# Adding Scalable Power and Thermal Management to Nexus FPGAs Using L-ASC10

## Application Note

FPGA-AN-02079-1.0

January 2024

## Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

## Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

# Contents

# Figures

# Tables

# Acronyms in This Document

A list of acronyms used in this document.

| Acronym | Definition |
|---|---|
| ASC | Analog Sense and Control (Lattice L-ASC10 device) |
| ASC-I/F | Three-wire interface from FPGA to ASC |
| ASCCLK | 8 MHz Clock Output from ASC0 |
| DSP | Digital Signal Processing |
| EBR | Embedded Block Ram |
| ECP5 | Lattice FPGA |
| EECMOS | Electrically Erasable CMOS Memory |
| EFB | Embedded Function Block |
| GPIO | General Purpose Input / Output |
| HDL | Hardware Definition Language (Verilog or VHDL) |
| HEX | Hexadecimal |
| IP | Intellectual Property |
| LRAM | Large Block Random Access Memory |
| LUT | Look Up Table |
| PFR | Platform Firmware Resiliency |
| PIO | Programmable Input / Output |
| PLL | Phase Locked Loop |
| PTM | Platform Manager |
| RDAT | Read Data (part of ASC-I/F) |
| RISC-V | Reduced Instruction Set Computer Version 5 |
| RTL | Register Transfer Level (high level hardware design language) |
| SDK | Software Development Kit |
| SOC | System On a Chip |
| SRAM | Static Random Access Memory |
| UART | Universal Asynchronous Receive and Transmit |
| VIT | Voltage Current Temperature (monitor and fault logging in ASC) |
| WDAT | Write Data (part of ASC-I/F) |
| WRCLK | Write Clock (part of ASC-I/F) |

# 1. Introduction

Complex hardware systems require a large number of resources for sensing analog signals, programmable digital control, and many GPIOs to provide interfaces with external components. Often, it is desirable to have a flexible system where the above-mentioned resources can be easily scaled based upon an application's growing requirements. The Lattice Platform ManagerTM 2 family consists of the Hardware Management Expander (L-ASC10), which, in combination with the hardware management controllers MachXO2TM, MachXO3TM, ECP5TM, and NexusTM family FPGAs, enables designers to create a flexible and scalable hardware system with centralized control. The Nexus product line includes the following FPGA families: MachXO5TM-NX, CrossLinkTM-NX, CertusTM-NX, and CertusProTM-NX with LUT densities ranging from 25 k up to 100 k.

The Lattice Diamond™ software includes the Platform Designer tool, which is used to configure L-ASC10 and generate the control and interface logic for the Lattice FPGA. With the Platform Designer tool, the user can easily build solutions for voltage, current, temperature monitoring, fault-logging, hot-swap, trimming and margining, power supply sequencing, and supervision by adding L-ASC10 device(s) to a Lattice FPGA design. The tool also provides a mechanism to export the generated logic to be included in a new or existing design based on a Lattice FPGA. The export feature enables building systems using L-ASC10 devices with any of the Nexus FPGA devices supported by the Lattice Radiant™ software.

The Lattice Radiant software is used to integrate the exported logic into the top-level design of the Nexus FPGA device. The Nexus family of FPGAs provides the next level of centralized system management, offering features such as security, root of trust, and RISC-V soft CPUs using Lattice's Sentry™ Stack, and Propel™ tools. This document explains the procedure to add L-ASC10 devices to new or existing Nexus FPGA designs. It describes the Platform Designer tool settings and external hardware connections to construct a system incorporating power and thermal management with a Nexus FPGA. As an example, this document will be using the MachXO5-NX device, in most of the screen captures, as the Nexus FPGA. For details on adding L-ASC10 devices to new or existing designs based on other Lattice FPGAs, please refer to the documents listed in the Related Literature section.

# 2. Overview

## 2.1. L-ASC10 (Hardware Management Expander)

The L-ASC10 (Analog Sense and Control—10 Rails) is a Hardware Management (Power, Thermal, and Control Plane Management) Expander designed to be used with the Lattice FPGAs to implement the Hardware Management Control function in a circuit board. The L-ASC10 (referred to as ASC) enables seamless scaling of power supply voltage and current monitoring, temperature monitoring, sequence, and margin control channels. The ASC includes dedicated interfaces supporting the exchange of monitor signal status and output control signals with a central control FPGA. Up to eight ASC devices can be used to implement a hardware management system with a central control design.

The ASC provides three types of analog sense channels: voltage (nine standard channels and one high voltage channel), current (one standard voltage and one high voltage), and temperature (two external and one internal). The device incorporates nine general-purpose 5 V-tolerant open-drain digital input/output pins, four high-voltage charge-pumped outputs (HVOUT1–HVOUT4) for driving the gates of power MOSFETs, and four TRIM outputs for adjusting the output voltages of DC-DC converters.

The dedicated ASC Interface (ASC-I/F) is a reliable serial channel used to communicate with the Lattice FPGA in a scalable star topology. The centralized control algorithm in the FPGA monitors signal status and controls output behavior via this ASC-I/F. The ASC I2C interface is used by the FPGA for ASC background programming, interface configuration, and additional data transfer such as parameter measurement, I/O control, or status.

## 2.2. Nexus FPGAs

The Nexus product line includes the following FPGA families: MachXO5-NX, CrossLink-NX, Certus-NX, and CertusPro-NX. Any of these FPGAs can be used as the centralized hardware management controller that is capable of interfacing with up to eight L-ASC10 devices. The Nexus FPGA products have densities ranging from 25,000 to 100,000 Look-Up Tables (LUTs). The Nexus FPGA products include many of the following features: Embedded Block Ram (EBRs), Large Memory (LRAM), DSP Multipliers, UFM, PLLs, Oscillator, and ADC. The Nexus FPGA products are supported with the following Lattice software tools. Propel SDK, Propel Builder, and the Lattice Radiant software. These tools can be used to create secure systems based on Lattice's Sentry Solution Stack. Such systems provide a NIST SP8000-193-compliant Platform Firmware Resiliency (PFR) Root of Trust solution for authenticated and secure updates and platform management.

# 3. Adding L-ASC10 to Nexus FPGAs

The Platform Designer tool provides an integrated design environment that enables the following: configuring the analog settings of the ASC device(s), implementing the hardware management algorithm, and generating the programming files required to configure the devices on the circuit board. The Platform Designer software includes views, spread sheets, and graphical interfaces for the following:

- Global parameters (ASC options and device options)
- Analog parameters (current monitor settings, voltage monitor settings, temperature monitor settings, voltage trim settings, high voltage charge pump settings)
- System components (fan controller, fault logger, hot swap)
- Control functions (ports, nodes, and logic)

Before starting to build a system with a Nexus FPGA and ASC devices, there are some design constraints that need to be considered:

- The ASC uses 3.3 V logic for both the I2C and ASC-I/F (three-wire interface) to communicate with the FPGA (please see the System Connections section). These connections should be made on the Nexus device using the wide-range I/O with a Vcc that is powered by the same 3.3 V supply that is powering the ASC devices.
- The Full-Featured Fault Logger is not exportable from Platform Designer because it is based on hardware specific to MachXO2/3 and ECP FPGAs. However, the VIT Fault Logger is supported because the ASC device stores the fault information locally, which can later be retrieved using the I2C interface.
- The Nexus FPGA devices do not support the JTAG to I2C EFB that is used by the Diamond Programmer to erase, program, and verify the ASC devices. Thus, the PTM programming mode in Diamond Programmer is not available in Radiant Programmer. However, ASC programming is supported by the Nexus FPGA using an I2C port in at least three modes:
  a. Erase, program, and verify the ASC devices at the initial board test and bring-up based on the switch or jumper setting. Subsequent boot times are the fastest as the EECMOS memory of the ASC devices is already configured. This mode is similar to PTM programming using the Lattice Diamond Programmer on platforms based on MachXO2 or MachXO3 FPGAs.
  b. Program (or optionally verify) the ASC SRAM every boot. This adds about 15 ms per ASC at 100 kHz to program only (double that to verify). This supports in-the-field design updates. This mode is similar to platform designs based on the ECP5 FPGA.
  c. Erase, program, and verify the ASC devices when a field update is present. Otherwise, boot without configuring the ASCs. This is a new mode that is only supported using Nexus FPGAs. While the MachXO2/3 and ECP5 do support in-the-field updates, they only support configuring the ASC's SRAM over I2C every boot.
- The 8 MHz clock signal that comes from ASC0 should be connected to a primary clock pin of the Nexus FPGA (please see the System Connections section).
- Several software tools are required to build a Nexus-based platform design with ASC(s):
  a. Platform Designer Tool with the Diamond software to generate the ASC Interface and sequence RTL and the ASC programming file (ascx.hex).
  b. Propel Builder to create a RISC-V-based SOC.
  c. Propel SDK for firmware and driver support.
  d. Lattice Radiant software and Lattice Radiant Programmer to generate the programming file for the Nexus device and to program it.
  e. HEX Reader utility to convert the ASC programming file into a "C" type data array for configuring the ASC from the I2C Master IP.

# 4. Design Flow

Designing a platform management system based on a Nexus FPGA with ASC device(s) is not a linear step-by-step process since it involves multiple software tools. The design could start with either Platform Designer in the Lattice Diamond software by selecting the analog settings of the ASC and defining a power sequence, or it could start with the Lattice Propel Builder and the Lattice Radiant software to create an SOC that is ready to send debug messages over the UART. This design flow supports an iterative approach so that once all the parts are in place and working together, design changes can easily be made to the SOC, the firmware, the power sequence, or the ASC settings. The updated design is then processed in the Lattice Radiant software to support simulation, or programming evaluation boards, or programming the target system for validation. In the following sections, we will provide a brief description of the tools for building a system based on Nexus FPGAs and ASCs. For details on using the tools, the reader should consult the User Guide for the MachXO5-NX and L-ASC10 Platform Power Management Using RISC-V Demo (FPGA-UG-02194).

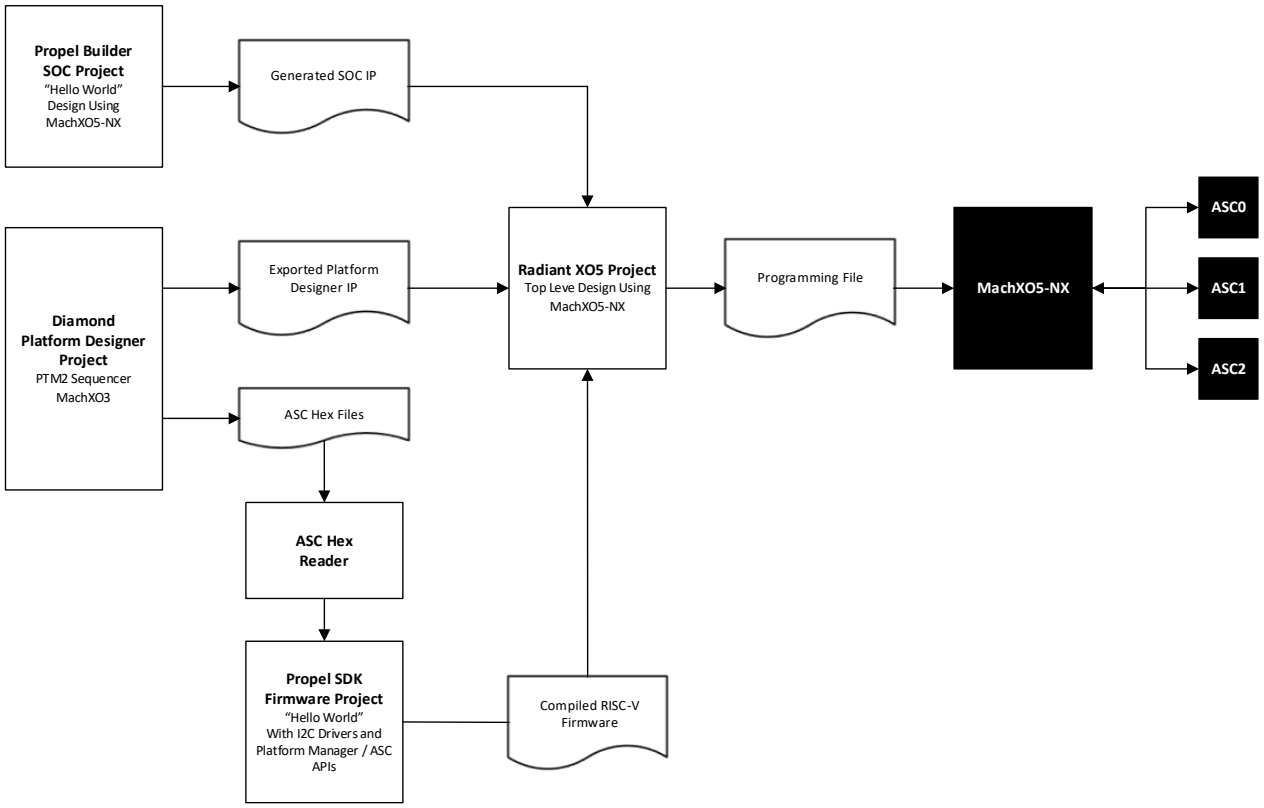


**Figure 4.1. Example Nexus System Design Flow with MachXO5-NX and (3) ASCs**

## 4.1. Platform Designer in Diamond

A new project can be started in the Lattice Diamond software based on any of the MachXO2 or MachXO3 that enables adding L-ASC10 devices, as illustrated in Figure 4.2. Smaller devices may only enable one or two L-ASC10 devices, whereas larger devices (6900 and above) will enable adding up to the maximum number (eight L-ASC10 devices).

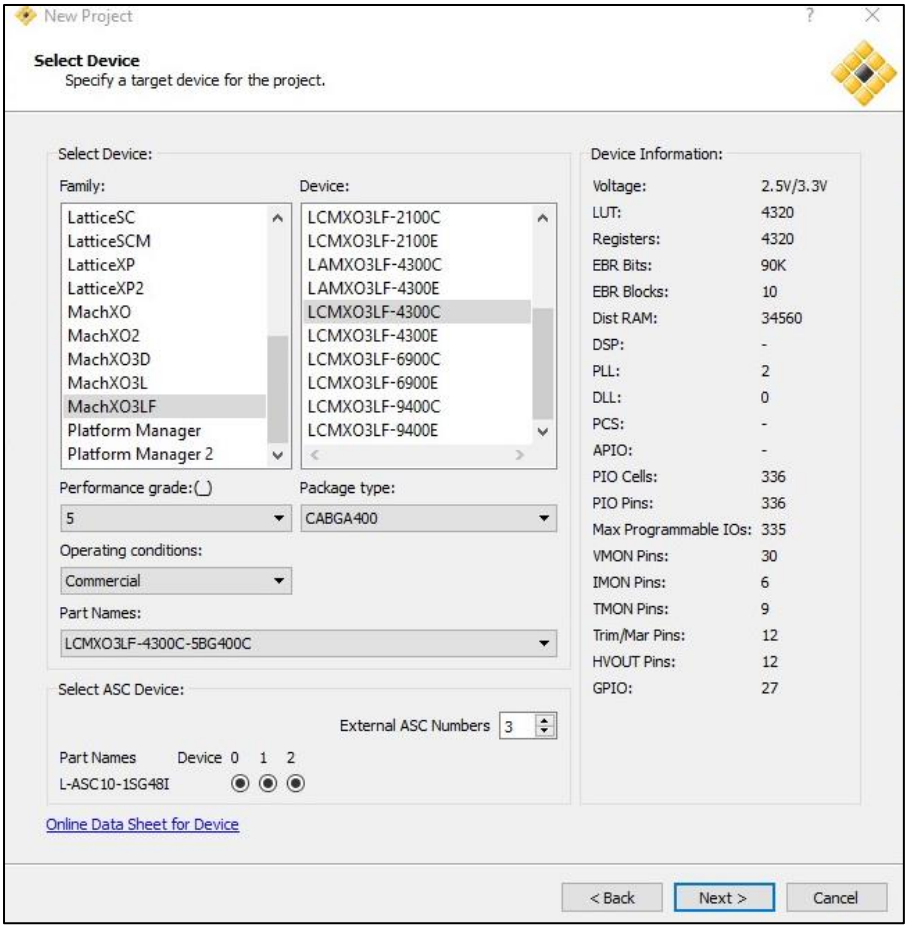


**Figure 4.2. Lattice Diamond – Platform Designer: New Project**

When the Platform Designer is shown, the tool needs to be configured to export the IP. This will disable synthesis, place and route, and generating a bit stream for the target device. This is why the target FPGA device is not critical.

In the **Global** view on the Device Options tab, check the **Export Platform Designer IP** box, fill in the folder path, and select Nexus from the drop-down list next to **Export IP To**, as shown in Figure 4.3.

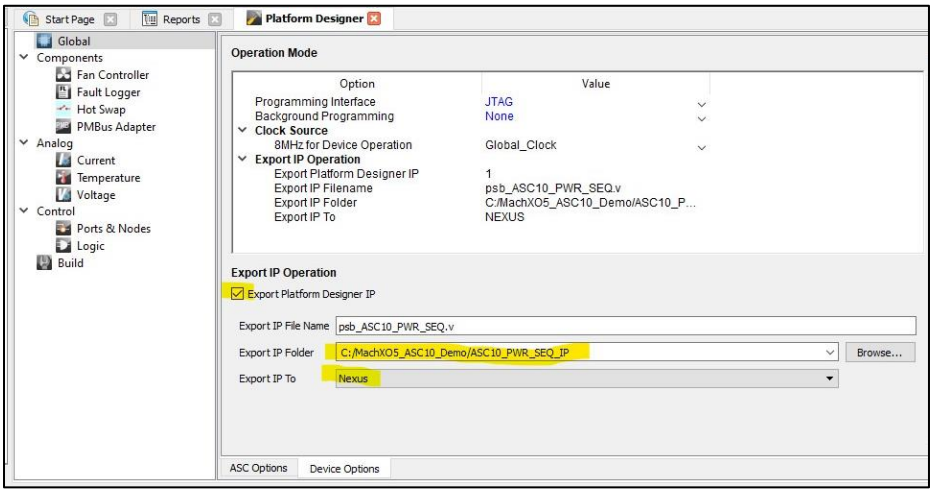


**Figure 4.3. Lattice Diamond – Platform Designer: Global Device Options**

Next is the **Logic** view on the **Inserted HDL** tab. Uncheck the **Enable EFB Module,** as shown in Figure 4.4. This is because the Nexus FPGA devices do not have the MachXO2/3-based embedded function block.

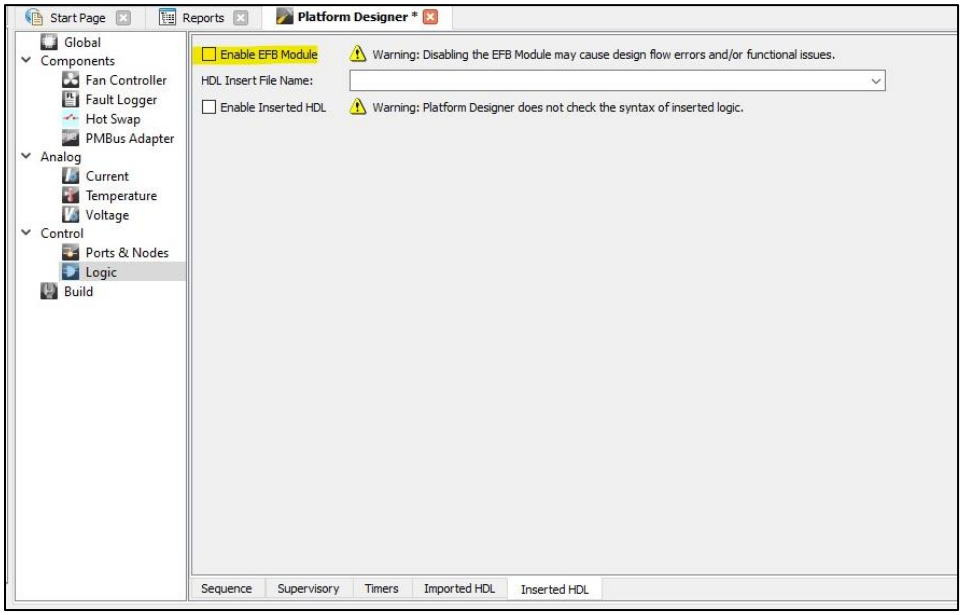


**Figure 4.4. Lattice Diamond – Platform Designer: Disable EFB Logic**

Now the Analog views and the Control views can be used just like any Platform Manager 2 project: configuring the Analog settings and defining the power-up and power-down sequences in the Logic view. For additional information on this, please see the Platform Designer User Guide.

Once the analog settings have been made and the logic sequence defined, the **Compile** button in the **Build** view (shown in Figure 4.5) can be used to generate the IP and ASC programming files. The IP is generated, and the files are written into the Export IP folder defined in Figure 4.3. Now the Platform Designer IP is ready to be imported into a Radiant project for the target Nexus FPGA device.

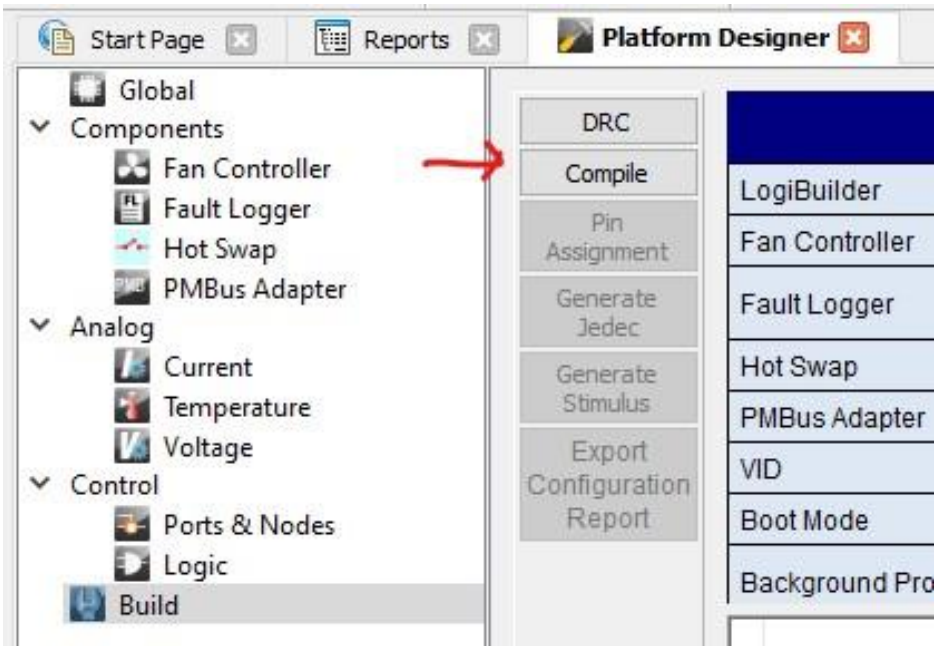


**Figure 4.5. Lattice Diamond – Platform Designer: Compiling the IP**

## 4.2. Propel Builder

The Lattice Propel Builder is used to assemble the system on a chip (SOC) based on the RISC-V soft processor. The GPIO module is used to “handshake” between the RISC-V firmware and the Platform Designer IP. This interface is fully customizable based on the needs of the system design. The other key peripheral in the SOC is the I$^2$C Master, which is used to configure the ASC device(s). Figure 4.6 shows an example of a SOC schematic that can be used with the Platform Designer IP and ASC device(s).

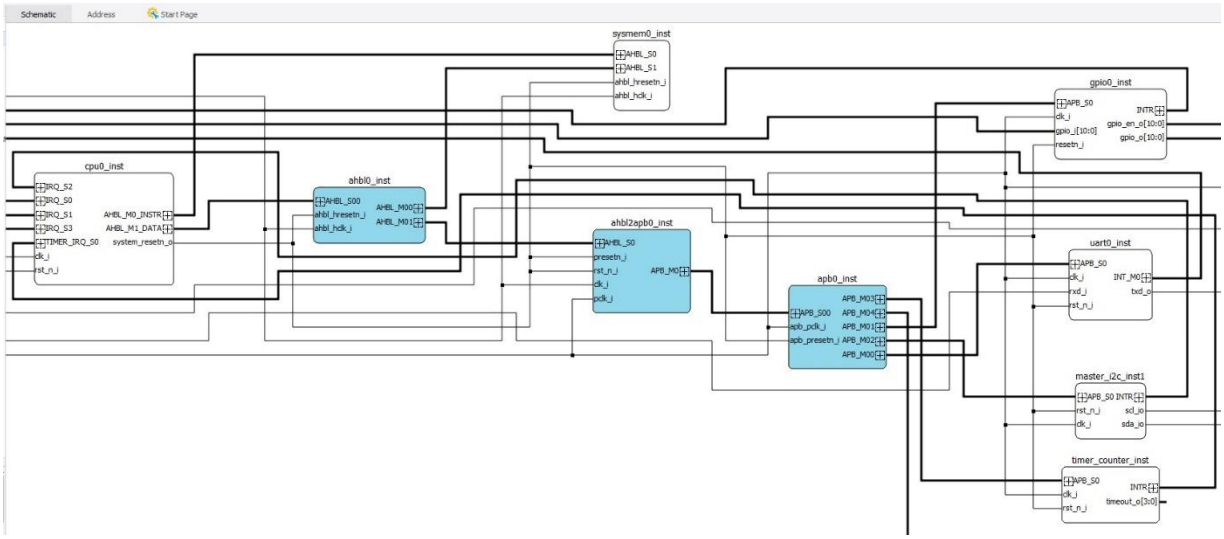


**Figure 4.6. Propel Builder – SOC Schematic Diagram with GPIO and I2C Master Peripherals**

The generate function in the Lattice Propel Builder will create all the RTL files needed to build the SOC on the Nexus FPGA fabric using the Lattice Radiant software.

## 4.3. Radiant FPGA Design Suite

In the Lattice Radiant software, a custom top-level module will have to be created by the designer. This top-level module will instantiate both the Platform Designer IP and the SOC IP and then connect the two together with wires for “handshaking” using the GPIO peripheral. The top module will also bring the ASC-I/F and I$^2$C signals to the I/O port list for mapping to specific pins.

When the Lattice Radiant project is first started, the exported Platform Designer IP files will need to be added. Below is a list of the files. Note that the first three files in the list contain the name of the Lattice Diamond project and will depend on the user’s name. The part of the filename that remains the same is the prefix *psb_* and suffix *_lgb*.

- psb_ASC10_PWR_SEQ_tmpl.v
- psb_ASC10_PWR_SEQ.v
- ASC10_PWR_SEQ_lgb.v
- RDAT.v
- WDAT.v
- CLKRST/src/rtl/verilog/clkrst.v
- CLKRST/src/rtl/verilog/clkrst_core.v
- ASCVM/src/rtl/verilog/ASCVM.v
- ASCVM/src/rtl/verilog/ASCVM_DATAPATH.v
- ASCVM/src/rtl/verilog/ASCVM_NX_RAM_DIST.v
- ASCVM/src/rtl/verilog/ASCVM_NX_RAM_EBR.v

Figure 4.7 shows the list of input files after the Platform Designer IP files have been added. The template file *psb_ASC10_PWR_SEQ_tmpl.v* has been *excluded* from the design, so it is grayed out. If the top-level ports of the Platform Design IP are modified, then the template file will reflect that change, and it can be used to copy and paste into the Lattice Radiant top-level design file.

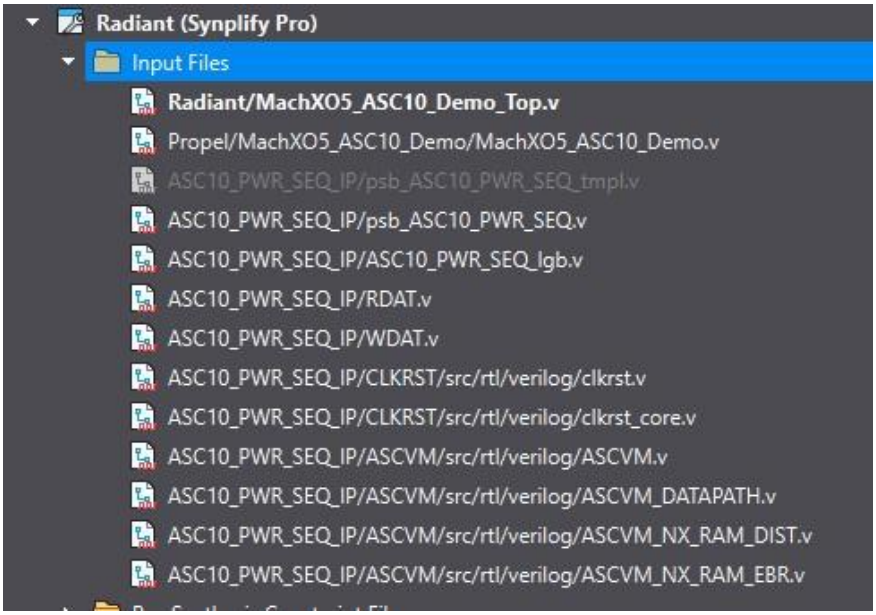


**Figure 4.7. Radiant – Input File List with Exported Platform Designer IP**

With the SOC and Platform Designer IPs combined into one Lattice Radiant project for the Nexus FPGA, the Lattice Propel SDK can be used to write and compile the RISC-V firmware.

## 4.4. Propel SDK

The Lattice Propel software development kit (SDK) supports ***C*** and ***C++*** programming languages and compiles the code for the soft-core RISC-V processor in the Nexus FPGA. Drivers for the GPIO and $I^2C$ peripherals are available from the MachXO5-NX and L-ASC10 Platform Power Management Using RISC-V Demo, as well as APIs to configure the ASC device(s) and read back the faults. Figure 4.8 shows the interactive development environment (IDE) of the Lattice Propel SDK.

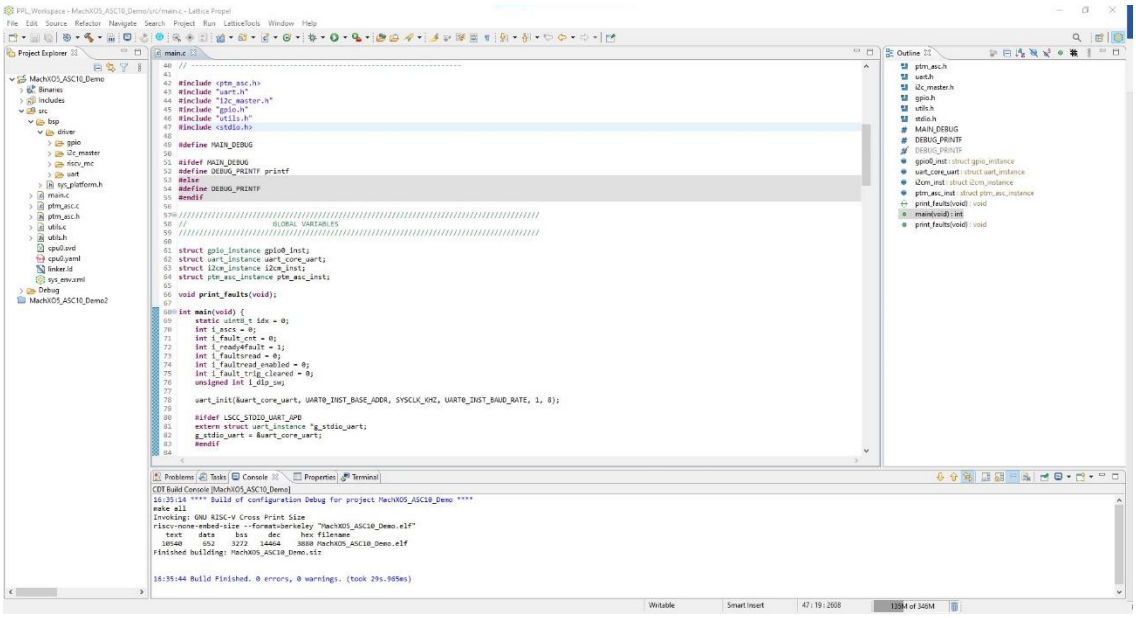

**Figure 4.8. Propel SDK – IDE main.c file**

## 4.5. ASC HEX Reader Utility

The last piece of the puzzle is to convert the ASC programming file into a ***C-type*** data array so the RISC-V firmware can configure the ASC device(s) over I²C. The ASC HEX Reader Utility is available in the MachXO5-NX and L-ASC10 Platform Power Management Using RISC-V Demo projects. The utility is run from a command prompt to set the name of the ***.h*** file and the array name, as shown in Figure 4.9.

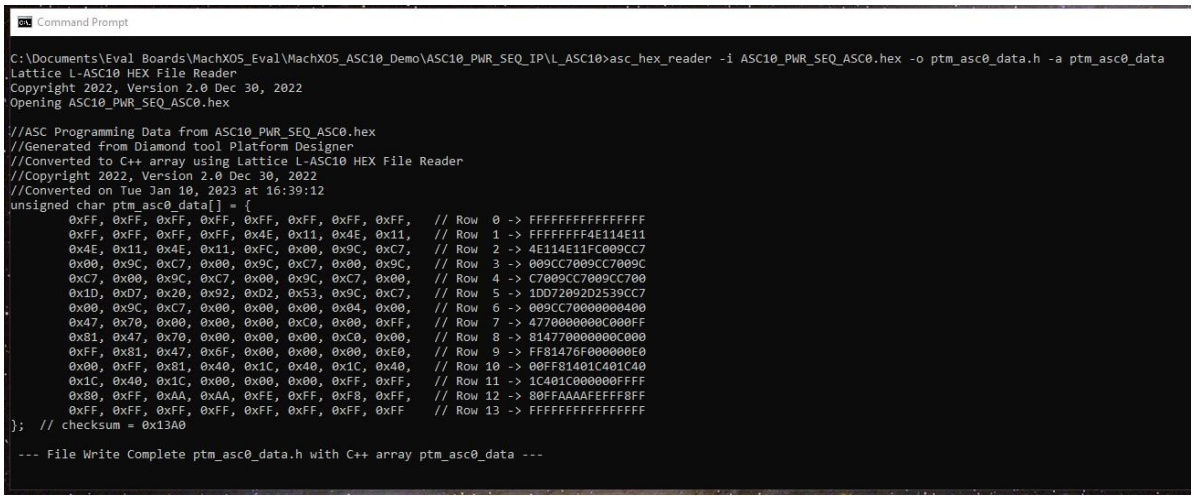


**Figure 4.9. Command Prompt – ASC Hex Reader Utility**

## 4.6. Number of L-ASC10 Devices with Nexus FPGAs

When an ASC is added to a Nexus FPGA, an interface for the ASC is instantiated inside the FPGA fabric and interconnected inside the FPGA fabric with the hardware (HDL) design. The interface is created using the logic resources of the Nexus FPGA. A minimum of 160 LUTs are required for each ASC. Since the smallest Nexus device has more than enough LUTs to support eight ASC interfaces, there is no restriction as to the number of ASC devices as there is for the MachXO2 and MachXO3 FPGAs.

In addition to LUT utilization, each ASC will require a set of PIOs for the external ASC-I/F connections, as listed in Table 4.1.

**Table 4.1. Additional PIOs Required After Instantiation of an ASC Interface**

| Signal Name (Connect to Nexus 3.3 V PIO) | Description |
|---|---|
| wrclk_x | Write Clock for ASC Interface |
| rdat_x | Read data signal for ASC Interface |
| wdat_x | Write data signal for ASC Interface |
| ASC0_RSTN[1] | Reset signal for all mandatory ASCs |
| ASC0_CLK[2,3] | 8-MHz Clock signal from ASC0 to Nexus |
| ASC_SDA | $I^2C$ Data |
| ASC_SCL | $I^2C$ Clock |

**Notes:**

1. All mandatory ASCx_RSTN should be combined to a single PIO pin; optional ASCx_RSTN require separate PIO pins.
2. Only ASC0 has an active clock.
3. Should be connected to a Clock input of Nexus FPGA.

Table 4.2 lists the minimum number of PIOs and LUTs associated with the number of ASC devices in a Nexus FPGA design.

**Table 4.2. Minimum Resources Needed per Additional ASC.**

| Number of ASCs | Minimum No. of Nexus PIOs | Minimum No. of LUTs[1] |
|---|---|---|
| 1 | 7 | 360 |
| 2 | 10 | 520 |
| 3 | 13 | 680 |
| 4 | 16 | 840 |
| 5 | 19 | 1000 |
| 6 | 22 | 1160 |
| 7 | 25 | 1320 |
| 8 | 28 | 1480 |

**Note:**

1. Number of LUTs associated with ASC Interface

# 5. System Connections – Nexus and L-ASC10

The external ASCs for a project can be designated as either mandatory or optional. The mandatory or optional designation determines the external connections between the Nexus FPGA and the ASC device(s). The mandatory or optional designation must be specified in the Platform Designer tool. Assign the ASC(s) to be mandatory or optional on the Platform Designer tool's main window, select the **Global** view, and choose the ASC Options tab. This tab displays the Reset Options field with the Reset Type for each ASC. By clicking the Reset Type, a drop-down list will present the mandatory and optional choices (with the exception of ASC0, which is always mandatory). When Optional is selected, the Reset Source is added to the design and shown in the view. Figure 5.1 shows the drop-down list for ASC1 and the reset source for ASC2 (ASC2_RSTN_I).

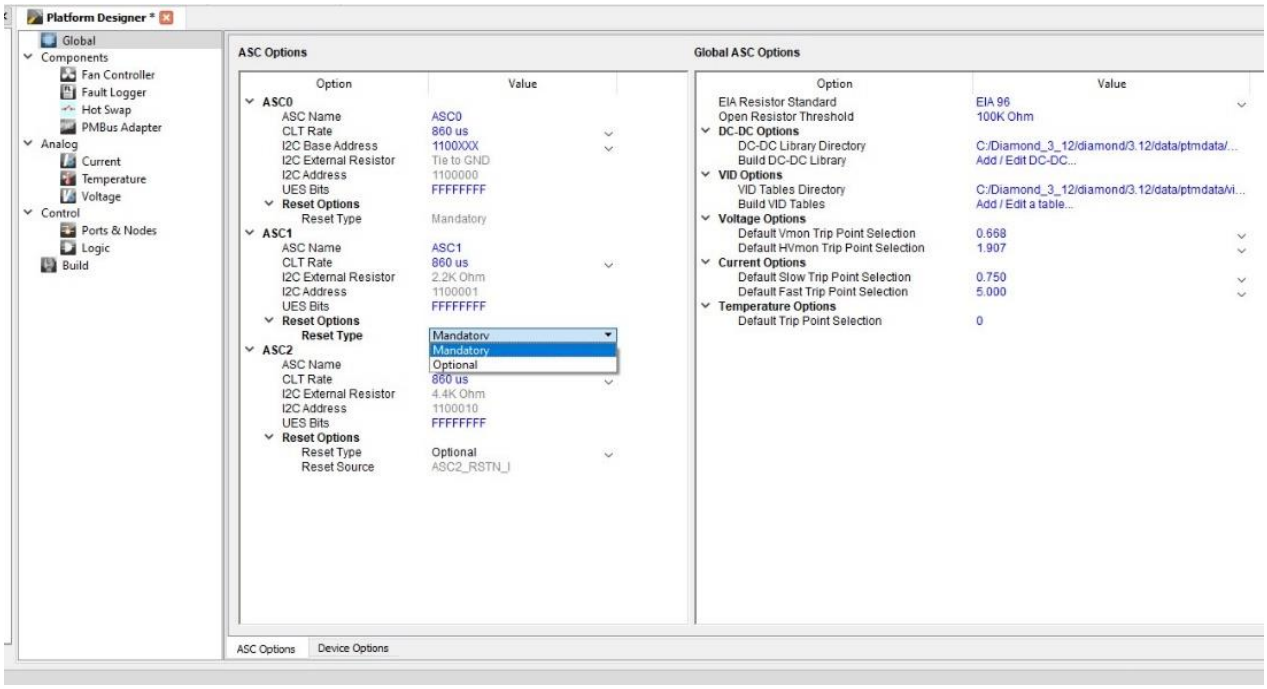


**Figure 5.1. Setting the ASC Reset Type in Platform Designer**

Figure 5.2 shows an example of an optional ASC2 connection with a focus on the reset signal, where ASC2 is part of a removable module. When the module is not installed, R1 holds the input signal ASC2_RSTN_I low. Thus, the logic associated with ASC2 within the Nexus FPGA is held in reset. When the module is installed, R2 will pull the reset signal high when ASC2 is out of reset.

If ASC2 was not part of a removable module (part of the main circuit board), then R2 should only be installed if ASC2 is installed.

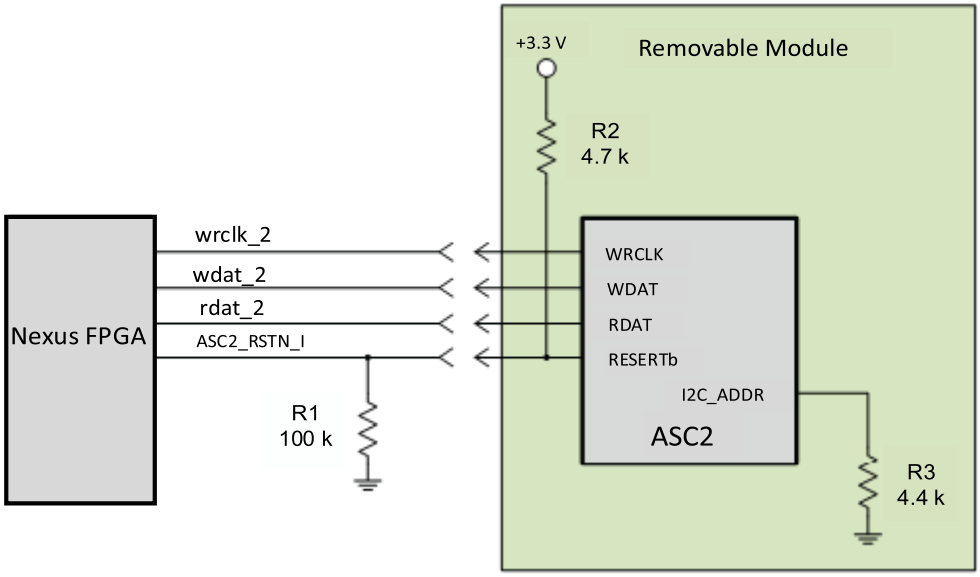


**Figure 5.2. Example Connections for Optional ASC2**

Table 5.1 provides a detailed description of the connections between the Nexus FPGA and the L-ASC10 devices.

**Table 5.1. Hardware Connection Between Nexus FPGA and L-ASC10**

| Signal / Pin | Mandatory ASC | Optional ASC |
|---|---|---|
| RESETb | The ASC0 RESETb and other mandatory ASC resets and ASCx RESETb pins must be connected to a PIOx on the Nexus FPGA externally.<br><br>This PIOx should be common to all the mandatory RESETb pins. It should be assigned to the ASC0_RSTN signal in the design software. Figure 5.3 shows the connection for mandatory ASC(s). | The RESETb pin of each optional ASC should be connected to a unique PIOx pin on the MachXO2 or MachXO3. Each reset signal is treated individually, so that only the registers associated with a particular optional ASC will reset when the reset input is driven low. The rest of the system, both mandatory and other optional ASCs, will continue to operate normally without interruption. Figure 5.4 shows the connection for the optional ASC(s). |
| ASCCLK | The ASC0 ASCCLK will be enabled in the software by default, and this pin must be connected to a Nexus FPGA primary clock input.<br><br>Other mandatory ASCs (ASC1, ASC2, etc.) ASCCLK pin should be *NO CONNECT*. Refer to Figure 5.3, which shows this connection.<br><br>An external 8 MHz clock source can be used as the system clock instead of the ASC0 ASCCLK. In this case, the ASCCLK output will be disabled, and the external clock should be connected to the primary clock input of the Nexus FPGA. The user must specify that an external clock source is being used in Platform Designer. | The ASCCLK pin for an optional ASC should be *NO CONNECT* and left open. Refer to Figure 5.3, which shows this connection. |
| ASC Interface | The ASC-I/F bus uses three signals: WRCLK, WDAT, and RDAT. The ASC-I/F bus operates at 8 MHz and includes error checking and reporting capabilities. The ASC-I/F pins on external ASC devices must be connected to three PIOx pins on the Nexus device. These three PIO pins are assigned using the design software. The design software automatically instantiates the interface for communicating with the ASC devices. Each ASC device requires its own unique ASC-I/F link, as shown in Figure 5.3 and Figure 5.4. The VCCA pin for an external ASC0 device must be connected to the VCCIO of the I/O bank used for the PIO assignment of WRCLK, WDAT, and RDAT. Care should be taken that the I/O bank used for the ASC-I/F link is not exposed to significant SSO noise, as this can degrade the performance of the analog monitors. See | |

| Signal / Pin | Mandatory ASC | Optional ASC |
|---|---|---|
| | L-ASC10 and Platform Manager 2 Hardware Checklist (FPGA-TN-02175) for more details. | |
| I²C (SCL/SDA) | The user MUST connect the SDA and SCL pins on all external ASC devices to PIO pins that are assigned as the SDA and SCL pins of an I²C master that is instantiated within the Nexus device. The I²C bus uses the SDA and SCL pins and operates from 100 to 400 kHz. External pull-up resistors to +3.3 V are required in all configurations. It is recommended to add an external RC noise filter to the SDA and SCL pins of the Nexus device. | |
| I²C Write Protect | To use the I²C write-protect feature in the design, the user MUST connect the ASCx GPIO1 pin to a unique PIOx pin on the Nexus device. Note that this feature is optional. | |

## 5.1. I²C Address Pin

Each ASC connected to a system is identified by a unique 7-bit address. The 3-bit LSB of the slave address for ASC(s) is set by connecting the I2C_ADDR pin to ground via a given resistor value. The seven states of the 3-bit LSB have a one-to-one correspondence with the ASC number designation in the platform management configuration. Table 5.2 shows the relationship between the resistor values and the three LSBs of the I²C address/ASC device number.

**Table 5.2. Hardware Connection Between Nexus FPGA and L-ASC10**

| Raddr Value[1] | 3 LSB of I²C Slave Address | ASC Device Number |
|---|---|---|
| None (Tie to GND) | 000 | 0 |
| 2.2 kΩ | 001 | 1 |
| 4.4 kΩ[2] | 010 | 2 |
| 7 kΩ[3] | 011 | 3 |
| 10 kΩ | 100 | 4 |
| 13.7 kΩ | 101 | 5 |
| 17.8 kΩ | 110 | 6 |
| None (Tie to VCCA) | 111 | 7 |

**Notes:**

1. All resistor values should be +/-1% tolerance or better.
2. For designs that utilize E-96 resistors a value of 4.42 kΩ can also be used.
3. For designs that utilize E-96 resistors a value of 7.15 kΩ can also be used.

## 5.2. Nexus FPGA PIO Pins

The PIO pins of the Nexus FPGA product line support various logic standards, both single-ended and differential. There are multiple PIO banks, and each bank has an associated set of VCCIO power supplies and ground pins. All PIOs in a given bank are referenced to the VCCIO for that bank. When configured as outputs, the PIO output standard must match the VCCIO for that bank (i.e., LVCMOS33 outputs must reside in a bank that has a VCCIO of 3.3 V). If a PIO is not programmed for use in the software, that pin defaults to an input pin with an internal pull-down resistor. Unused pins can be left floating or tied to GND. The voltage level and DC current at the FPGA PIO pin must not exceed the rated maximum for the specific bank (please refer to the respective Nexus FPGA data sheet). The FPGA PIOs that interface to the ASC must be LVCMOS33 and powered by 3.3 V. Figure 5.3 shows an example of three mandatory ASCs connected to the Nexus FPGA, and Figure 5.4 shows an example of one mandatory and two optional ASCs connected to the Nexus FPGA.

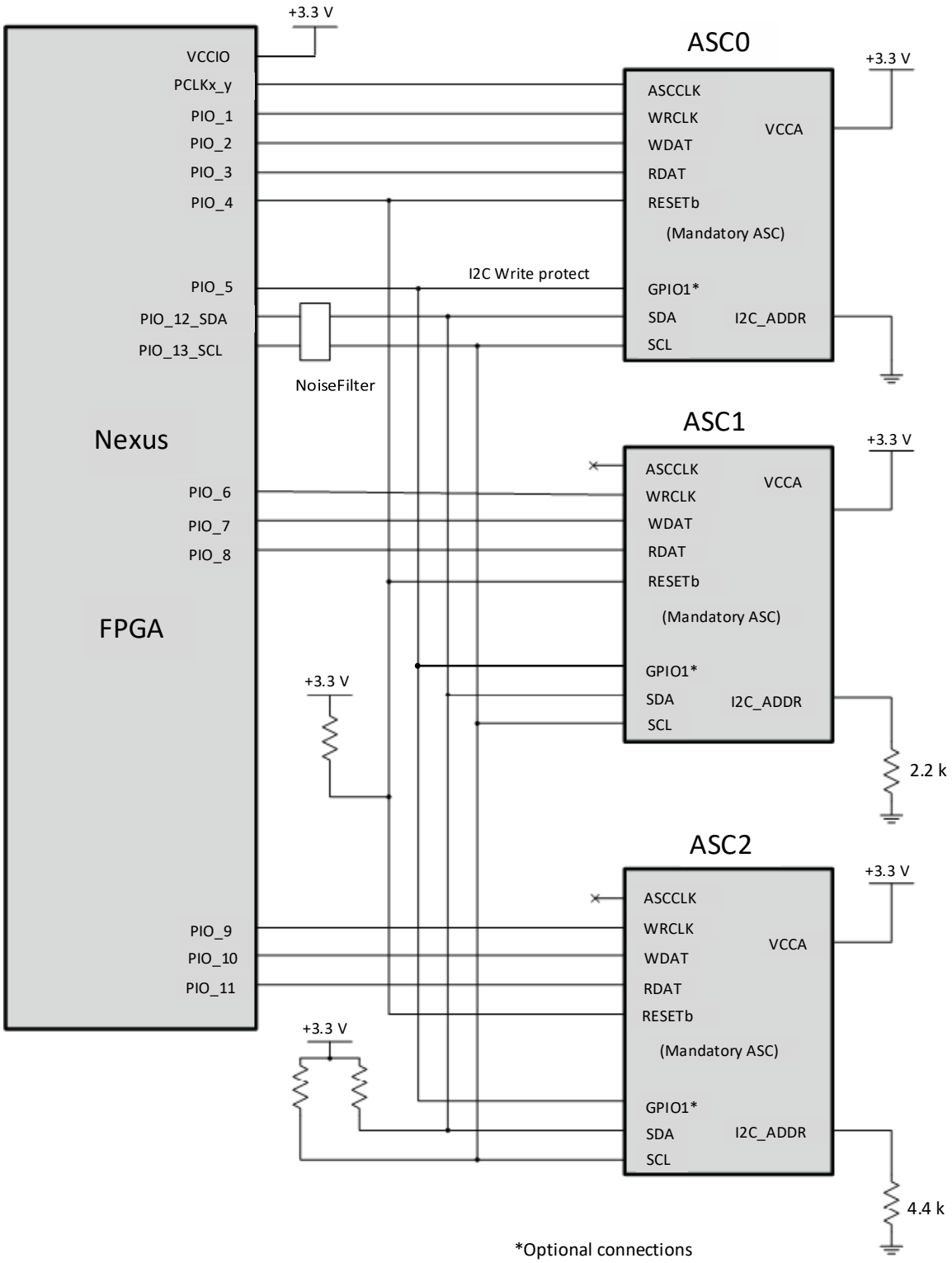


**Figure 5.3. Adding Mandatory ASC(s)**

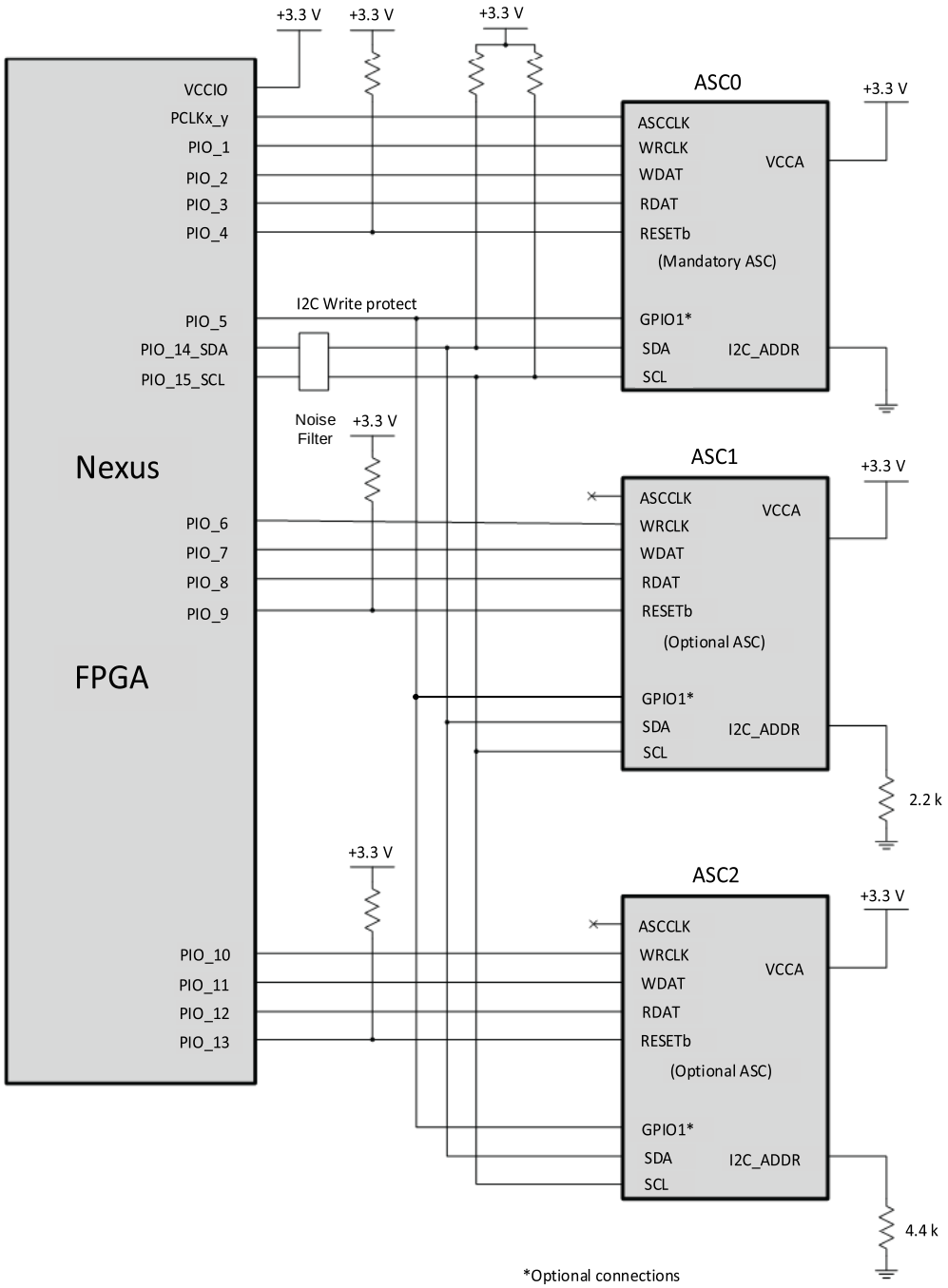


**Figure 5.4. Adding Optional and Mandatory ASC(s)**

# 6. Summary

This document has shown how to combine Nexus FPGAs (MachXO5-NX) and L-ASC10 together to provide the next level of platform management. Enabling designers to create scalable hardware systems with analog interfaces and a centralized digital control. The Platform Designer tool makes it easy to configure the L-AS10 device(s) and export the interface and control logic for inclusion in Nexus FPGA designs to build hardware systems for power and thermal management. The capability of the hardware system can be easily scaled up by adding more L-ASC10 devices to the Nexus FPGA. Using Lattice's Sentry Solution Stack, systems can provide a NIST SP8000-193-compliant Platform Firmware Resiliency (PFR) Root of Trust solution for authenticated and secure updates with centralized platform management. This document only provides a brief overview of how to include ASC device(s) into a Nexus FPGA design. It is recommended that the user read the related literature before designing an ASC-based system using any of the Nexus FPGAs. MachXO5-NX, CrossLink-NX, Certus-NX, and CertusPro-NX. For details and a step-by-step design description, please see the User Guide to the MachXO5-NX and L-ASC10 Platform Power Management Using RISC-V Demo (FPGA-UG-02194).

# References

- Lattice Nexus Platform web page
- CertusPro-NX web page
- Certus-NX web page
- CrossLink-NX web page
- Platform Manager 2 and L-ASC10 web page
- L-ASC10 In-System Programmable Hardware Management Expander Data Sheet (FPGA-DS-02038)
- MachXO5-NX Family Data Sheet (FPGA-DS-02102)
- CrossLink-NX Family Data Sheet (FPGA-DS-02049)
- Certus-NX Family (FPGA-DS-02078)
- CertusPro-NX Family (FPGA-DS-02086)
- L-ASC10 and Platform Manager 2 Hardware Checklist (FPGA-TN-02175)
- MachXO5-NX Hardware Checklist (FPGA-TN-02274)
- FPGA-UG-02194, MachXO5-NX and L-ASC10 Platform Power Management Using RISC-V Demo
- Dual Boot and Background Programming with Platform Manager (FPGA-TN-02078)
- Adding Scalable Power and Thermal Management to ECP5 (FPGA-AN-02019)
- Platform Designer User Guide
- Lattice Radiant FPGA design software
- Lattice Diamond FPGA design software
- Lattice Propel FGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

# Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

# Revision History

**Revision 1.0, January 2024**

| Section | Change Summary |
|---|---|
| All | Production release. |


www.latticesemi.com