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 05/19/2023	Updated Development Kit Ordering Info Added information on Direct Load and Boot Using Avalanche ID Updated Avalanche ID Removed Eval Kit's Non-Socketed Ordering Option Added 96-Ball FBGA Socket Ordering Info
1.2	05/21/2024	Upgraded Board Hardware to rev. 0.9 from rev.0.8
1.3	08/20/2024	Updated Image Loading to DDR Sequence

Contents

1. Overview

The Avalanche Serial Dual QSPI P-SRAM™ for Xilinx Versal ACAP VCK190 Development Kit 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 card 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. Avalanche P-SRAM™ Product Support & Development Kit Ordering Options

The Avalanche P-SRAM™ daughterboard with a socket can be populated with either one of the following Serial Dual QSPI P-SRAM™ devices:

Table 2: Avalanche Serial Dual QSPI P-SRAM™ Product Support & Development Kit Socketed Ordering Options

Device Part #	Density	Voltage	Organization	Package	Development Kit Socketed Ordering Options
AS301G208-0108X0MCC	1Gb	2.7 V to 3.6V	Dual QSPI	96-Ball FBGA	AK301G208XILCC
AS302G208-0108X0MCC	2Gb	2.7 V to 3.6V	Dual QSPI	96-Ball FBGA	AK302G208XILCC

Device Part #	Density	Voltage	Organization	Package	Development Kit Socketed Ordering Options
AS304G208-0108X0MCC	4Gb	2.7 V to 3.6V	Dual QSPI	96-Ball FBGA	AK304G208XILCC
AS308G208-0108X0MCC	8Gb	2.7 V to 3.6V	Dual QSPI	96-Ball FBGA	AK308G208XILCC

3. Ordering Options

3.1 Development Kit

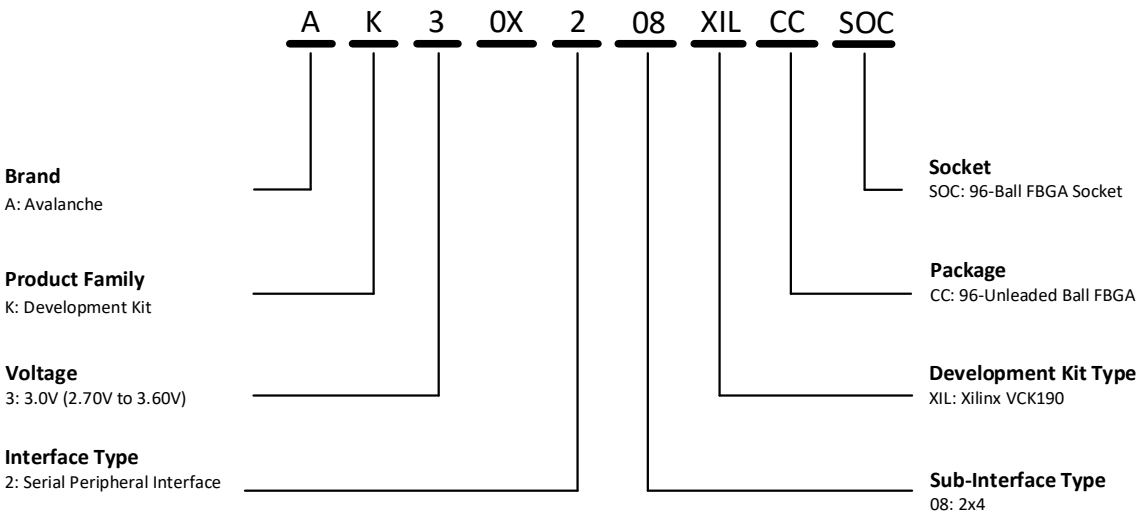


Figure 1: Development Kit Ordering Information

3.2 96-Ball FBGA Socket

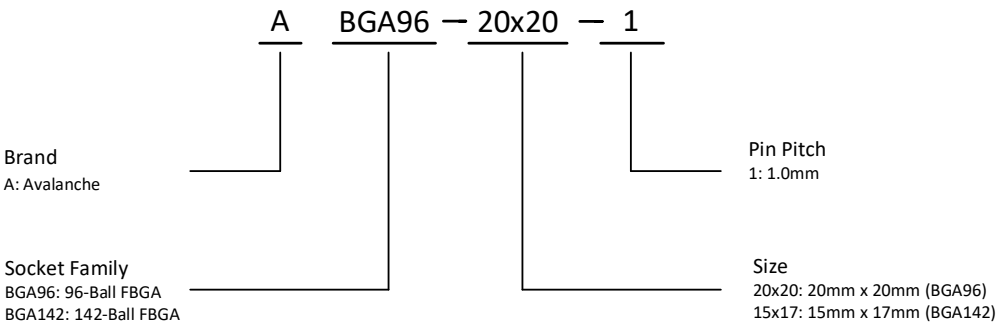


Figure 2: 96-Ball FBGA Socket Ordering Information

4. Development Kit Package Contents

1. An Avalanche daughterboard (3 x 2.5 inches)
2. A 96-FBGA socket

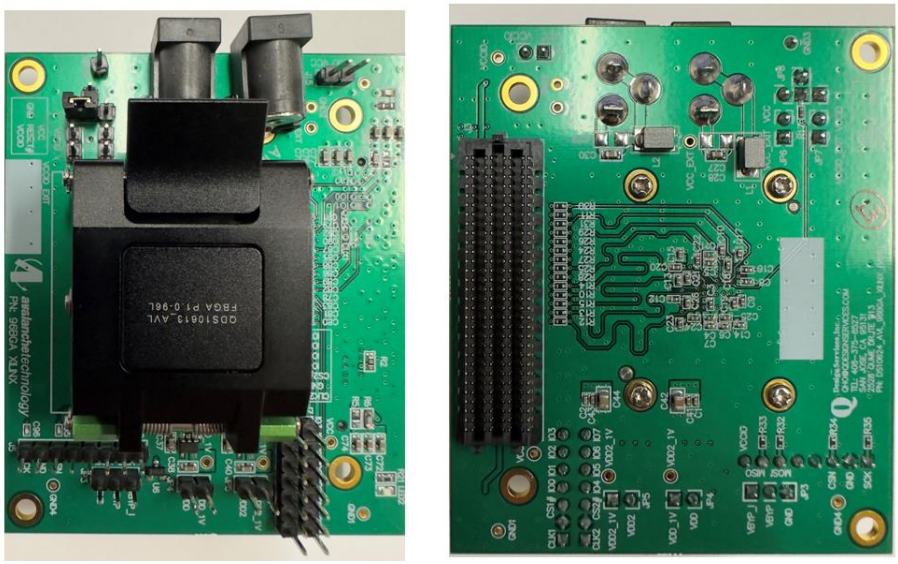


Figure 3: Serial Dual QSPI P-SRAM™ 96-Ball FPGA Socket Daughter Card (Front & Back)

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
- Serial Dual QSPI P-SRAM™ for Xilinx Versal ACAP VCK 190 Development Kit:
  - Avalanche daughterboard with a 96-FBGA socket
- 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	✓

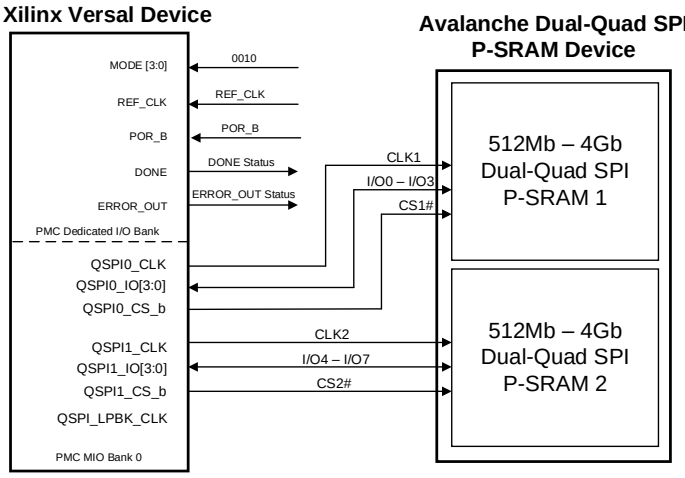


Figure 4: Dual-Parallel Quad SPI Interface Example

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

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 5 below.

Table 5: 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 card 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. Turn on the VCK190 power using power switch SW13
5. Hit the Reset button SW2 on the VCK190

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 5: 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

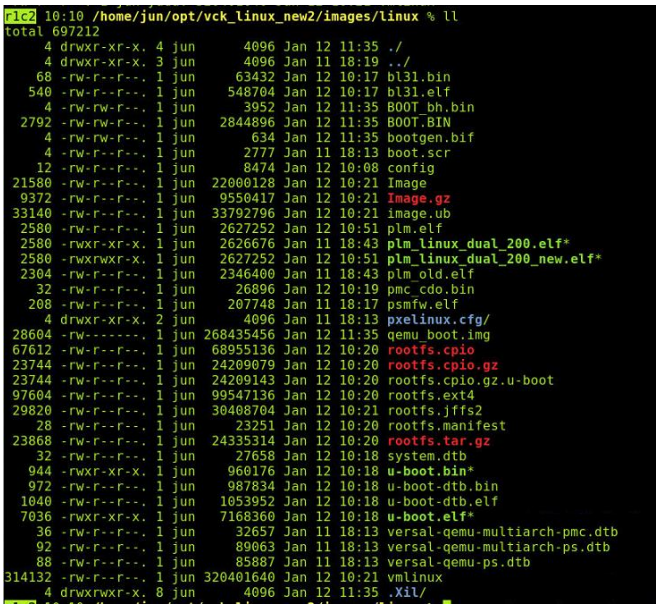


Figure 6: PetaLinux Boot Image Files

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

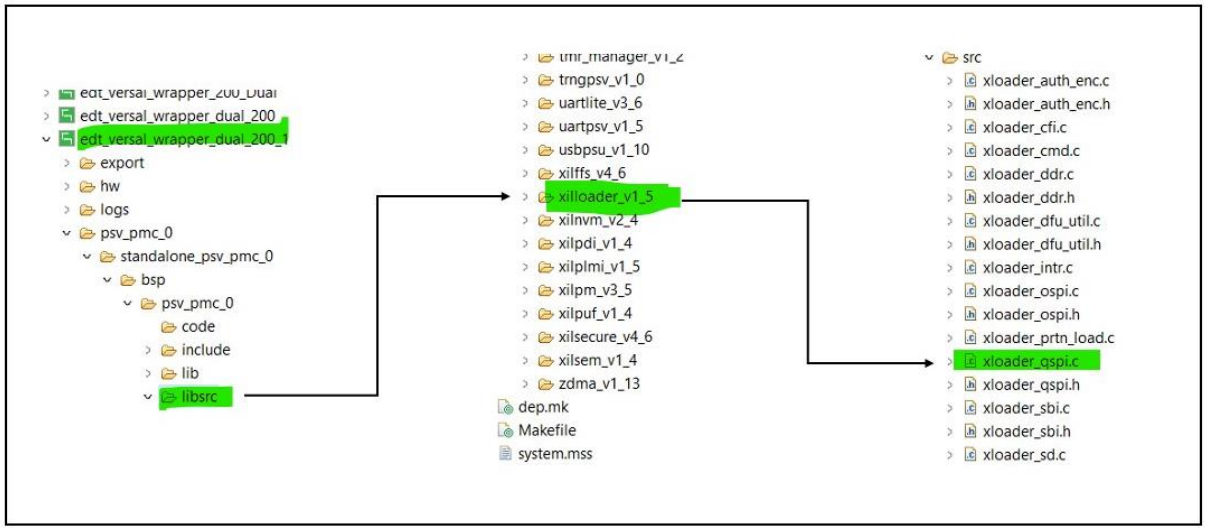


Figure 7: xloader_qspi.c file location

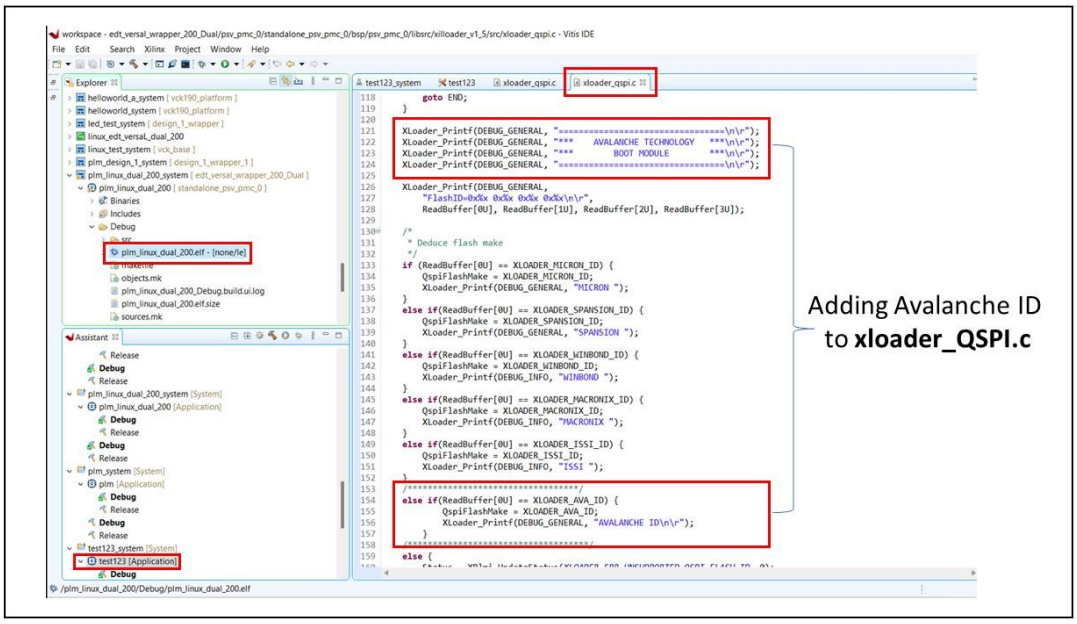


Figure 8: Adding Avalanche ID to Xloader_QSPI.c

```
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 9: 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 downloaded 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™ daughter card. 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

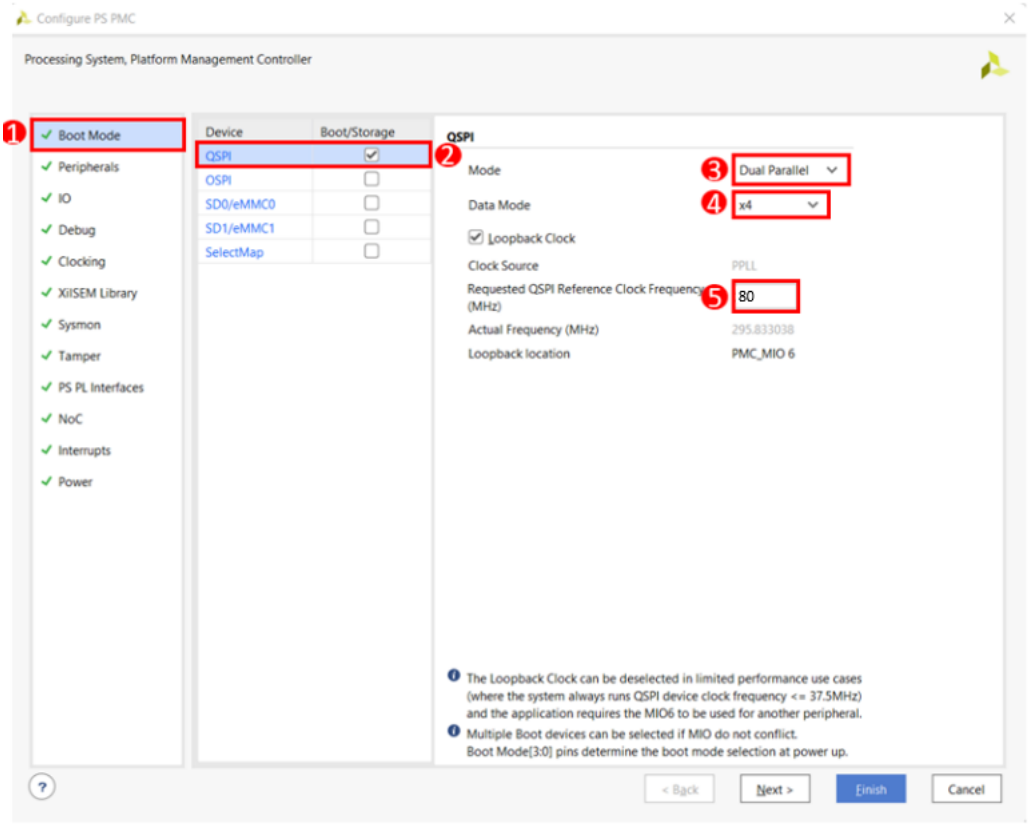


Figure 10: Steps to Configure Boot Images Using Vivado Hardware Platform

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 11: 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™ Daughter Card on Versal Board using JTAG and U-Boot Commands

Step 9.1: Loading and running u-boot Image to DDR Memory

- Set the boot mode pins to JTAG [SW1 to ON-ON-ON-ON] on Hardware, power on and connect to board.
- Launch xsdb and connect to Hardware.
  - xsdb% connect
- Once the Hardware is connected, you will see the following targets when "ta" command is issued on xsdb terminal.

1. Versal xcvc1902
2. RPU (PS POR is active)
3. Cortex-R5 #0 (PS POR is active)
4. Cortex-R5 #1 (PS POR is active)
5. APU (FPD domain isolation)
6. Cortex-A72 #0 (FPD domain isolation)
7. Cortex-A72 #1 (FPD domain isolation)
8. PPU
9. MicroBlaze PPU (Sleeping after reset)
10. PSM

11. PMC
12. PL
13. DPC

- Set the target for programming:
  - xsdb% targets 1
- Issue the following command to program the Hardware in JTAG boot mode.
  - xsdb% device program BOOT.BIN
  - xsdb% RST

Reset system and load the u-boot. The u-boot prompt will be displayed.

Step 9.2: Writing U-boot Image to an Avalanche Serial Dual QSPI P-SRAM™

- Execute the following command in U-Boot on serial console to probe the flash device
  - Versal> sf probe 0 0 0
- On xsdb terminal, execute the following commands to write QSPI boot image to DDR
  - xsdb% targets -set -nocase -filter {name =~ "Versal *"}
  - xsdb% dow -force -data <BOOT.BIN> 0x100000
- On the serial console, execute the following commands in U-Boot to write the boot image to QSPI Flash
  - Versal> sf write 0x100000 0 <size>
- Disconnect the power, change the boot mode to QSPI [SW1 to ON-ON-OFF-ON], then reconnect the power.

Reset system and load the u-boot from QSPI flash and u-boot prompt will be displayed.

Step 9.3: Writing PetaLinux Images to an Avalanche Serial Dual QSPI P-SRAM™

- Execute the following command in U-Boot on serial console to probe the flash device
  - Versal> sf probe 0 0 0
- On xsdb terminal, execute the following commands
  - xsdb% dow -force -data <image.ub> 0x200000
- Then on serial console, execute the following commands in U-Boot to write the image to QSPI Flash
  - Versal> sf write 0x200000 0x00f40000 <size>
- On xsdb terminal, execute the following commands
  - xsdb% dow -force -data <boot.scr> 0x20000000
- Then on serial console, execute the following commands in U-Boot to write the image to QSPI Flash
  - Versal> sf write 0x20000000 0x3E80000 <size>
- Disconnect power.

Step 9.4: Booting Images from an Avalanche Serial Dual QSPI P-SRAM™ Daughter Card

Connect Power to power cycle both the Avalanche daughter card and the VCK190 to complete Linux boot from Avalanche Serial Dual QSPI device. System resets and load the u-boot and then the Linux image from the QSPI flash and linux prompt will be displayed

7. Avalanche Serial Dual QSPI P-SRAM™ Daughter Card

The Avalanche Serial Dual QSPI P-SRAM™ daughter card (part # AK3XXG208XILCCSOC) is an 8-layer board.

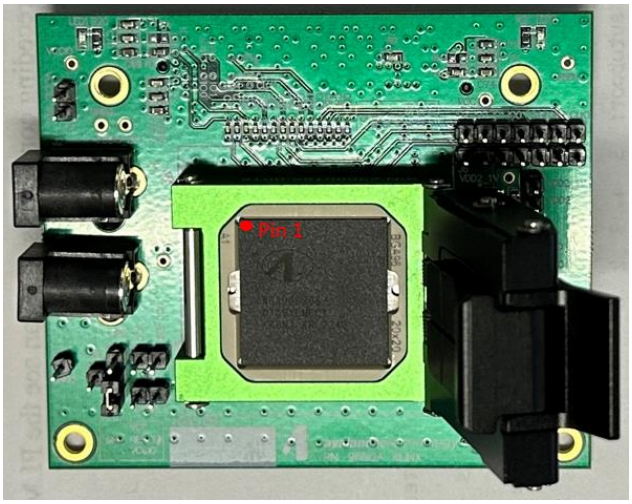
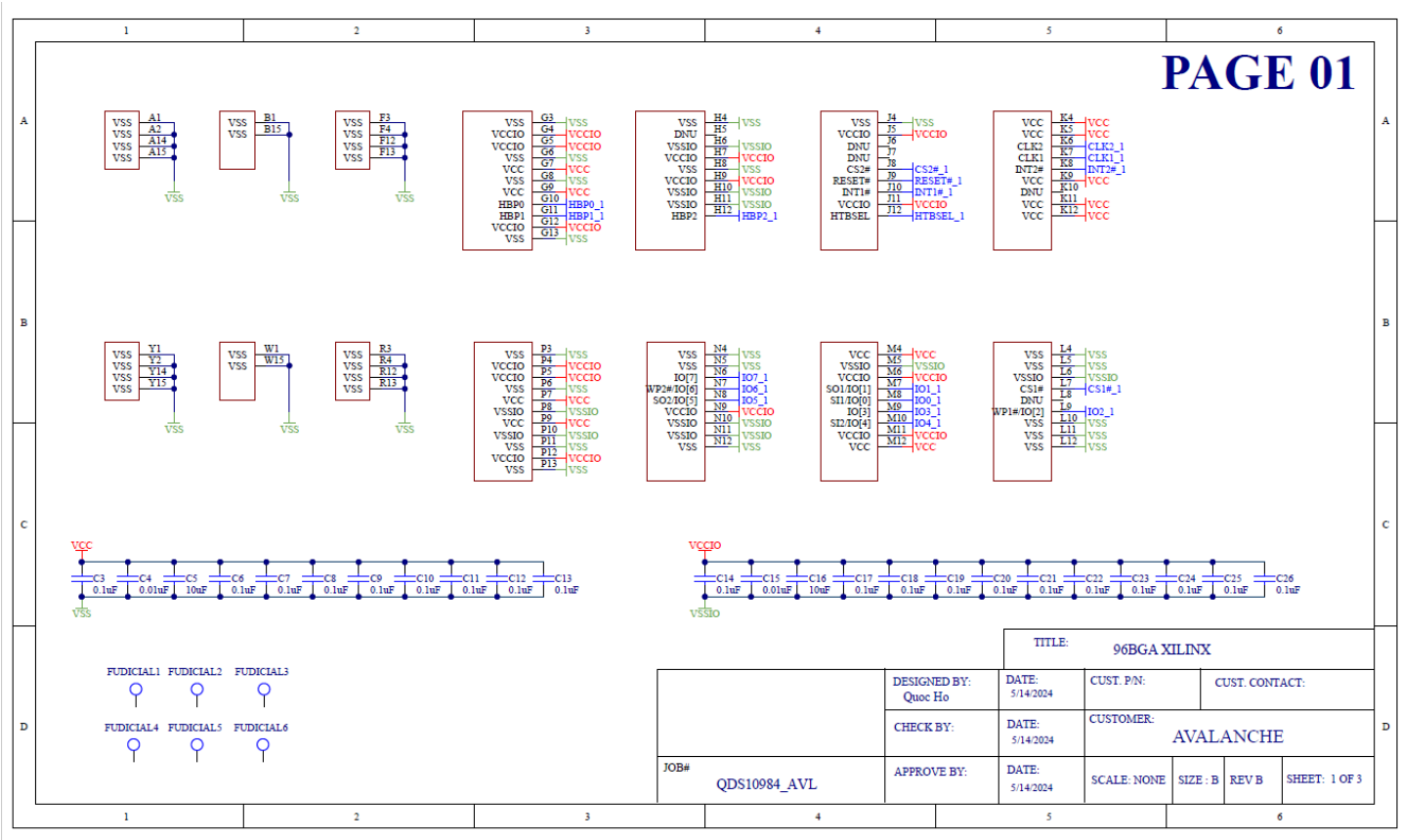


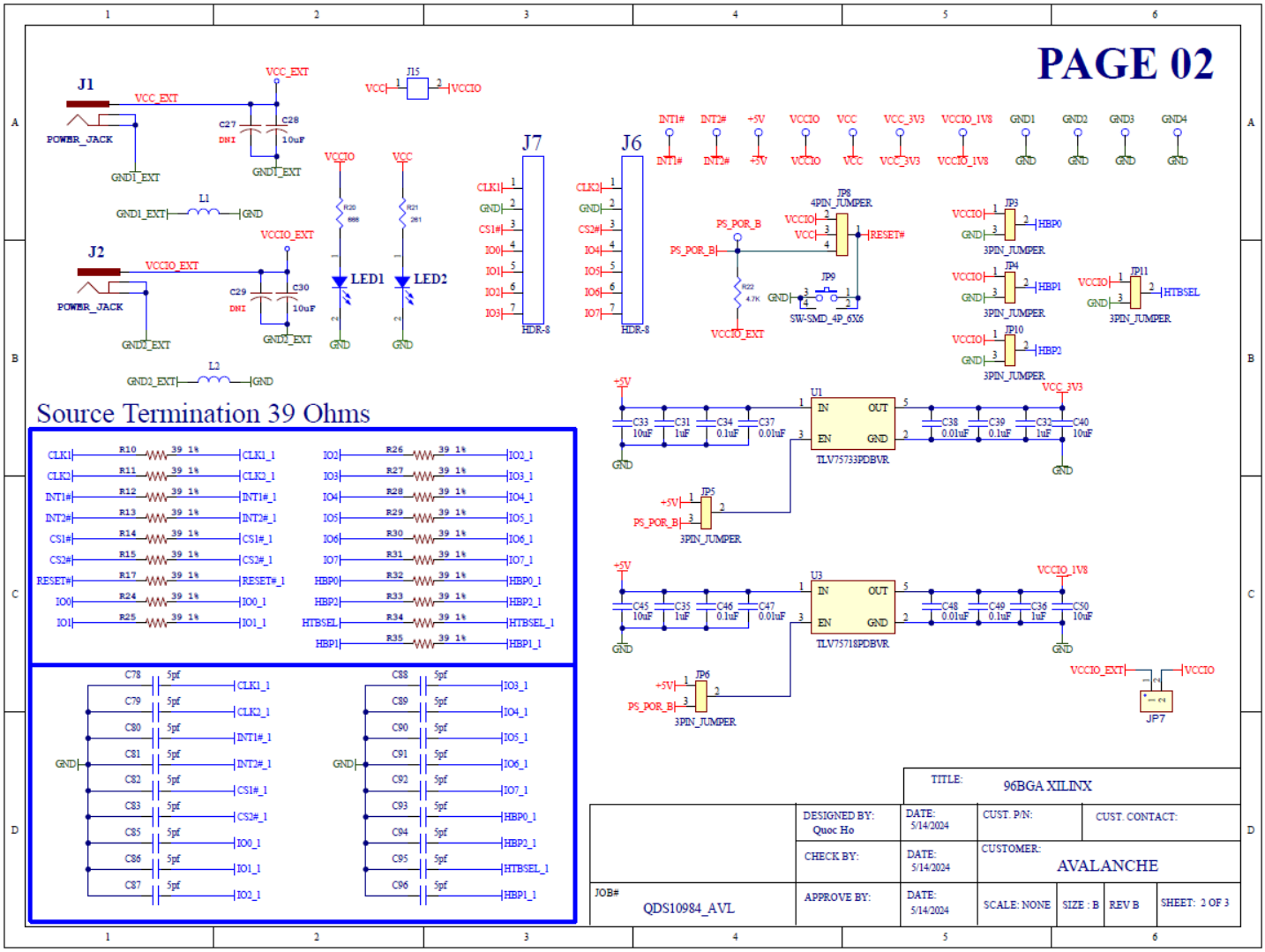
Figure 12: Avalanche Serial Dual QSPI P-SRAM™ Daughter Card with AS30XG208 Device

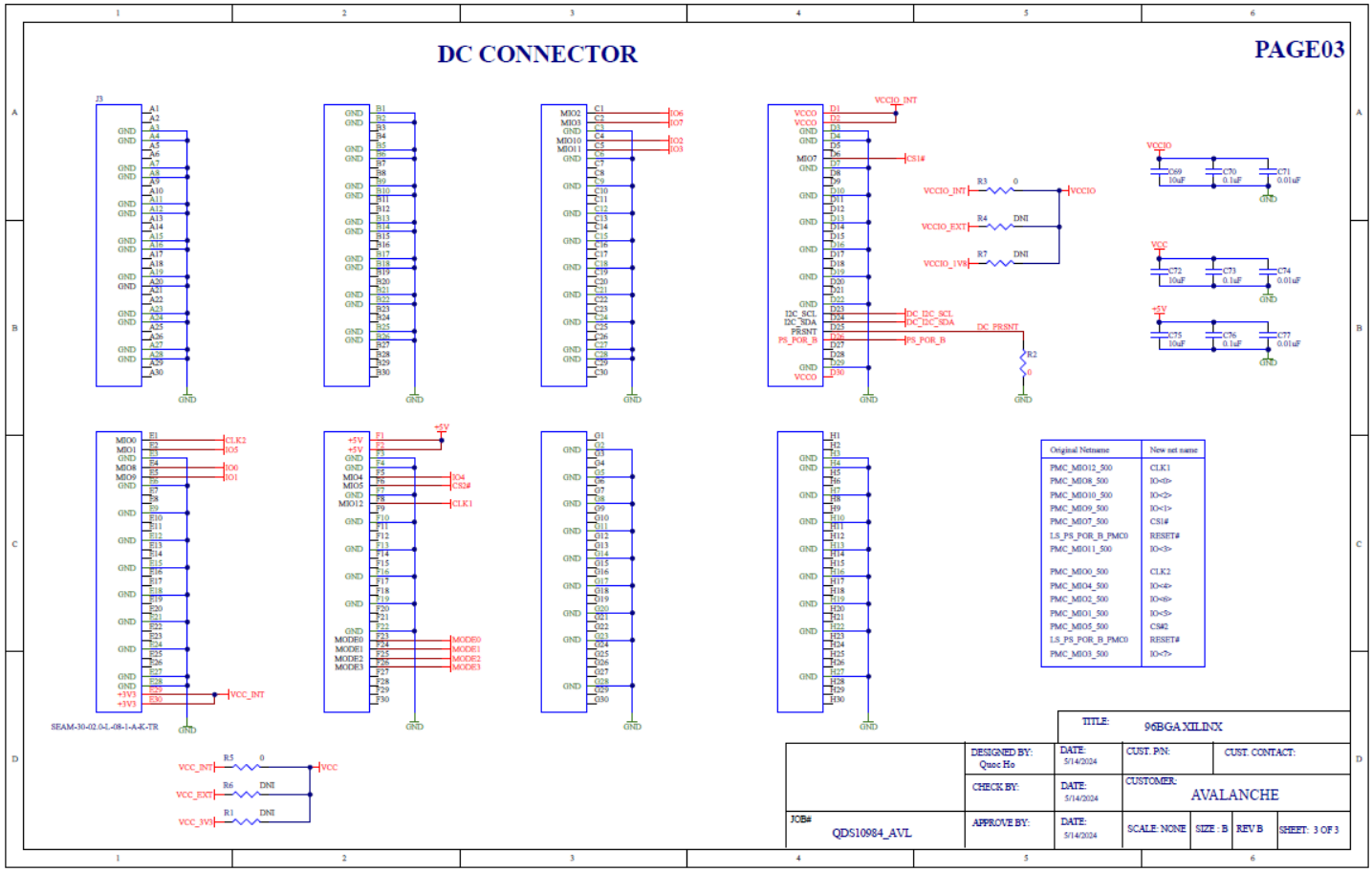
8. Schematic and BOMs

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

Schematic




PAGE 02
Source Termination 39 Ohms
TITLE: 96BGA XILINX
DESIGNED BY: Quoc Ho
DATE: 5/14/2024
CUST. P/N:
CUST. CONTACT:
CHECK BY:
DATE: 5/14/2024
CUSTOMER: AVALANCHE
JOB# QDS10984_AVL
APPROVE BY:
DATE: 5/14/2024
SCALE: NONE
SIZE : B
REV B
SHEET: 2 OF 3


DC CONNECTOR
PAGE03
SEAM-30-02.0-L-08-1-A-K-TR
VCC_INT
VCCIO_INT
VCCIO
VCC
+5V
CLK2
CLK1
CS2#
MODE0
MODE1
MODE2
MODE3
DC_I2C_SCL
DC_I2C_SDA
DC_PRSNT
PS_POR_B
R1 DNI
R2 0
R3 0
R4 DNI
R5 0
R6 DNI
R7 DNI
Original Netname
New net name
PMC_MIO12_500
CLK1
PMC_MIO8_500
IO<0>
PMC_MIO10_500
IO<2>
PMC_MIO9_500
IO<1>
PMC_MIO7_500
CS1#
LS_PS_POR_B_PMC0
RESET#
PMC_MIO11_500
IO<3>
PMC_MIO0_500
CLK2
PMC_MIO4_500
IO<4>
PMC_MIO2_500
IO<6>
PMC_MIO1_500
IO<5>
PMC_MIO5_500
CS#2
LS_PS_POR_B_PMC0
RESET#
PMC_MIO3_500
IO<7>
TITLE:
96BGA XILINX
DESIGNED BY:
Quoc Ho
DATE:
5/14/2024
CUST. P/N:
CUST. CONTACT:
CHECK BY:
DATE:
5/14/2024
CUSTOMER:
AVALANCHE
JOB#
QDS10984_AVL
APPROVE BY:
DATE:
5/14/2024
SCALE: NONE
SIZE : B
REV B
SHEET: 3 OF 3

BOMs

Item	Quantity	Value/Comment	Designator	Footprint
1	12	TP	+5V, GND1, GND2, GND3, GND4, INT1#, INT2#, PS_POR_B, Vcc, Vcc_3V3, Vccio, Vccio_1V8	Test Point
2	20	0.1µF	C3, C6, C7, C8, C9, C10, C11, C12, C13, C14, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26	N_0402
3	2	0.01µF	C4, C15	N_0402
4	2	10µF	C5, C16	N_0402
5	2	DNI	C27, C29	N_1206
6	9	10µF	C28, C30, C33, C40, C45, C50, C69, C72, C75	N_0603
7	4	1µF	C31, C32, C35, C36	N_0603
8	7	0.1µF	C34, C39, C46, C49, C70, C73, C76	N_0603
9	7	0.01µF	C37, C38, C47, C48, C71, C74, C77	N_0603
10	18	5pF	C78, C79, C80, C81, C82, C83, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96	N_0402
11	1	88-BGA	DU1	BGA88
12	6	Fudicial	Fudicial1, Fudicial2, Fudicial3, Fudicial4, Fudicial5, Fudicial6	Fudicial
13	2	Power_Jack	J1, J2	Power Jack
14	1	SEAM-30-02.0-L-08-1-1-K-TR	J3	Samtec Connector
15	2	HDR-8	J6, J7	HDR1X7
16	1	HDR-2Pin	J15	HDR-2
17	6	3Pin_Jumper	JP3, JP4, JP5, JP6, JP10, JP11	Jumper_3Pin
18	1	HDR-TH_2P-P2.54	JP7	1x2 Header with Shunt_051914
19	1	4Pin_Jumper	JP8	Jumper_4Pin
20	1	SW-SMD_4P_6X6	JP9	SW-SMD_4P_6X6
21	2	2743019447	L1, L2	FB_2743019447
22	2	RED	LED1, LED2	'N_0603
23	4	DNI	R1, R4, R6, R7	N_0603
24	3	0	R2, R3, R5	N_0603
25	19	39	R10, R11, R12, R13, R14, R15, R17, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35	N_0603
26	1	666	R20	N_0603
27	1	261	R21	N_0603
28	1	4.7K	R22	N_0603
29	1	TLV75733PDBVR	U1	SOT23-5
30	1	TLV75718PDBVR	U3	SOT23-5
31	2	TP	Vcc_Ext, Vccio_Ext	Test Point