LED	LED1, LED2	N_0603
20	3	0R	R2, R3, R6	N_0603
21	2	DNI	R4	N_0603
22	19	39.1% Resistor	R10, R11, R12, R13, R14, R15, R17, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R20	N_0402
23	1	666 Resistor	R20	N_0603
24	1	261 Resistor	R21	N_0603
25	2	TI TLV75710PDBVR	U4, U5	SOT23-5
26	1	TI TLV810S	U6	SOT23-3
27	2	TP	Vcc_ext, Vccio_ext	Test Point