Power Sequencing with Fault Logging Demo using ASC Bridge Board with MachXO3-9400 Development Board and ASC Breakout Board

User Guide

FPGA-UG-02079 Version 1.0

November 2018

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
ASC	Analog Sense and Control
ASC-I/F	ASC 3-wire 8 MHz serial Interface (wdat, wrclk, and rdat)
BGA	Ball Grid Array
DIP	Dual In-line Package
EFB	Embedded Function Block
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input and Output
GUI	Graphic User Interface
HVOUT	High Voltage Output (12 V Open Drain or Charge Pump)
IC	Integrated Circuit
I^2C	Inter Integrated Circuit (serial communication bus)
IMON	Current Monitor
L-ASC10	Hardware Expander ASC IC (48-pin QFN package)
LED	Light Emitting Dummy
LPTM21	Platform Manager 2 IC (237-Ball BGA package includes ASC)
LPTM21L	Platform Manager 2/Hardware Expander IC (100-Ball BGA package includes ASC)
LUT	Look Up Table
OPN	Ordering Part Number
PIO	Programmable Input and Output
POT	Potentiometer (variable resistor with wiper output)
QFN	Quad-Flat No-Leads
rdat	Read Data (from ASC device)
RTL	Register Transfer Logic
SPI	Serial Peripheral Interface
TMON	Temperature Monitor
UFM	User Flash Memory
USB	Universal Serial Bus
VMON	Voltage Monitor
wdat	Write Data (to ASC device)
wrclk	Write Clock (to ASC device)

1. Introduction

This demo shows how the MachXO3™ FPGA can be used as a Central Controller in a distributed Hardware Management system with four separate Power Planes. Each Power Plane is controlled and monitored by a separate ASC device. All of the ASC devices are monitored and controlled by a single design (the Central Control) in the MachXO3. Some of the features of this demo include the following:

- Sequential Power Plane sequencing of four Power Planes (debug mode)
- Parallel Power Plane sequencing of four Power Planes (normal mode)
- Sequencing 31 pseudo DC/DC enables and/or reset signals
- Emergency and normal shutdown sequences
- Mandatory and optional ASC expander devices
- Re-sequencing after a shutdown of entire system or just one Power Plane
- Fault log enable after the Power Plane sequence is complete
- Fault log with 32-bit timestamp (one second time base)
- Fault log disable while reading and clearing the fault records
- Fault log with user configurable 32 bits

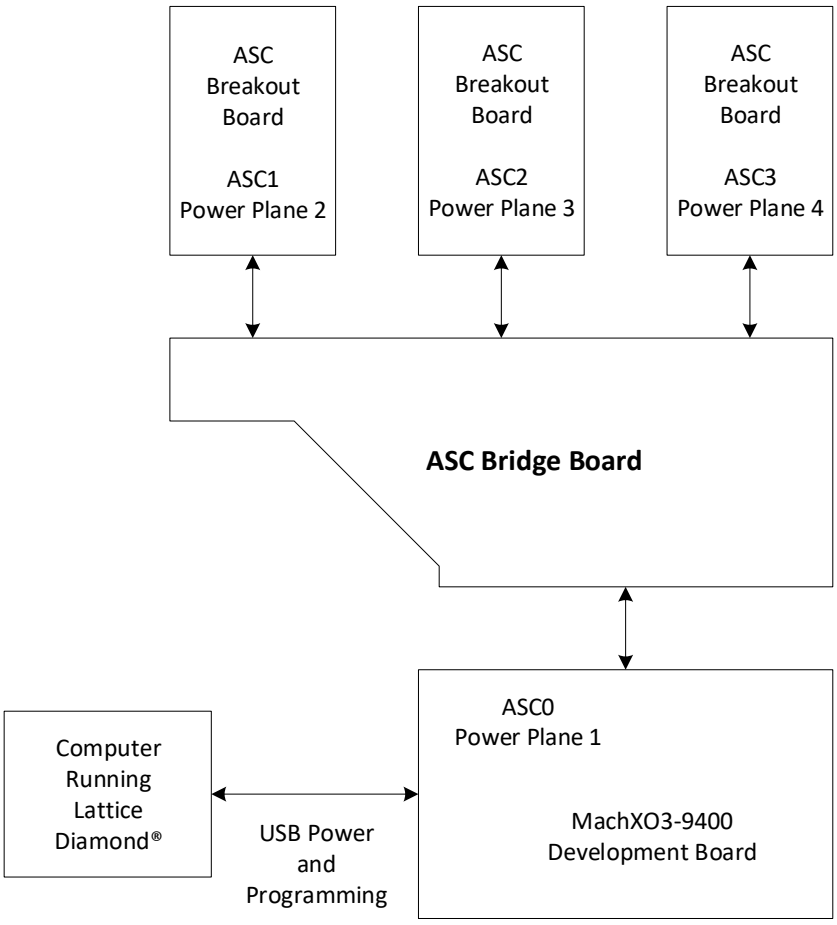


Figure 1.1. Block Diagram of Power Sequencing with Fault Logging Demo

2. Demo Requirements

2.1. Hardware Requirements

Some assembly is required. Nylon standoffs and screws are included with the ASC Bridge Board.

- MachXO3-9400 Development Board (OPN: LCMXO3LF-9400C-ASC-B-EVN)
- ASC Bridge Board (OPN: L-ASC-BRIDGE-EVN)
- Three ASC Breakout Boards (OPN: LPTM-ASC-B-EVN)
- Small screwdriver (for adjusting POT on the MachXO3-9400 Development Board)
- Oscilloscope with a standard 10-Meg probe (for measuring the Fault Recording time)

2.2. Software Requirements

- Lattice Diamond Design Software (includes Platform Designer), version 3.10 or later - www.latticesemi.com/diamond
- Lattice Diamond Programmer Tool - http://www.latticesemi.com/programmer

3. Setting up the Demo

Figure 3.1 identifies the key resources for the demo and shows the fully assembled system.

To set up the demo:

1. Install two standoffs on each ASC Breakout Board as shown.
2. Install four standoffs on the ASC Bridge Board as shown.
3. Connect the ASC Bridge Board to the MachXO3-9400 Development Board as shown.
   a. Connect ASC Bridge Board J3 to MaxhXO3 X2.
   b. Connect ASC Bridge Board J2 to MachXO3 X3.
4. Install J11 jumper on each ASC Breakout Board.
5. Set the *Reset* jumper J12 to Optional on each ASC Breakout Board.
6. Slide all the POTs on each ASC Breakout Board to the *0 V* position.
7. Rotate POT1 on the MachXO3-9400 Development Board to the *0 V* position (fully clockwise).
8. Make sure that SW3 on the ASC1 Breakout Board only has the *1* position down; all others are up.
9. Make sure that SW3 on the ASC2 Breakout Board only has the *2* position down; all others are up.
10. Make sure that SW3 on the ASC3 Breakout Board only has the *3* position down; all others are up.
11. Set SW3 on the ASC Bridge Board to the *XO3* position.
12. Connect the ASC1 Breakout Board to the *ASC0* location using J6 as shown.
13. Connect the ASC2 Breakout Board to the *ASC1* location using J13 as shown.
14. Connect the ASC3 Breakout Board to the *ASC2* location using J7 as shown.
15. Make sure that all of the levers of SW1 on the MachXO3 Development Board are in the up position
16. Connect a USB cable to J11 on the MachXO3-9400 Development Board (located at the bottom) and to a computer with Diamond and the demo installed.

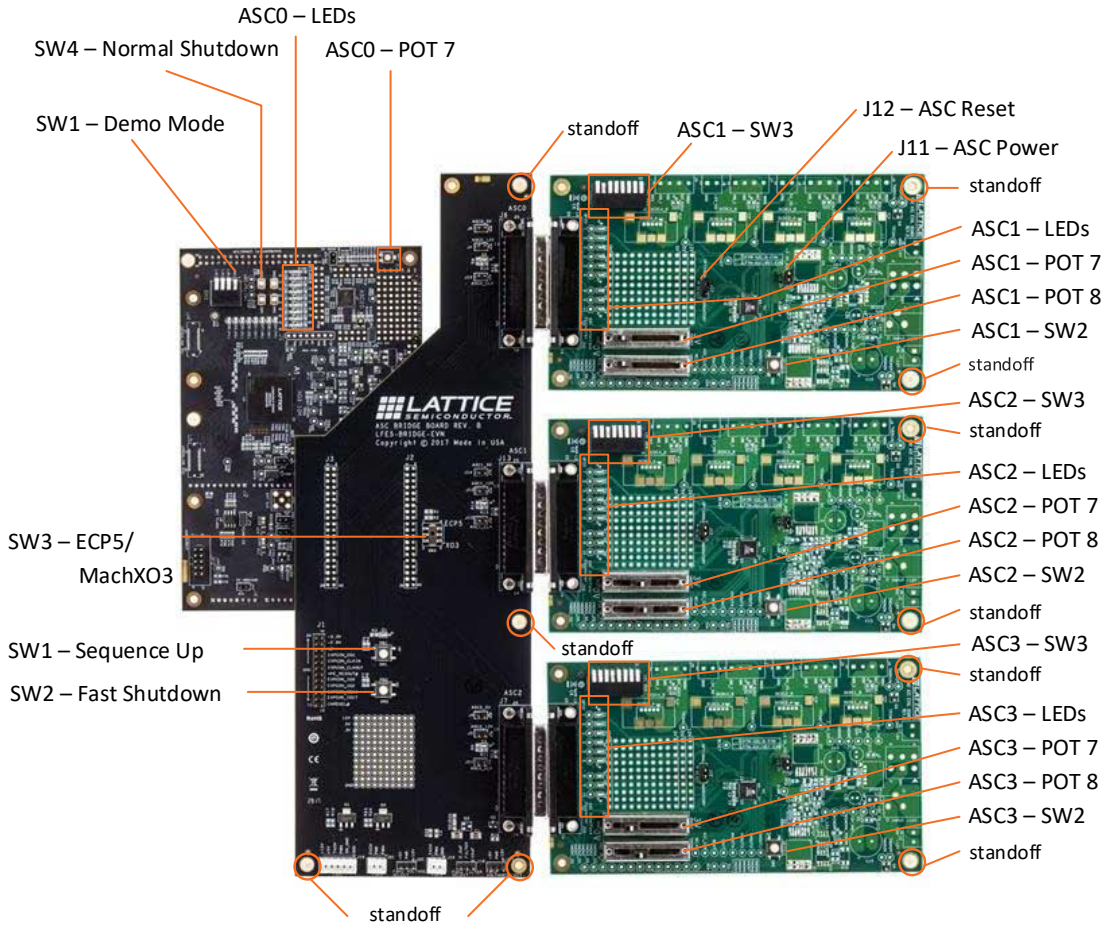


Figure 3.1. Resources for the Sequencing with Fault Logging Demo

4. Demo Package Components and Directory Structure

Table 4.1. Demo Package Components

Folder	Filename	Description
Top		
	MachXO3_ASC_Bridge_Demo_V0.zip	Archive of demo – downloaded from website
	MachXO3_ASC_Bridge_Demo.ldf	Diamond Project
	MachXO3_ASC_Bridge_Demo.lpf	Diamond Preference
Top\impl1		
	MachXO3_ASC_Bridge_Demo_impl1.jed	MachXO3 Programming
	MachXO3_ASC_Bridge_Demo.ptm	Platform Designer Project
	MachXO3_ASC_Bridge_Demo.lgb	Platform Designer Logic (xml)
	MachXO3_ASC_Bridge_Demo_ASC0.asc	Platform Designer ASC0 (xml)
	MachXO3_ASC_Bridge_Demo_ASC1.asc	Platform Designer ASC1 (xml)
	MachXO3_ASC_Bridge_Demo_ASC2.asc	Platform Designer ASC2 (xml)
	MachXO3_ASC_Bridge_Demo_ASC3.asc	Platform Designer ASC3 (xml)
Top\impl1\MachXO3_ASC_Bridge_Demo_ptm		
	MachXO3_ASC_Bridge_Demo_ASC0.hex	ASC 0 Programming
	MachXO3_ASC_Bridge_Demo_ASC1.hex	ASC 1 Programming
	MachXO3_ASC_Bridge_Demo_ASC2.hex	ASC 2 Programming
	MachXO3_ASC_Bridge_Demo_ASC3.hex	ASC 3 Programming

5. Programming the ASC and MachXO3 Devices

To program the ASC and MachXO3 devices:

1. Download the **MachXO3_ASC_Bridge_Demo.zip** file from the Lattice website to a folder in your computer.
2. Open Diamond version 3.10 or later.
3. Click **File > Open > Archived Project** and select the **MachXO3_ASC_Bridge_Demo.zip** file.
4. Browse to the destination directory where the demo files will be unarchived.
5. Open Platform Designer (a tool included with Diamond) by double-clicking the **MachXO3_ASC_Bridge_Demo.ptm** file in the Diamond File List.

   Note: You can also click the **Platform Designer** icon on the toolbar or click the **Tools > Platform Designer** menu as shown in Figure 5.1 to open the Platform Designer tool.

Figure 5.1. Platform Designer Menu Item and Icon

6. On the left side of the interface, click **Build** as shown in Figure 5.2.

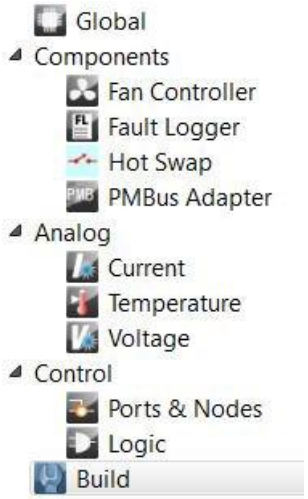


Figure 5.2. Platform Designer Build Selected

7. Click the **Compile** button.
8. When the compilation process is complete, click the **Pin Assignment** button. This opens the Diamond tool in Spreadsheet view. No further editing is required since all the pins are already assigned for the demo.

   Note: Ignore the Unknown Object errors in the output window. This is a known issue for version 3.10.
9. Click the **Platform Designer** tab.
10. Click the **Generate Jedec** button to generate the .jed file for the MachXO3 and the .hex files for the ASCs. When the processes are complete, the **Summary Status** switched from showing the red *x* **Not Current** to the green check mark **Current**.
11. Open Diamond Programmer by clicking the **Programmer** button on the toolbar as shown in Figure 5.3.

Figure 5.3. Diamond Programmer Menu Item and Icon

Note: You can also select **Tools > Programmer** on the menu to open Diamond Programmer.

12. To program the MachXO3 and the four L-ASC10 devices, click the **Program** button on the toolbar as shown in Figure 5.4.

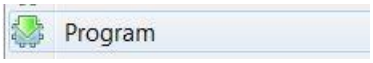


Figure 5.4. Diamond Designer Program Menu Item and Icon.

When the green **PASS** status is displayed, the platform is ready for the demo.

6. Running the Sequencing Demo

The sequencing demo turns on LEDs, which simulates DC/DC enables and/or reset signals of a system. The sequence pauses in several places waiting for the POTs to be close to the right position. This simulates the delay of a DC/DC supply either ramping up or down. In the following sections, the logical signal names used in Platform Designer are shown in **bold**.

6.1. Sequential Power Plane Power Up

This part of the demo sequences ON the LEDs controlled by the ASCs in Power Plane 1, then Power Plane 2, then Power Plane 3, and finally Power Plane 4. Eight LEDs on the MachXO3-9400 Development Board are used to display the system status or fault count, depending on the piano switch SW1-4 position, as listed in Table 6.1.

Table 6.1. MachXO3-9400 Development Board Status LEDs

LED	Logical Signal Name	SW1-4 Position PIO_XO3_SW1_4_Status_Mode	
		Up = True	Down = False
D4	**PIO_Status_LED_0**	Fault Log Enabled	Fault Count Bit 0
D8	**PIO_Status_LED_1**	ASC1 Breakout Board Installed	Fault Count Bit 1
D7	**PIO_Status_LED_2**	ASC2 Breakout Board Installed	Fault Count Bit 2
D1	**PIO_Status_LED_3**	ASC3 Breakout Board Installed	Fault Count Bit 3
D2	**PIO_Status_LED_4**	Power Plane 1 – Powered Up	Fault Count Bit 4
D5	**PIO_Status_LED_5**	Power Plane 2 – Powered Up	Fault Count Bit 5
D3	**PIO_Status_LED_6**	Power Plane 3 – Powered Up	Fault Count Bit 6
D6	**PIO_Status_LED_7**	Power Plane 4 – Powered Up	Fault Count Bit 7

Before the sequence is started, the following status LEDs on the MachXO3-9400 Development Board should be on:

- D4 (**PIO_Status_LED_0**) – Fault Logging Enabled
- D8 (**PIO_Status_LED_1**) – ASC1 Breakout Board Installed
- D7 (**PIO_Status_LED_2**) – ASC2 Breakout Board Installed
- D1 (**PIO_Status_LED_3**) – ASC3 Breakout Board Installed

To run the power ON sequence:

1. Press **SW1 (PIO_BRG_SW1_Restart)** on the ASC Bridge Board to start the sequence. The LEDs on the MachXO3-9400 Development Board turn on in the order listed below:
   a. D9 **(A0_GPIO1_LED)**
   b. D10 **(A0_GPIO2_LED)**
   c. D11 **(A0_GPIO3_LED)**
2. Using the small screwdriver, slowly rotate **POT1** on the MachXO3-9400 Development Board, counter-clockwise to mid-scale (about 1.5 V so that **A0_VM7_POT_OK** is true). LED D12 (**A0_GPIO4_LED**) turns on. Then the following LEDs turn on in the order listed below:
   a. D13 **(A0_GPIO5_LED)**
   b. D14 **(A0_GPIO6_LED)**
   c. D15 **(A0_GPIO8_LED)**
   d. D16 **(A0_GPIO9_LED)**
   e. D17 **(A0_GPIO10_LED)**

Power Plane 1 is now completely powered up and the MachXO3-9400 Development Board status LED D2 (**PIO_Status_LED_4**) is on. Power Plane 2 starts to power up. The LEDs on ASC1 Breakout Board turn on in the order listed below:

f. D2 **(A1_GPIO2_LED)**

g. D3 **(A1_GPIO3_LED)**

3. Slide **ASC1 – POT7** to mid-scale (about 1.5 V so that **A1_VM7_POT_OK** is true). The following LED turns on:
   a. D4 **(A1_GPIO4_LED)**
4. Slide **ASC1 – POT8** to mid-scale (about 1.5 V so that **A1_VM8_POT_OK** is true). The LEDs turn on in the order listed below:
   a. D5 **(A1_GPIO5_LED)**
   b. D6 **(A1_GPIO6_LED)**
   c. D7 **(A1_GPIO8_LED)**
   d. D8 **(A1_GPIO9_LED)**

   Power Plane 2 is now completely powered up and the MachXO3-9400 Development Board status LED **D5** (**PIO_Status_LED_5**) is on. Power Plane 3 starts to power up. The LEDs on ASC2 Breakout Board turn on in the order listed below:
   e. D2 **(A2_GPIO2_LED)**
   f. D3 **(A2_GPIO3_LED)**
5. Slide **ASC2 – POT7** to mid-scale (about 1.5 V so that **A2_VM7_POT_OK** is true). The following LED turns on:
   a. D4 **(A2_GPIO4_LED)**
6. Slide **ASC2 – POT8** to mid-scale (about 1.5 V so that **A2_VM8_POT_OK** is true). The LEDs turn on in the order listed below:
   a. D5 **(A2_GPIO5_LED)**
   b. D6 **(A2_GPIO6_LED)**
   c. D7 **(A2_GPIO8_LED)**
   d. D8 **(A2_GPIO9_LED)**

   Power Plane 3 is now completely powered up and the MachXO3-9400 Development Board status LED D3 (**PIO_Status_LED_6**) is on. Power Plane 4 starts to power up. The LEDs on ASC3 Breakout Board turn on in the order listed below:
   e. D2 **(A3_GPIO2_LED)**
   f. D3 **(A3_GPIO3_LED)**
7. Slide **ASC3 – POT7** to mid-scale (about 1.5 V so that **A3_VM7_POT_OK** is true). The following LED turns on:
   a. D4 **(A3_GPIO4_LED)**
8. Slide **ASC2 – POT8** to mid-scale (about 1.5 V so that **A3_VM8_POT_OK** is true). The LEDs turn on in the order listed below:
   a. D5 **(A3_GPIO5_LED)**
   b. D6 **(A3_GPIO6_LED)**
   c. D7 **(A3_GPIO8_LED)**
   d. D8 **(A3_GPIO9_LED)**

All four Power Planes are now completely powered up and the MachXO3-9400 Development Board status LED D6 (**PIO_Status_LED_7**) is on.

6.2. Sequential Power Plane Power Down

This part of the demo sequences OFF the LEDs in reverse order so that Power Plane 4 is the first to power down, then Power Plane 3, then Power Plane 2, and finally Power Plane 1 is powered down.

To run the power OFF sequence:

1. Press **SW4** (**PIO_XO3_SW4_Shutdown_Normal**) on the MachXO3-9400 Development Board to start the power-down sequence. The LEDs on the ASC3 Breakout Board turn off in the order listed below:
   a. D8 **(A3_GPIO9_LED)**
   b. D7 **(A3_GPIO8_LED)**
   c. D6 **(A3_GPIO6_LED)**
   d. D5 **(A3_GPIO5_LED)**
2. Slide **ASC3 – POT8** towards zero (less than 1 V so that **A3_VM8_POT_MIN** is false). The following LED turns off:
   a. D4 **(A3_GPIO4_LED)**
3. Slide **ASC3 – POT7** towards zero (less than 1 V so that **A3_VM7_POT_MIN** is false). The LEDs turn off in the order listed below:
   a. D3 **(A3_GPIO3_LED)**
   b. D2 **(A3_GPIO2_LED)**

   Power Plane 4 is now completely powered down and MachXO3-9400 Development Board status LED D6 (**PIO_Status_LED_7**) is off. Power Plane 3 starts to power down. The LEDs on the ASC2 Breakout Board turn off in the order listed below:
   c. D8 **(A2_GPIO9_LED)**
   d. D7 **(A2_GPIO8_LED)**
   e. D6 **(A2_GPIO6_LED)**
   f. D5 **(A2_GPIO5_LED)**
4. Slide **ASC2 – POT8** towards zero (less than 1 V so that **A2_VM8_POT_MIN** is false). The following LED turns off:
   a. D4 **(A2_GPIO4_LED)**
5. Slide **ASC2 – POT7** towards zero (less than 1 V so that **A2_VM7_POT_MIN** is false). The LEDs turn off in the order listed below:
   a. D3 **(A3_GPIO3_LED)**
   b. D2 **(A3_GPIO2_LED)**

   Power Plane 3 is now completely powered down and the MachXO3-9400 Development Board status LED D3 (**PIO_Status_LED_6**) is off. Power Plane 2 starts to power down. The LEDs on the ASC1 Breakout Board turn off in the order listed below:
   c. D8 **(A1_GPIO9_LED)**
   d. D7 **(A1_GPIO8_LED)**
   e. D6 **(A1_GPIO6_LED)**
   f. D5 **(A1_GPIO5_LED)**
6. Slide ASC1 – POT8 towards zero (less than 1 V so that **A1_VM8_POT_MIN** is false). The following LED turns off:
   a. D4 **(A1_GPIO4_LED)**
7. Slide ASC1 – POT7 towards zero (less than 1 V so that **A1_VM7_POT_MIN** is false). The LEDs turn off in the order listed below:
   a. D3 **(A1_GPIO3_LED)**
   b. D2 **(A1_GPIO2_LED)**

Power Plane 2 is now completely powered down and the MachXO3-9400 Development Board status LED D5 (**PIO_Status_LED_5**) is off. Power Plane 1 starts to power down. The LEDs on the MachXO3-9400 Development Board turn off in the order listed below:

c. D17 **(A0_GPIO10_LED)**
d. D16 **(A0_GPIO9_LED)**
e. D15 **(A0_GPIO8_LED)**
f. D14 **(A0_GPIO6_LED)**
g. D13 **(A0_GPIO5_LED)**

8. Using a small screwdriver, slowly rotate POT1 clockwise towards zero (less than 1.0 V so that **A0_VM7_POT_MIN** is false). The LEDs turn off in the order listed below:
   a. D12 **(A0_GPIO4_LED)**
   b. D11 **(A0_GPIO3_LED)**
   c. D10 **(A0_GPIO2_LED)**
   d. D9 **(A0_GPIO1_LED)**

All four Power Planes are now completely powered down and the MachXO3-9400 Development Board status LED D2 (**PIO_Status_LED_4**) is off.

6.3. Re-sequence with Emergency (Fast) Shutdown

This part of the demo powers down all four Power Planes without sequencing.

To power down without sequencing:

1. Repeat all the steps in the Sequential Power Plane Power Up section.
2. Press SW2 (**PIO_BRG_Shutdown_Fast**) on the ASC Bridge Board. All the LEDs in all four Power Planes turn off at the same time.
3. Press SW1 (**PIO_BRG_Restart**) on the ASC Bridge Board. The LEDs in Power Plane 1 sequences on, then Power Plane 2, then Power Plane 3, and finally Power Plane 4. The LEDs sequence on without delay since all the POTs are left around the 1.5 V position.
4. Repeat all the steps in the Sequential Power Plane Power Down section.

6.4. Parallel Power Plane Power Up and Power Down

This part of the demo starts the power up sequence in all four Power Planes at the same time.

To power up at the same time:

1. On the MachXO3-9400 Development Board, change the position of the SW1-1 from up to down (**PIOXO3_SW1_1_Sequential_Mode**).
2. Press SW1 (**PIO_BRG_Restart**) on the ASC Bridge Board. All the LEDs in all four Power Planes turn on at the same time in the respective order for each board listed below:

   MachXO3-9400 Development Board
   a. D9 **(A0_GPIO1_LED)**
   b. D10 **(A0_GPIO2_LED)**
   c. D11 **(A0_GPIO3_LED)**

   ASC 1 Breakout Board
   a. D2 **(A1_GPIO2_LED)**
   b. D3 **(A1_GPIO3_LED)**

ASC 2 Breakout Board

a. D2 **(A2_GPIO2_LED)**

b. D3 **(A2_GPIO3_LED)**

ASC 3 Breakout Board

a. D2 **(A3_GPIO2_LED)**

b. D3 (**A3_GPIO3_LED**)

3. Choose any Power Plane to proceed with by moving the respective POT(s) close to the mid-range (about 1.5 V). The LEDs in that plane sequence on. Continue to move all the POTs until all the Power Planes are powered up.
4. Press SW4 (**PIO_XO3_SW4_Shutdown_Normal**) on the MachXO3-9400 Development Board. The LEDs in all the Power Planes turn off at the same time, in the respective order for each board listed below:

   MachXO3-9400 Development Board

   a. D17 (**A0_GPIO10_LED**)

   b. D16 (**A0_GPIO9_LED**)

   c. D15 (**A0_GPIO8_LED**)

   d. D14 (**A0_GPIO6_LED**)

   e. D13 (**A0_GPIO5_LED**)

   ASC 1 Breakout Board

   a. D8 **(A1_GPIO9_LED)**

   b. D7 (**A1_GPIO8_LED**)

   c. D6 (**A1_GPIO6_LED**)

   d. D5 (**A1_GPIO5_LED**)

   ASC 2 Breakout Board

   a. D8 **(A2_GPIO9_LED)**

   b. D7 (**A2_GPIO8_LED**)

   c. D6 (**A2_GPIO6_LED**)

   d. D5 (**A2_GPIO5_LED**)

   ASC 3 Breakout Board

   a. D8 **(A3_GPIO9_LED)**

   b. D7 (**A3_GPIO8_LED**)

   c. D6 (**A3_GPIO6_LED**)

   d. D5 (**A3_GPIO5_LED**)
5. Choose any Power Plane to proceed with by moving the respective POT(s) towards zero (less than 1.0 V). The LEDs in that plane sequence off. Continue to move all the POTs until all the Power Planes are powered down.
6. Repeat Step 2 to Step 5, but feel free to use a different order of Power Plane in both the power-up and power-down steps.

6.5. Removing and Installing Power Planes 2 - 4

This part of the demo shows the **Reset Type: Optional** design feature. Power Planes 2 - 4 are associated with ASC Breakout Boards 1 - 3. This section details the steps to remove and install Power Plane 2 (ASC1 Breakout Board). The procedure, however, is also applicable to Power Planes 3 and 4 (ASC2 and ASC3 Breakout Boards).

To remove and install Power Plane 2:

1. Repeat steps 2 and 3 in the Parallel Power Plane Power Up and Power Down section so that all four Power Planes are powered up.
2. Press SW2 (**A1_GPIO10_SW2**) on ASC 1 Breakout Board and the LEDs on that board turn off in the following order:
   a. D8 **(A1_GPIO9_LED)**
   b. D7 **(A1_GPIO8_LED)**
   c. D6 **(A1_GPIO6_LED)**
   d. D5 **(A1_GPIO5_LED)**
3. Slide ASC1 – POT8 towards zero (less than 1 V so that **A1_VM8_POT_MIN** is false) and the following LED turns off:
   a. D4 **(A1_GPIO4_LED)**
4. Slide ASC1 – POT7 towards zero (less than 1 V so that **A1_VM7_POT_MIN** is false) and the LEDs turn off in the order listed below:
   a. D3 **(A1_GPIO3_LED)**
   b. D2 **(A1_GPIO2_LED)**

   Power Plane 2 is now completely powered down and the MachXO3-9400 Development Board status LED D5 (**PIO_Status_LED_5**) is off.
5. Carefully unplug the ASC 1 Breakout Board from the ASC Bridge Board and the MachXO3-9400 Development Board status LED D8 (**PIO_Status_LED_1**) turns off.
6. Carefully plug the ASC 1 Breakout board back into the ASC Bridge Board and the MachXO3-9400 Development Board status LED D8 (**PIO_Status_LED_1**) turns on.
7. Press SW2 (**A1_GPIO10_SW2**) on the ASC 1 Breakout Board and the LEDs on that board turn on in the order listed below:
   a. D2 **(A1_GPIO2_LED)**
   b. D3 **(A1_GPIO3_LED)**
8. Slide ASC1 – POT7 close to the mid-range (about 1.5 V so that **A1_VM7_POT_OK** is true) and the following LED turns on:
   a. D4 **(A1_GPIO4_LED)**
9. Slide ASC1 – POT8 close to the mid-range (about 1.5 V so that **A1_VM8_POT_OK** is true) and the LEDs turn on in the order listed below:
   a. D5 **(A1_GPIO5_LED)**
   b. D6 **(A1_GPIO6_LED)**
   c. D7 **(A1_GPIO8_LED)**
   d. D8 **(A1_GPIO9_LED)**

Power Plane 2 is now completely powered up and the MachXO3-9400 Development Board status LED D5 (**PIO_Status_LED_5**) is on. This portion of the demo can be repeated for both Power Planes 3 and 4. Also, all three Power Planes (2 - 4) can be removed and then reinstalled in any sequence. However, if you are swapping ASC Breakout Board positions, make sure to set SW3 (I2C_ADX) accordingly (see the Setting up the Demo section).

7. Running the Fault Logging Demo

Move the piano switch SW1-4 on the MachXO3-9400 Development Board from up to down position so that the status LEDs display in binary the number of fault records (see Table 6.1). If all the steps in the Running the Sequencing Demo section are followed correctly, then the status LEDs should all be off, indicating that no faults are recorded. If any LEDs are on, follow the procedure in the Clearing the Faults section to clear the faults.

7.1. Creating Under Voltage Faults

Power up all four Power Planes by repeating the steps either in the Sequential Power Plane Power Up section or in the Parallel Power Plane Power Up and Power Down section.

Note: In the following procedure, allow a short delay between each fault event. There is a 2.5-second fault blanking logic in the design to prevent multiple fault records from being recorded for a single event.

To create under voltage faults:

1. Using a small screwdriver, quickly rotate POT1 on the MachXO3-9400 Development Board clockwise towards zero and then back to mid-scale (so that **A0_VM7_POT_OK** is false for a moment). The status LEDs on the MachXO3-9400 Development Board display binary 1; indicating one fault record is stored.
2. Quickly slide ASC1 – POT7 on the ASC1 Breakout Board towards zero and then back to mid-scale (so that **A1_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 2; indicating two fault records are stored.
3. Quickly slide ASC1 – POT8 on the ASC1 Breakout Board towards zero and then back to mid-scale (so that **A1_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 3; indicating three fault records are stored.
4. Quickly slide ASC2 – POT7 on the ASC2 Breakout Board towards zero and then back to mid-scale (so that **A2_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 4; indicating four fault records are stored.
5. Quickly slide ASC2 – POT8 on the ASC2 Breakout Board towards zero and then back to mid-scale (so that **A2_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 5; indicating five fault records are stored.
6. Quickly slide ASC3 – POT7 on the ASC3 Breakout Board towards zero and then back to mid-scale (so that **A3_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 6; indicating six fault records are stored.
7. Quickly slide ASC3 – POT8 on the ASC3 Breakout Board towards zero and then back to mid-scale (so that **A3_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 7; indicating seven fault records are stored.

7.2. Creating Over Voltage Faults

(Continuing from the previous section.)

To create over voltage faults:

1. Using a small screwdriver, quickly rotate POT1 on the MachXO3-9400 Development Board counter-clockwise towards 3 V and then back to mid-scale (so that **A0_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 8; indicating eight fault records are stored.
2. Quickly slide ASC1 – POT7 on the ASC1 Breakout Board towards 3 V and then back to mid-scale (so that **A1_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 9; indicating nine fault records are stored.
3. Quickly slide ASC1 – POT8 on the ASC1 Breakout Board towards 3 V and then back to mid-scale (so that **A1_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 10; indicating ten fault records are stored.

4. Quickly slide ASC2 – POT7 on the ASC2 Breakout Board towards 3 V and then back to mid-scale (so that **A2_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 11; indicating eleven fault records are stored.
5. Quickly slide ASC2 – POT8 on the ASC2 Breakout Board towards 3 V and then back to mid-scale (so that **A2_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 12; indicating twelve fault records are stored.
6. Quickly slide ASC3 – POT7 on the ASC3 Breakout Board towards 3 V and then back to mid-scale (so that **A3_VM7_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 13; indicating 13 fault records are stored.
7. Quickly slide ASC3 – POT8 on the ASC3 Breakout Board towards 3 V and then back to mid-scale (so that **A3_VM8_POT_OK** is momentarily false). The status LEDs on the MachXO3-9400 Development Board display binary 14; indicating 14 fault records are stored.

7.3. Reading Fault Records

7.3.1. Reading and Saving the Fault Records from the MachXO3 UFM

To read the Fault Records from the UFM, which is inside the MachXO3, and save them to a file:

1. In Diamond Programmer double-click on the current Operation to open the **MachXO3LF – LCMXO3LF – Device Properties** dialog box.
2. Under **Device Operation**, select the following options as shown in in Figure 7.1.
   - **Access mode** – **Flash Background Mode**
   - **Operation** – **XFLASH Read and Save**
3. Under **Readback Options**, save the records into a .jed file. In **Save to file**, click the **Browse** button to select the target folder and enter the file name of the .jed file.
4. Click **OK**.

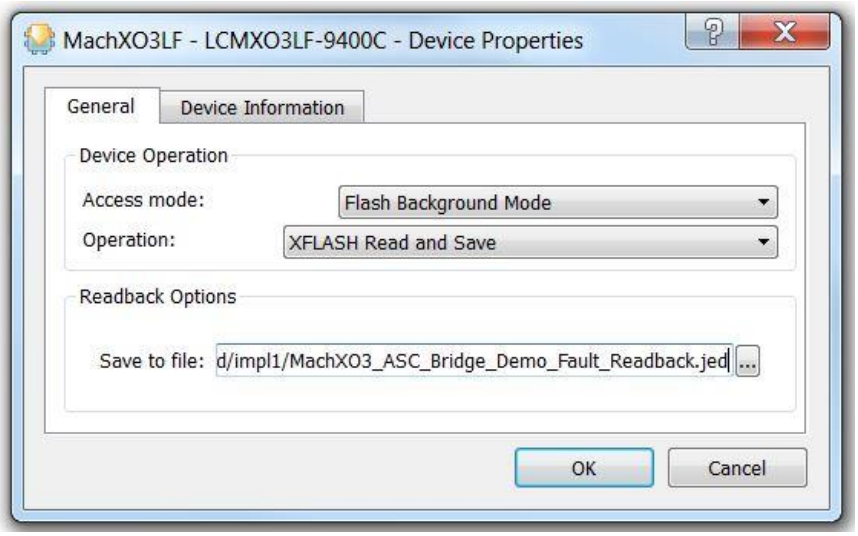


Figure 7.1. Reading the Fault Records from the UFM

5. Click the **Program** button.
6. After Diamond Programmer completes the process and the yellow DONE status is displayed, use Notepad or other text editors to open the .jed file just written by Diamond Programmer.
7. Search (Ctrl-F) for the second occurrence of the *L* character. This is the start of the UFM. The text file should look similar to Figure 7.2.

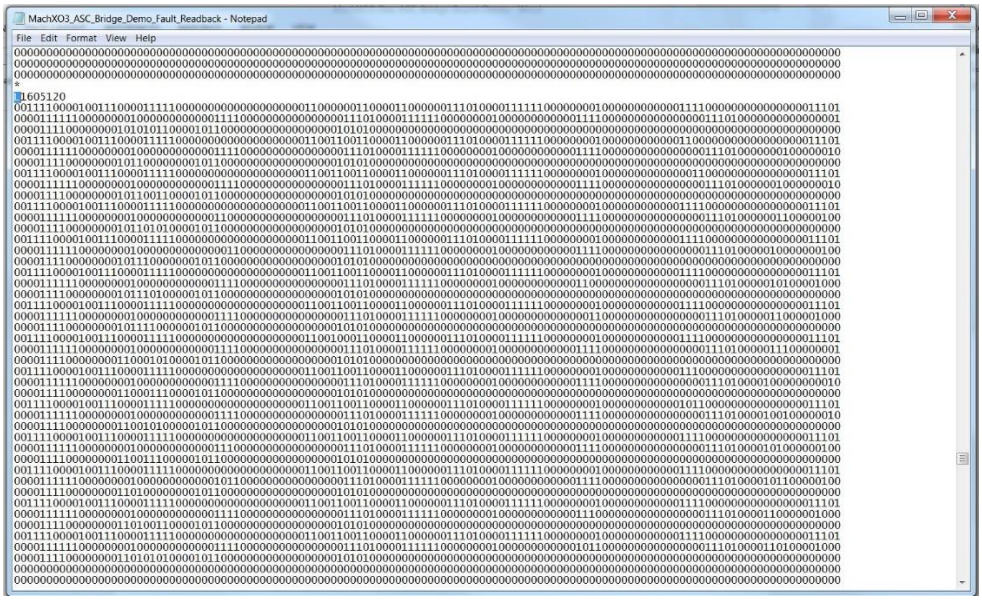


Figure 7.2. Notepad View of All 14 Fault Records

7.3.2. Understanding the Fault Record

Each fault record is 39 bytes (size = 0x27 as shown in Figure 7.3) and uses three UFM pages (16 bytes per page).

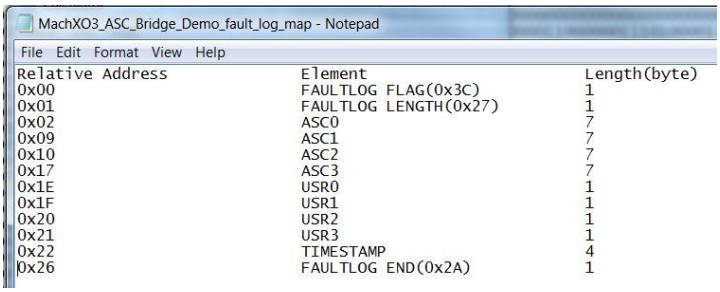


Figure 7.3. Fault Log Map File for the Demo

Two of the fault records are copied and annotated in Listing 7.1 and Listing 7.2.

In Listing 7.1, the first under-voltage fault is identified by both VMON_7A and VMON_7B status bits for ASC0 being zero. VMON_7A is the windowed status bit and is true if the voltage is between the lower-limit and upper-limit. VMON_7B is the status bit of the lower-limit comparator. In normal operation, both bits should be a one.

In Listing 7.1, the USER bytes indicate Fault Record 0, Fault in Power Plane 1 (ASC0), and all four Power Planes are powered up.

In Listing 7.1, the Timestamp bytes indicate the fault occurred 2,987 seconds after power is applied to the MachXO3-9400 Development Board.

In Listing 7.2, the second to last over-voltage fault is identified by seeing that VMON_7A status is zero and VMON_7B status is one for ASC3. VMON_7A is the windowed status bit and is true if the voltage is between the lower-limit and upper-limit. VMON_7B is the status bit of the lower-limit comparator. In normal operation, both bits should be a one.

In Listing 7.2, the USER bytes indicate Fault Record 12, Fault in Power Plane 4 (ASC3), and all four Power Planes are powered up.

Note: The Fault Counter in the design is incremented after the trigger event so USER0 is always one less than the Fault Count displayed by the Status LEDs on the MachXO3-9400 Development Board.

In Listing 7.2, the Timestamp bytes indicate the fault occurred 3,027 seconds after power is applied to the MachXO3-9400 Development Board.

Listing 7.1. UFM Pages 0, 1, and 2 - Fault Record 1

```
UFM Page 0
Flag     Length   ASC0-0   ASC0-1   ASC0-2   ASC0-3   ASC0-4   ASC0-5
00111100 00100111 00001111 10000000 00000000 00000110 00000110 00011000
                                                        ||
                                                  VMON_7B|
                                                   VMON_7A

ASC0-6   ASC1-0   ASC1-1   ASC1-2   ASC1-3   ASC1-4   ASC1-5   ASC1-6
00011101 00001111 11000000 00100000 00000001 11100000 00000000 00011101

UFM Page 1
ASC2-0   ASC2-1   ASC2-2   ASC2-3   ASC2-4   ASC2-5   ASC2-6   ASC3-0
00001111 11000000 00100000 00000001 11100000 00000000 00011101 00001111
ASC3-1   ASC3-2   ASC3-3   ASC3-4   ASC3-5   ASC3-6   USER0    USER1
11000000 00100000 00000001 11100000 00000000 00011101 00000000 00000001
                                                          |        |
                                                Fault Count = 0    |
                                                          Fault Flag ASC0

UFM Page 2
USER2    USER3    TIME-0   TIME-1   TIME-2   TIME-3   Flag
00001111 00000000 10101011 00001011 00000000 00000000 00101010
    |                 |
    |                 Fault at 2987 seconds
    All Four Power Planes Up
```

Listing 7.2. UFM Pages 36, 37, and 38 - Fault Record 13

```
UFM Page 36
Flag     Length   ASC0-0   ASC0-1   ASC0-2   ASC0-3   ASC0-4   ASC0-5
00111100 00100111 00001111 10000000 00000000 00000110 01100110 00011000
ASC0-6   ASC1-0   ASC1-1   ASC1-2   ASC1-3   ASC1-4   ASC1-5   ASC1-6
00011101 00001111 11000000 00100000 00000001 11100000 00000000 00011101

UFM Page 37
ASC2-0   ASC2-1   ASC2-2   ASC2-3   ASC2-4   ASC2-5   ASC2-6   ASC3-0
00001111 11000000 00100000 00000001 11100000 00000000 00011101 00001111
ASC3-1   ASC3-2   ASC3-3   ASC3-4   ASC3-5   ASC3-6   USER0    USER1
11000000 00100000 00000001 11000000 00000000 00011101 00001100 00001000
                           ||                              |      |
                     VMON_7B|                Fault Count = 12      |
                      VMON_7A                        Fault Flag ASC3

UFM Page 38
USER2    USER3    TIME-0   TIME-1   TIME-2   TIME-3   Flag
00001111 00000000 11010011 00001011 00000000 00000000 00101010
    |                 |
    |                 Fault at 3027 seconds
    All Four Power Planes Up
```

7.4. Measuring the Fault Log Recording Time

To measure the fault log recording time:

1. Connect an oscilloscope probe to **EXPCON_IO17**, pin 11 of J1, on the ASC Bridge Board. Set the vertical scale to 1 V / div. Set the trigger to rising edge around 1.5 V and set the time scale to 1 ms / div.
2. With the piano switch SW1-2 (**PIO_XO3_SW1_2_Enable_Fault_Log**) on the MachXO3-9400 Development Board in the up position and any or all Power Planes powered up, move a POT (in a powered up Power Plane) to cause a fault.
3. The captured pulse width on the oscilloscope is the total fault recording time from the initial trigger event; it should measure less than 3 ms.

   Note: If the POT is left in an over-voltage or under-voltage condition, faults reoccur every 2.5 seconds.

7.5. Clearing the Faults

To clear the faults:

1. Move the piano switch SW1-2 (**PIO_XO3_SW1_2_Enable_Fault_Log**) on the MachXO3-9400 Development Board from the up to down position to disable additional faults from being logged.
2. In Diamond Programmer, double-click on the current Operation to open the **MachXO3LF – LCMXO3LF – Device Properties** dialog box.
3. Under **Device Operation**, select the following options as shown in in Figure 7.4.
   - **Access mode: Flash Background Mode**
   - **Operation: XFLASH UFM Erase**
4. Click **OK**.

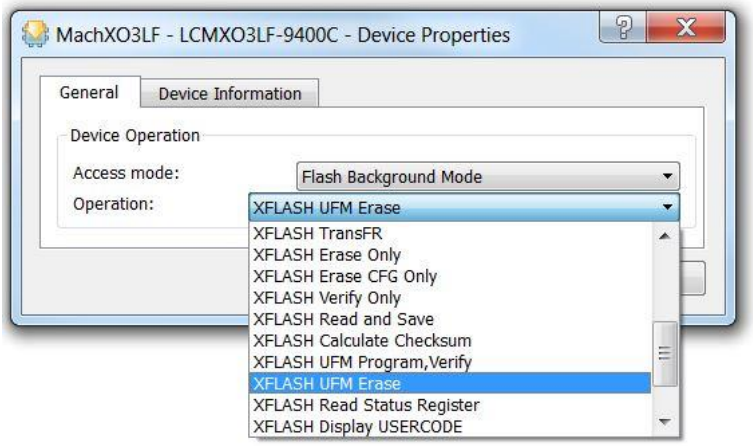


Figure 7.4. Erase the Fault Records in the UFM

5. Click the **Program** button.
6. After Diamond Programmer completes the process and the green **PASS** status is displayed, press SW5 (**PIO_XO3_SW5_Reset_Fault_Count**) on the MachXO3-9400 Development Board to reset the fault counter.
7. Move the piano switch SW1-2 on the MachXO3-9400 Development Board from the down to up position to enable logging of faults.

8. Analysis of Demo Design in Platform Designer

This section discusses a few of the design features used in this demo. Some may be obvious by going through the demo itself and some can only be appreciated by looking at the source code.

8.1. Logical Signal Names within Platform Designer

Platform Designer supports user defined Logical Signals so the engineer can include as much information in signal names as possible. This aids in documenting, maintaining, and troubleshooting the design. In this demo, the device, pin function, and board are included in the signal names so that you do not have to scan multiple documents and schematics.

8.1.1. Ports

In Platform Designer, the Ports are the external logic pins of the Platform Manager 2 design. In this demo, that means the MachXO3 digital I/O. By keeping the *PIO* in the Logical Name, it is clear in the other views that this signal is external, as shown in Figure 8.1. For example, the name **PIO_BRG_SW1_Restart** reveals that the signal is from switch 1 on the ASC Bridge board and it is used to restart the sequence.

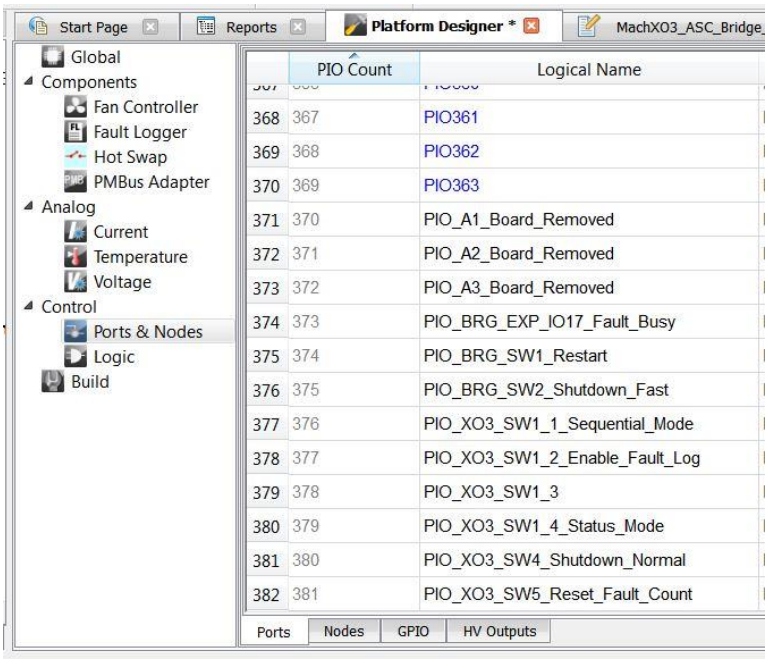


Figure 8.1. Platform Designer User Defined Port Names

8.1.2. Nodes

In Platform Designer, the Nodes are internal logic signals of the Platform Manager 2 design. Nodes are similar to a wire in a Verilog design and represent connections within the FPGA. By keeping *Node* in the name, it is clear in the other views that this signal is internal. Some of the node names are shown in Figure 8.2. The *_Powered_Up* and *_Powered_Down* nodes are semaphore flags that *hand-off* control from one sequence to the next. Note that the *_Fault* nodes are combinatorial so they can trigger the Fault Logger with minimal delay.

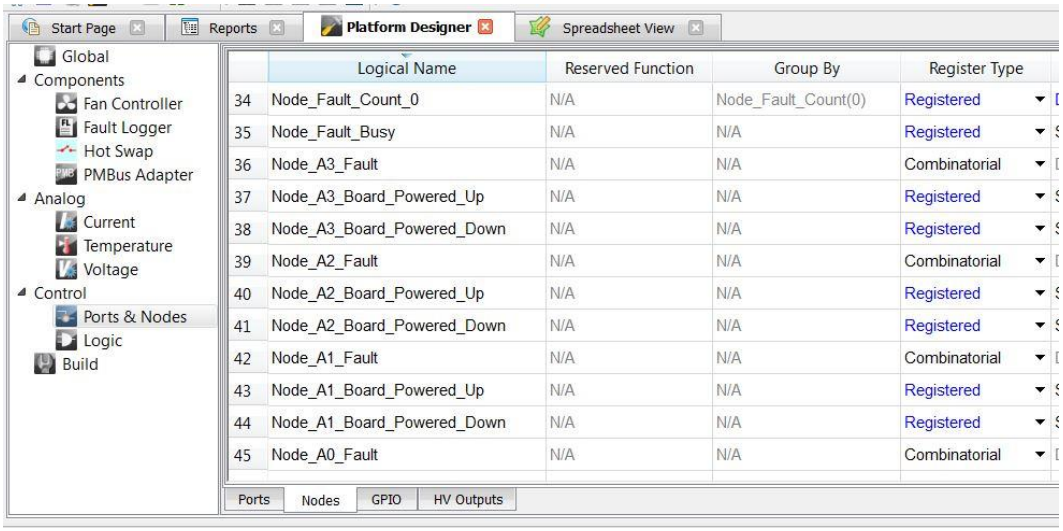


Figure 8.2. Platform Designer User Defined Node Names

8.1.3. GPIOs

In Platform Designer the GPIO are external inputs or open drain outputs from the ASC sections of the Platform Manager 2 design. In this demo the GPIO are all located on the L-ASC10 expander devices. By keeping both the ASC number and *GPIO* in the Logical Name it is clear in the other views the destination or origin of the signal. In this design, the GPIO names also include either LED or a switch number to indicate the function as shown in Figure 8.3.

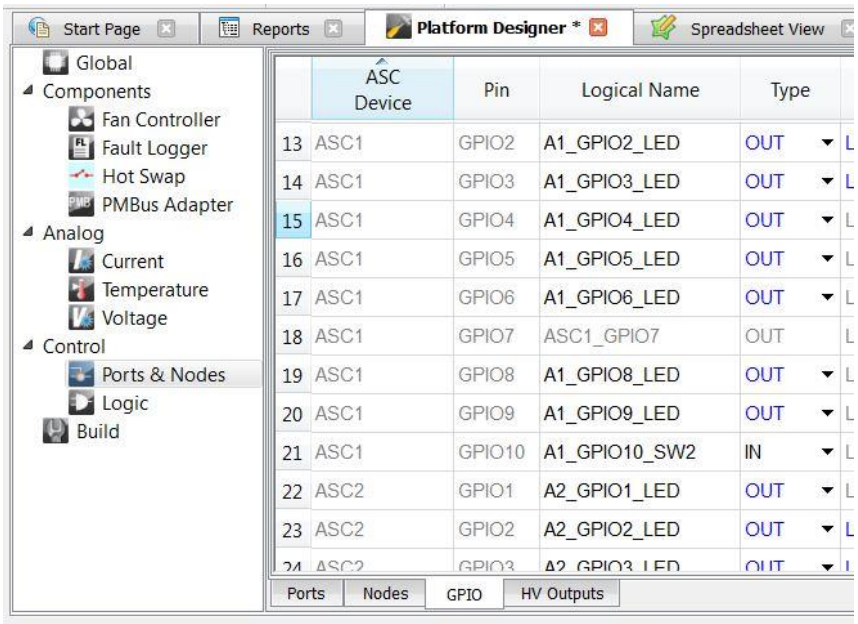
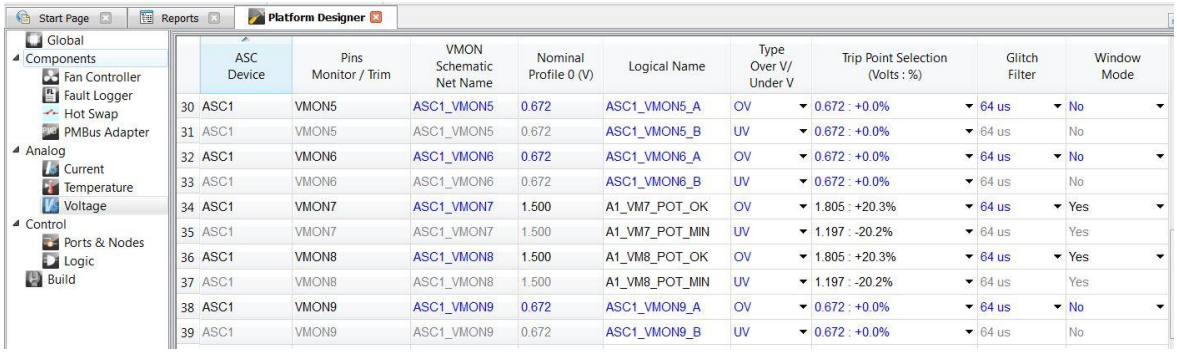


Figure 8.3. Platform Designer User Defined GPIO Logical Names

8.1.4. VMONs

In Platform Designer, the VMONs are external analog voltage inputs of the ASC sections of the Platform Manager 2 design. In this demo, the VMONs are all located on the L-ASC10 expander devices. By keeping both the ASC number and VMON number in the Logical Name it is clear in the other views the origin of the signal. As shown in Figure 8.4, the name **A1_VM7_POT_OK** indicates the POT is in the window (above 1.19 V and below 1.8 V) on VMON7 of ASC1 (Power Plane 2).



	ASC Device	Pins Monitor / Trim	VMON Schematic Net Name	Nominal Profile 0 (V)	Logical Name	Type Over V/ Under V	Trip Point Selection (Volts : %)	Glitch Filter	Window Mode
30	ASC1	VMON5	ASC1_VMON5	0.672	ASC1_VMON5_A	OV	0.672 : +0.0%	64 us	No
31	ASC1	VMON5	ASC1_VMON5	0.672	ASC1_VMON5_B	UV	0.672 : +0.0%	64 us	No
32	ASC1	VMON6	ASC1_VMON6	0.672	ASC1_VMON6_A	OV	0.672 : +0.0%	64 us	No
33	ASC1	VMON6	ASC1_VMON6	0.672	ASC1_VMON6_B	UV	0.672 : +0.0%	64 us	No
34	ASC1	VMON7	ASC1_VMON7	1.500	A1_VM7_POT_OK	OV	1.805 : +20.3%	64 us	Yes
35	ASC1	VMON7	ASC1_VMON7	1.500	A1_VM7_POT_MIN	UV	1.197 : -20.2%	64 us	Yes
36	ASC1	VMON8	ASC1_VMON8	1.500	A1_VM8_POT_OK	OV	1.805 : +20.3%	64 us	Yes
37	ASC1	VMON8	ASC1_VMON8	1.500	A1_VM8_POT_MIN	UV	1.197 : -20.2%	64 us	Yes
38	ASC1	VMON9	ASC1_VMON9	0.672	ASC1_VMON9_A	OV	0.672 : +0.0%	64 us	No
39	ASC1	VMON9	ASC1_VMON9	0.672	ASC1_VMON9_B	UV	0.672 : +0.0%	64 us	No

Figure 8.4. Platform Designer User Defined VMON Logical Names

8.2. Multiple State Machines

In Platform Designer, the Logic view is where the State Machine designs are edited in the Sequence tab, as shown in Figure 8.5. In this demo, there is a separate Sequence (State Machine) for each of the Power Planes (SM0, SM1, SM2, and SM3) and one additional state machine to count the faults.

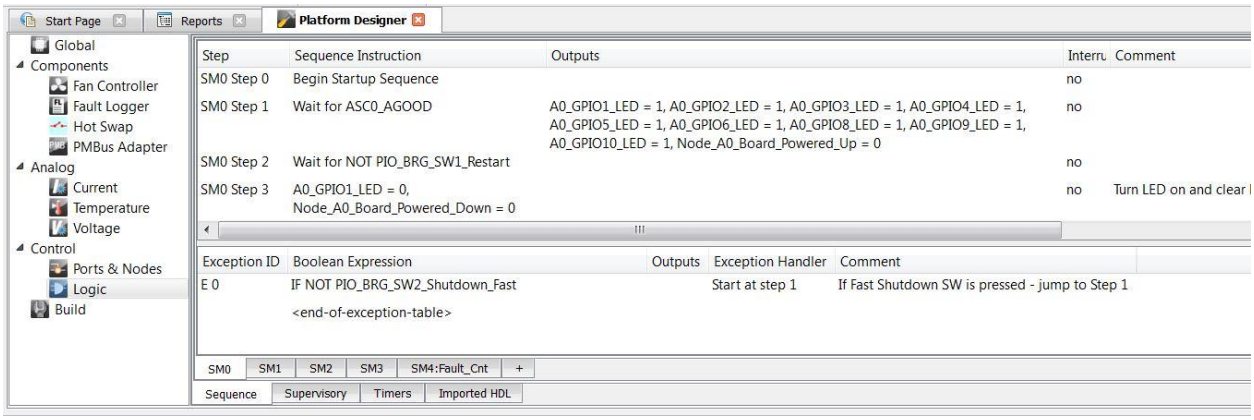


Figure 8.5. Platform Designer Logic Sequence View with Multiple State Machines

8.3. Semaphore Nodes

As mentioned in the Nodes section, the *_Powered_Up* and *_Powered_Down* nodes are used to transfer control from one Sequence (state machine) to the next.

8.3.1. Powered Up Nodes

The *Powered_Up* nodes are used to sequentially power up the Power Planes in order. In Step 20 of SM0, **Node_A0_Board_Powered_Up** is set true after all the Power Plane 1 LEDs are turned on. This triggers Step 2 in SM1 and starts the power up Sequence for Power Plane 2. The same logic is repeated for Power Planes 3 and 4 using **Node_A1_Board_Powered_Up** and **Node_A2_Board_Powered_Up** nodes.

8.3.2. Powered Down Nodes

The *_Powered_Down* nodes are used to sequentially power down the Power Planes in order. In Step 37 of SM3, **Node_A3_Board_Powered_Down** is set true after all the Power Plane 4 LEDs are turned off. This triggers Step 19 in SM2 and starts the power down Sequence for Power Plane 3. The same logic is repeated for Power Planes 2 and 1 using **Node_A2_Board_Powered_Down** and **Node_A1_Board_Powered_Down** nodes.

8.4. Fault Logging

This demo follows the guidance provided in the Fault Logging Using Platform Manager 2 (TN1277) technical note.

8.4.1. Fault Log Trigger

The Fault Logger component in Platform Designer requires a single trigger signal. In this demo, the **Node_Fault_Trigger** is the logical combination of individual fault signals from each of the Power Planes (Supervisory Logic Equations EQ 8 thru EQ 11), as shown in Figure 8.6. All the equations for the fault trigger use the Combinatorial Assignment Type instead of Registered to prevent delays in the logic. Delays in the trigger signal can result in false fault records by capturing the platform status after the fault is valid.

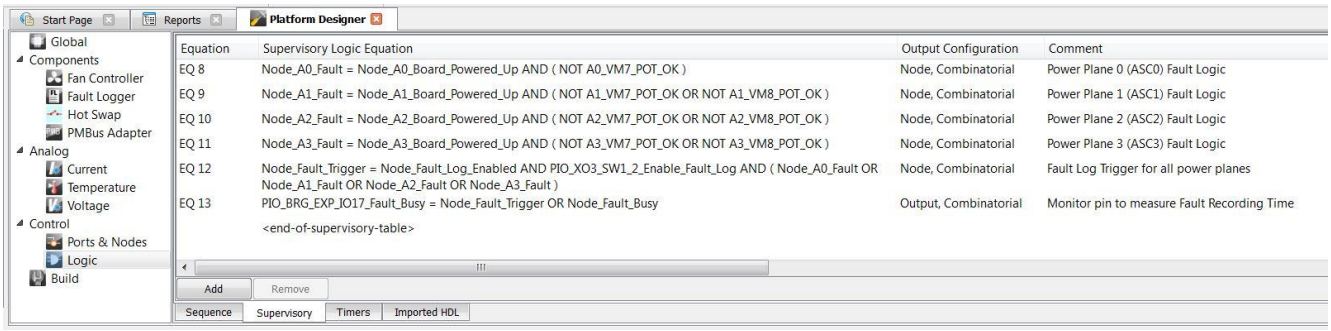


Figure 8.6. Platform Designer Supervisory Logic Equations for Fault Log Trigger

8.4.2. Fault Log Trigger Enable

In this demo, each Power Plane uses the respective *_Powered_Up* node to enable faults (Supervisory Logic Equations EQ 8 – EQ 11). This prevents false faults during either the power up or power down sequence. The piano switch (SW1-2) on the MachXO3-9400 Development Board is used to set the logic level of **PIO_XO3_SW1_2_Enable_Fault_Log**; when the switch is down the signal is false and fault logging is disabled. The signal **Node_Fault_Log_Enabled** must also be true to enable fault logging (see Supervisory Logic Equation EQ 12).

8.4.3. Fault Log Trigger Blocking

The faults in this demo are artificially created by moving a POT outside of the VMON window limits. This can result in a fault signal that is active significantly longer than the typical 3 ms it takes to record a fault and could generate tens of identical fault records. To record a single fault and allow time to return the POT into the VMON window a 2.5 second one-shot is implemented in **SM4:Fault_Cnt**, as shown in Figure 8.7. The signal **Node_Fault_Log_Enabled** is set in Step 2 while the Sequence is waiting for a fault to occur (**Node_Fault_Busy** is set true by the Fault Logger component from a trigger). After a fault is triggered **Node_Fault_Log_Enabled** is cleared in Step 3 and the Sequence waits in Step 4 for 2.5 seconds before looping back to Step 2 where fault logging is re-enabled. In this manner, the fault trigger signal is blocked to prevent multiple identical faults from being recorded from a single POT movement.

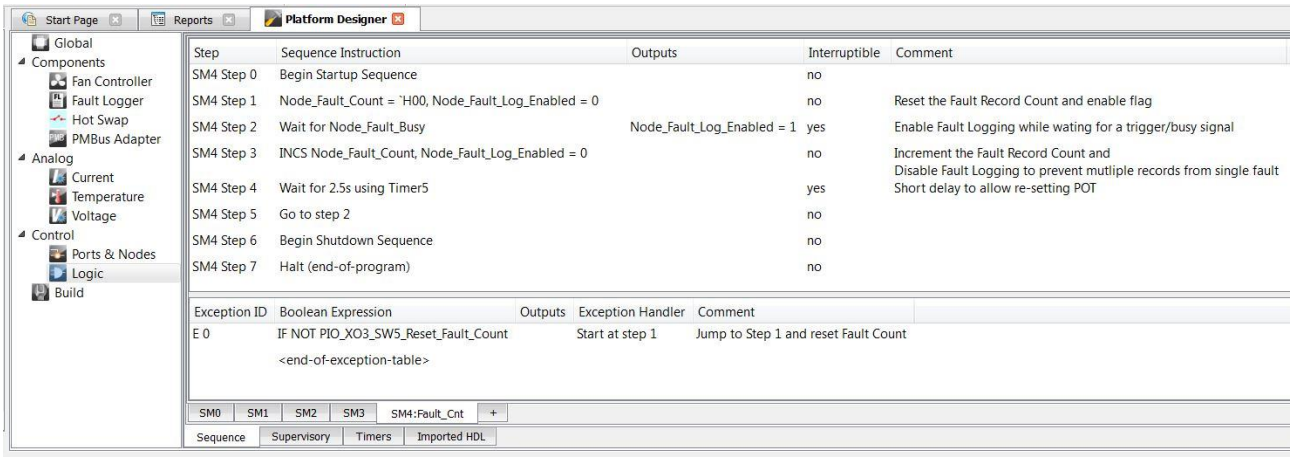


Figure 8.7. Platform Designer Logic Sequence View of SM4 : Fault Log Counter

8.4.4. Fault Log Counter

In Nodes view, eight nodes are grouped to create the fault log counter **Node_Fault_Count**. The count is reset in Step 1 of **SM4:Fault_Cnt** and incremented in Step 3 after each fault trigger event. The Fault Logger component does not keep count of faults, but the count is included in the record using the User Log Fields.

8.4.5. Fault Log User Log Fields

In the Fault Log Component, signals can be *dragged* from the list of **User Signals to Log** to the **User Log Fields**. User byte 0 contains the bits of the **Node_Fault_Count** and User byte 1 contains the four Power Plane fault signals. User bytes 2 and 3 contain the powered up and powered down signals for the four Power Planes, as shown in Figure 8.8. The unused 12 bits are recorded as zeros.

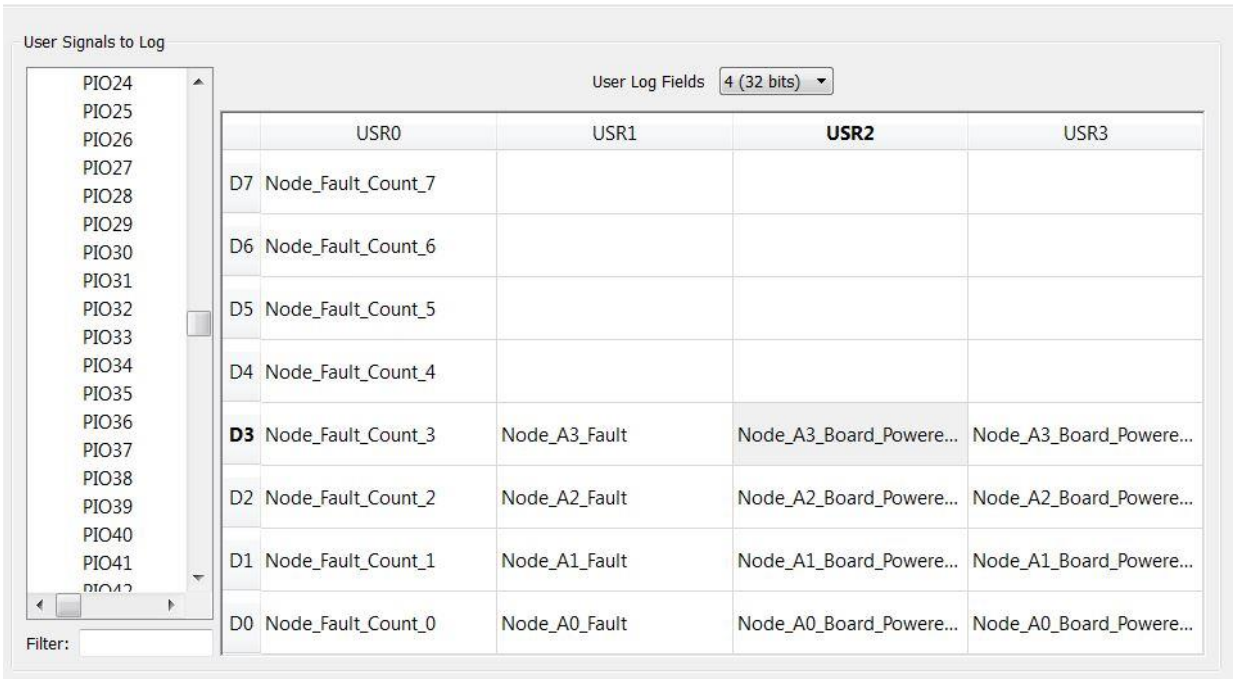


Figure 8.8. Platform Designer Fault Logger User Log - 32 Bits

8.5. Design Resources

In the Build view (see Figure 8.9), Platform Designer summarizes the resources used by this demo design. The Fault Logger component in this demo design is based on the Mico-8, which uses about 1,100 LUTs and 9 EBRs.

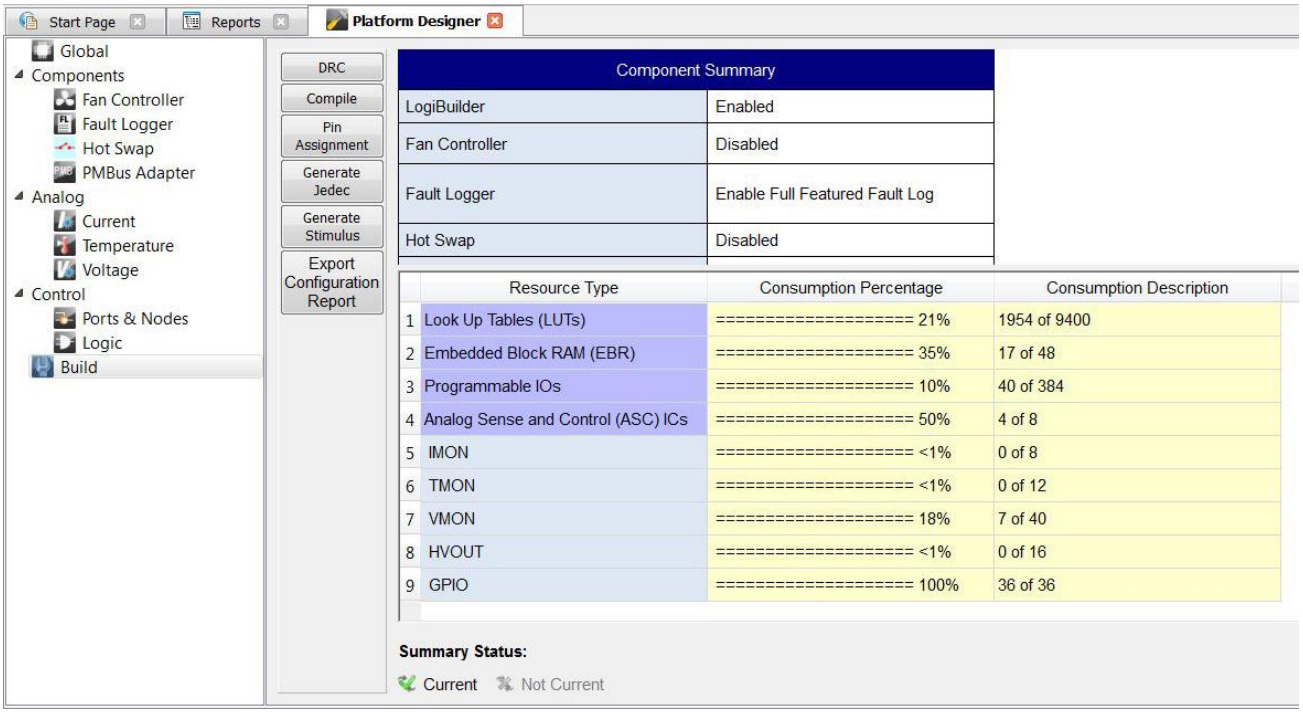


Figure 8.9. Platform Designer Summary of Resources

9. Design Considerations for Customer Application

9.1. Fault Logging to SPI

In this demo, the **Fault Logger** component is set to record the faults in the MachXO3 UFM. Alternatively, fault records can be stored in an external SPI flash memory device by selecting the **Log to External SPI Flash** radio button, as shown in Figure 9.1. When using external SPI storage, the records can be read and saved using Programmer in SPI Flash Programming mode, or via another SPI master. The Aardvark dongle can be used to read Fault Records from SPI if two items are populated by the user on the MachXO3-9400 Development Board; the header at JP2 is needed to connect to the cable and a zero ohm resistor at R161 to connect the SPI (U6) chip select to JP2 pin 9.

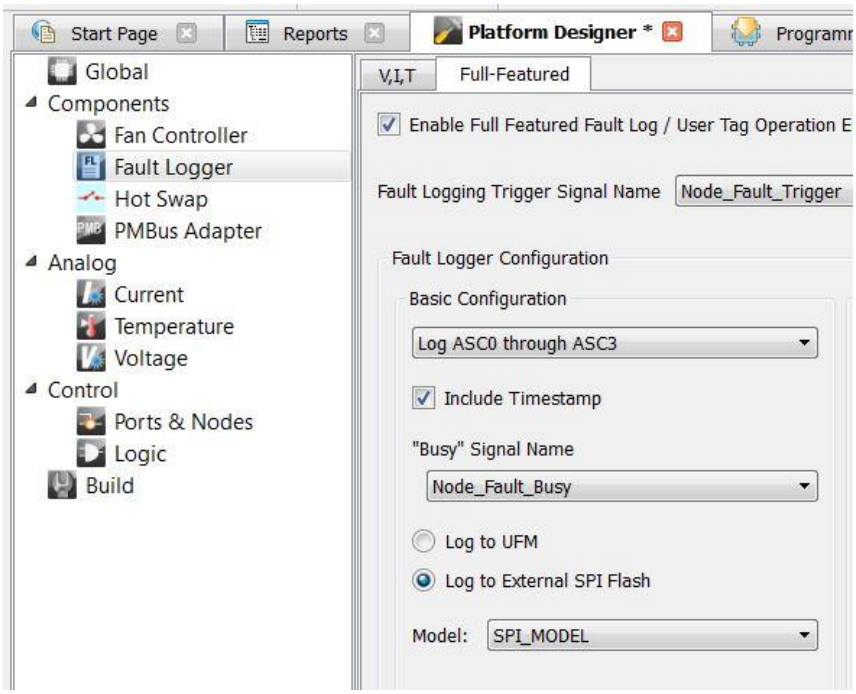


Figure 9.1. Platform Designer Fault Logger SPI Settings

9.2. Using an External Timestamp for Fault Logging

The internal timestamp is reset to zero every time the Platform Manager 2 is powered up. In applications where a real time clock is available, the clock data can be connected to the PIO pins and then mapped to the User Log Fields (shown in Figure 8.8) to record both date and time information.

References

For more information, refer to

- L-ASC10 Data Sheet (FPGA-DS-02038)
- MachXO3 Family Data Sheet (FPGA-DS-02032)
- Lattice Diamond Platform Designer User Guide (Included with Diamond install)
- ASC Bridge Board User Guide (FPGA-EB-02025)
- ASC Breakout Board User Guide (FPGA-EB-02023)
- MachXO3-9400 Development Board User Guide (FPGA-EB-02004)
- MachXO3 Programming and Configuration Usage Guide (TN1279)
- Fault Logging Using Platform Manager 2 (TN1277)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, November 2018

Section	Change Summary
All	Initial release

7th Floor, 111 SW 5th Avenue
Portland, OR 97204, USA
T 503.268.8000
www.latticesemi.com