Adding Scalable Power and Thermal Management to MachXO2 and MachXO3 Using L-ASC10

October 2019 Application Note FPGA-AN-02011

Introduction

Complex hardware systems require a large number of resources for sensing analog signals, programmable digital control and many GPIOs to provide interface with external components. It is desired to have a flexible system where above resources can be easily scaled as per application requirement. Lattice Platform Manager™ 2 family consists of Hardware Management Expander (L-ASC10), which in combination with hardware management controller, MachXO2™, MachXO3™, and ECP5™ FPGAs enable designers to create a flexible and scalable hardware system.

Lattice Diamond® software includes the Platform Designer tool which is used to configure L-ASC10 and MachXO2, MachXO3, or ECP5 FPGA. The tool also provides a mechanism to set up the interface between L-ASC10 and the controlling FPGA. With the help of 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 to a MachXO2, MachXO3, or ECP5 design.

This document explains the procedure to add L-ASC10 device to new or existing MachXO2 and MachXO3 designs. It describes the Platform Designer tool settings and external hardware connections to construct a system having power and thermal management using MachXO2 or MachXO3 and L-ASC10 devices. For details on adding L-ASC10 devices to new or existing ECP5 designs, please refer to AN6095, Adding Scalable Power and Thermal Management to ECP5 Using L-ASC10.

Overview

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 MachXO2 or MachXO3 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 MachXO2 or MachXO3. Up to eight ASC devices can be used to implement a hardware management system.

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) and four TRIM outputs for controlling the output voltages of DC-DC converters.

The dedicated ASC Interface (ASC-I/F) is a reliable serial channel used to communicate with a MachXO2 or MachXO3 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 I^2C interface is used by the FPGA for ASC background programming, interface configuration, and additional data transfer such as parameter measurement or I/O control or status.

MachXO2 and MachXO3 FPGAs

MachXO2 and MachXO3 hardware management controller are both ultra low power, instant-on and non-volatile FPGAs. The MachXO2 family has densities ranging from 256 to 6864 Look-Up Tables (LUTs) and the MachXO3 family has densities ranging from 640 to 9400 LUTs. Both families of devices also feature Embedded Block RAM (EBR), Distributed RAM, User Flash Memory (UFM), Phase Locked Loops (PLLs), pre-engineered source synchronous I/O support, advanced configuration support including dual-boot capability and hardened versions of com-

monly used functions such as SPI controller, I^2C controller and timer/counter. The architecture of both families of devices have several features such as programmable low swing differential I/Os and the ability to turn off I/O banks, on-chip PLLs and oscillators. Both the MachXO2 and MachXO3 families consist of different devices supporting either 3.3 V or 1.2 V as external VCC supply voltage.

Both the MachXO2 and MachXO3 family of devices offer enhanced I/O features such as drive strength control, slew rate control, PCI compatibility, bus-keeper latches, pull-up resistors, pull-down resistors, open drain outputs and hot socketing.

Both the MachXO2 and MachXO3 FPGAs also provide flexible, reliable and secure configuration from on-chip Flash memory. The FPGA can also configure themselves from external SPI Flash or be configured by an external master through the JTAG test access port or through the I^2C port.

Adding L-ASC10 to MachXO2 or MachXO3

The Platform Designer tool provides an integrated design environment that enables you to configure the MachXO2 or MachXO3 and the L-ASC10 device, implement the hardware management algorithm, simulate, assign pins, and finally generate the programming files required to configure the devices on the circuit board. Platform Designer software includes views, spread-sheet and graphical interfaces for the following:

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

Before starting the system building with MachXO2 or MachXO3 and L-ASC10 there some design requirements which need to be considered:

- MachXO2 and MachXO3 devices are available in various LUTs densities and operating supply voltages. However, the software supports only limited number of MachXO2 and MachXO3 devices for adding ASC(s) to them. The software GUI interface provides a section to add ASC(s) only if the selected device is enabled for adding ASC(s).
- Adding ASCs to MachXO2 or MachXO3 increases resource utilization and GPIOs required on the MachXO2 or MachXO3 device. If the existing hardware RTL design has resource utilization of around 80% of the FPGA, then it is recommended to shift the design to a higher density MachXO2 or MachXO3 device before adding ASC(s) to the design. Choosing higher density MachXO2 or MachXO3 will ensure there are sufficient resources available for adding ASC(s).

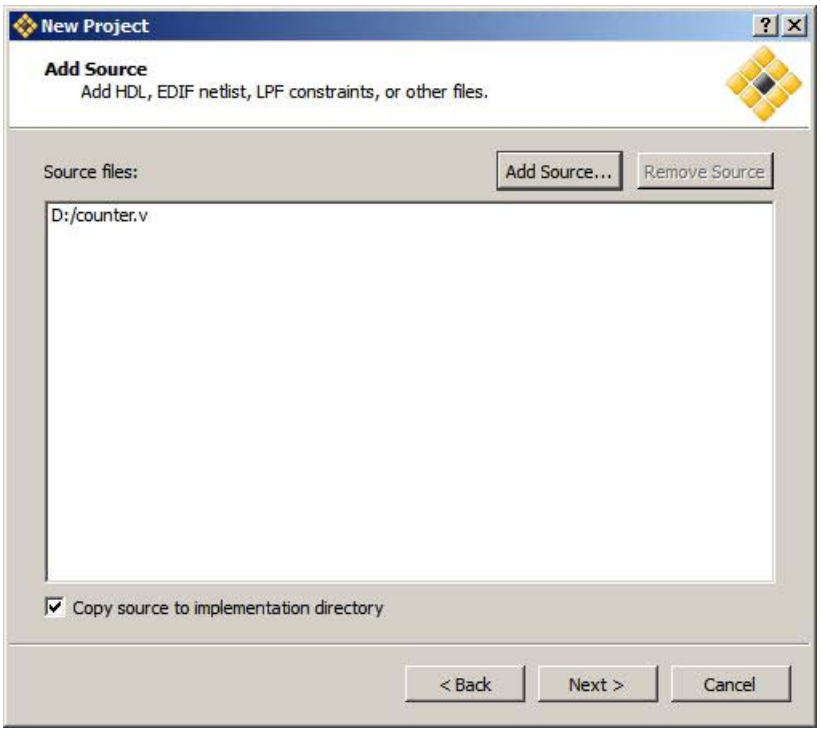
The procedure to add a single or multiple ASCs to the MachXO2 or MachXO3 FPGA using Platform Designer software tool is explained below.

1. Create a new Platform Manager 2 project.
   a. In the main Diamond window, from the File menu, choose **New > Project**.
   b. Click **Next** to open the Project Name dialog box.
   c. Click **Browse** to open the Browse for Folder dialog box and browse to the destination folder.
   d. Enter a Project Name.
   e. Click **Next** to open the Add Source dialog box.
2. Add the source files (HDL or other files) for the design. Diamond takes the source files and places them into the correct folders for the new project.
   a. Click **Add Source** to open a file browser.

b. Open the “source” folder, select all the files and click **Open** to add the files to the project.

c. Click **Next**. Figure 1 shows the window for adding the source file.

Figure 1. Add Source File Window



3. Select the device.

   a. Under Family, select **MachXO2** or **MachXO3**.

   b. Under Device, select the device based on the density required by application.

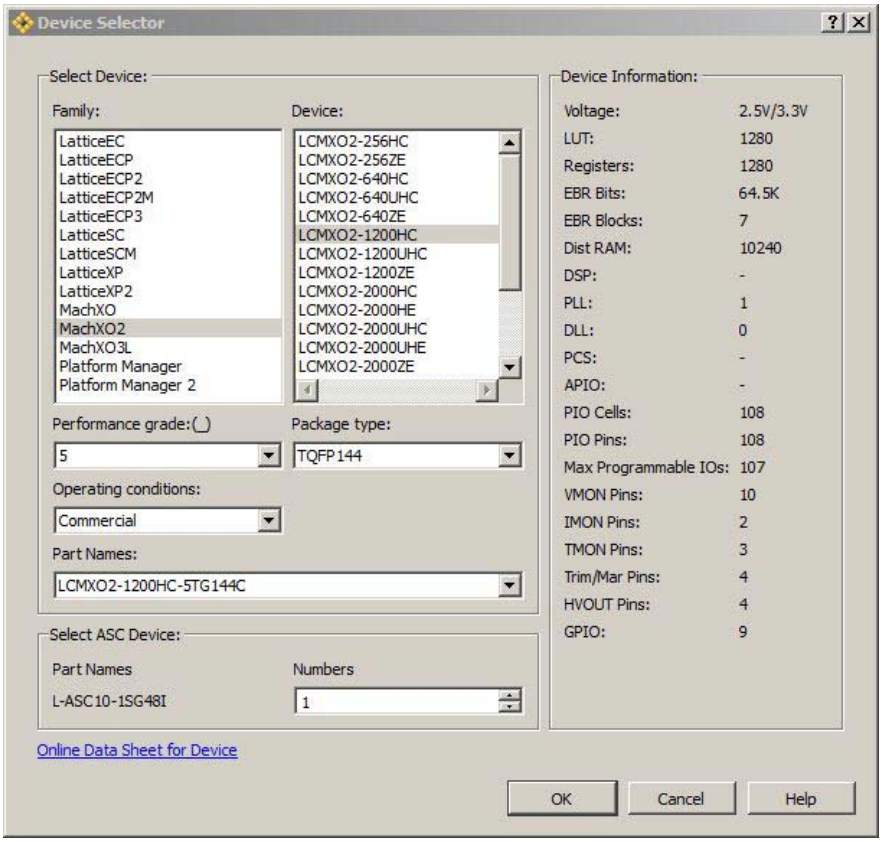
      If the selected MachXO2 or MachXO3 device is enabled for adding ASC, the **Select ASC Device** section is displayed at the bottom of the dialog box, as shown in Figure 2.

4. Select the ASC device.

   a. Under Numbers, click the increment /decrement button to set the number of ASC(s) required in the system. You can also enter a value in the Numbers box.

   b. The Device Information section displays resources available on the selected MachXO2 or MachXO3 and ASC(s) added to the design. Figure 2 shows an example of one ASC added to an LCMXO2-1200HC device.

Figure 2. Adding ASC to MachXO2 Device



c. Click **Next** to open the Select Synthesis Tool dialog box.

5. Select the synthesis tool.

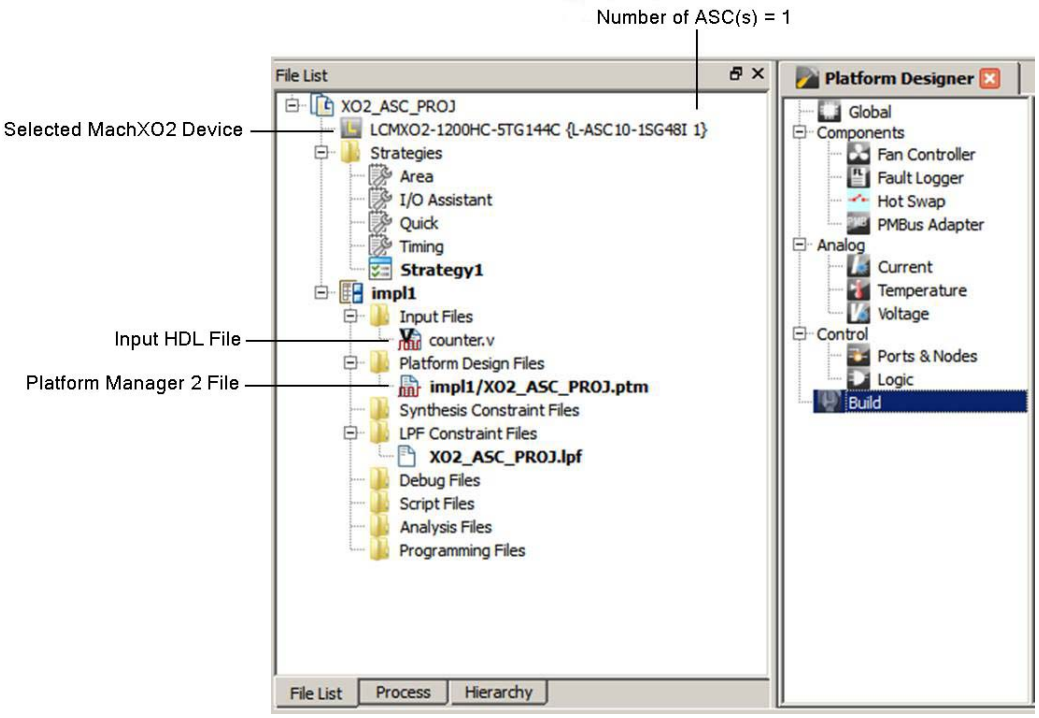
   a. Under Synthesis Tools, select **Synplify Pro®**.

   b. Click **Next** to open the Project Information summary.

6. Select the **Create Platform Builder File** box and enter the file name.
7. Generate the project by clicking **Finish**.

A new project with the selected MachXO2 or MachXO3 and L-ASC10 devices is created. The main window displays the <project name>.ptm file, the selected MachXO2 or MachXO3 device, and the number of ASC(s) used in the design, as shown in Figure 3.

Figure 3. MachXO2 and L-ASC10 Project



After constructing the *MachXO2+L-ASC10* or *MachXO3+L-ASC10* project, the designer can define settings for both the devices using Platform Designer tool. Platform Designer provides the interface for configuring the Power Management, Thermal Management, and Control Plane Functions of MachXO2 or MachXO3 and L-ASC10. The interface connections between MachXO2 or MachXO3 and ASC are assigned and managed using Diamond software and the Platform Designer tool. The logic design can be implemented using the integrated sequence and supervisory equation builder, or using imported HDL code, or both. Pin assignments for the *MachXO2+L-ASC10* or *MachXO3+L-ASC10* design can be done using Local Preference File (.lpf) or the Spreadsheet view from the Diamond Software. Refer to the Platform Designer User Guide for more information on the tool.

Appendix A. Adding L-ASC10 Device to a 4-Bit Counter Design on MachXO2 of this document illustrates an example of adding L-ASC10 to an existing MachXO2 design. It shows internal block level connections and external GPIOs of MachXO2 to be connected with L-ASC10 device.

Number of L-ASC10 Devices for MachXO2 or MachXO3

When an ASC is added to a MachXO2 or MachXO3 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 MachXO2 or MachXO3, therefore the number of ASCs that can be added is limited by the logic resources available in a particular MachXO2 or MachXO3 FPGA. A single ASC interface implementation uses approximately 160 LUTs, but the actual number depends upon the ASC control design.

The L-ASC10 device communicates with MachXO2 or MachXO3 FPGA over an ASC interface. Additional PIOs from the MachXO2 or MachXO3 device are required for the external connection between the L-ASC10 chip and the controlling FPGA. The necessary PIOs for external connection to an ASC interface are listed below. Table 1 lists signals required for a typical ASC0 to be connected to a PIO and Table 2 lists the maximum number of ASCs that can be added depending upon the MachXO2 or MachXO3 device density along with the minimum PIO connections.

Table 1. Additional GPIOs Required After Instantiation of an ASC Interface

Signal Name [Connect to a PIO of MachXO2 or MachXO3]	Description
wrclk_0	Write Clock for ASC interface
rdat_0	Read data signal for ASC interface
wdat_0	Write data signal for ASC interface
ASC0_RSTN[1]	Reset signal for ASC
ASC0_CLK[2, 3]	8-MHz Clock signal from ASC to MachXO2 or MachXO3

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 MachXO2 or MachXO3.

Table 2. Maximum Number of ASCs that can be Added to a MachXO2 or MachXO3 Device

Device	Maximum No. of ASCs that can be Added to a Given FPGA	Minimum No. of FPGA PIOs[1]
LCMXO2-640[2]	2	8
LCMXO2-1200[2]	4	14
LCMXO2-2000[2]	6	20
LCMXO2-4000[2]	8	26
LCMXO2-7000[2]	8	26
LCMXO3LF-640LF	2	8
LCMXO3LF-1300LF	4	14
LCMXO3LF-2100LF	6	20
LCMXO3LF-4300LF	8	26
LCMXO3LF-6900LF	8	26
LCMXO3LF-9400LF	8	26

1. Assuming all are mandatory ASCs.
2. Only MachXO2 types HC, HE, and UHC.

Enabling the HDL Design for *MachXO2+L-ASC10* or *MachXO3+L-ASC10* Project

Any new or existing HDL design added in the *MachXO2+L-ASC10* or *MachXO3+L-ASC10* project must be enabled for proper instantiation and operation within the project.

The Imported HDL feature in the Platform Designer tool is used to enable the HDL, map HDL ports to signals in Platform Designer and I/O signals in MachXO2, or MachXO3, or L-ASC10. The Imported HDL feature also supports parameter definition in Platform Designer. The procedure to import an existing HDL file using Platform Designer tool is described below.

To import an existing HDL design file:

1. Create a <project name>.ptm project with MachXO2 or MachXO3 and L-ASC10 as described in the Adding L-ASC10 to MachXO2 or MachXO3 section.
2. In the main window, select the **File List** tab under the left pane.
3. Right-click on **Input Files** and choose **Add > New File** or **Add > Existing File**.

   New File – A window opens for you to add a Verilog/VHDL source file. Browse to the location of the file and select the **Add to Implementation** check box.

   Existing File – A window opens for you to choose to search folders and select the Verilog/VHDL file.

4. Double-click on the **<project name>.ptm** design file. This opens the Platform Designer interface at the right-hand side.
5. Select **Logic** view.
6. Select the **Imported HDL** tab as shown in Figure 4.
7. Select the **Enable Imported HDL** check box.

   Platform Designer parses the HDL files selected in Step 3. An error is returned if there are no input files in the file list or if there are syntax errors in the file. Syntax errors are described in the HDL_parser.log file, as noted in the Error output window.

8. The *Port Declaration* section in the **Imported HDL** tab is used to assign signals from the Platform Designer signal pool to the top module input / output connections. The LOGIC SIGNAL POOL area is populated with the ASC signals (such as monitor alarms and output controls), any user generated nodes, and the FPGA PIO ports. The **Ports & Nodes** view can be used generate nodes and rename the ports shown in the LOGIC SIGNAL POOL.

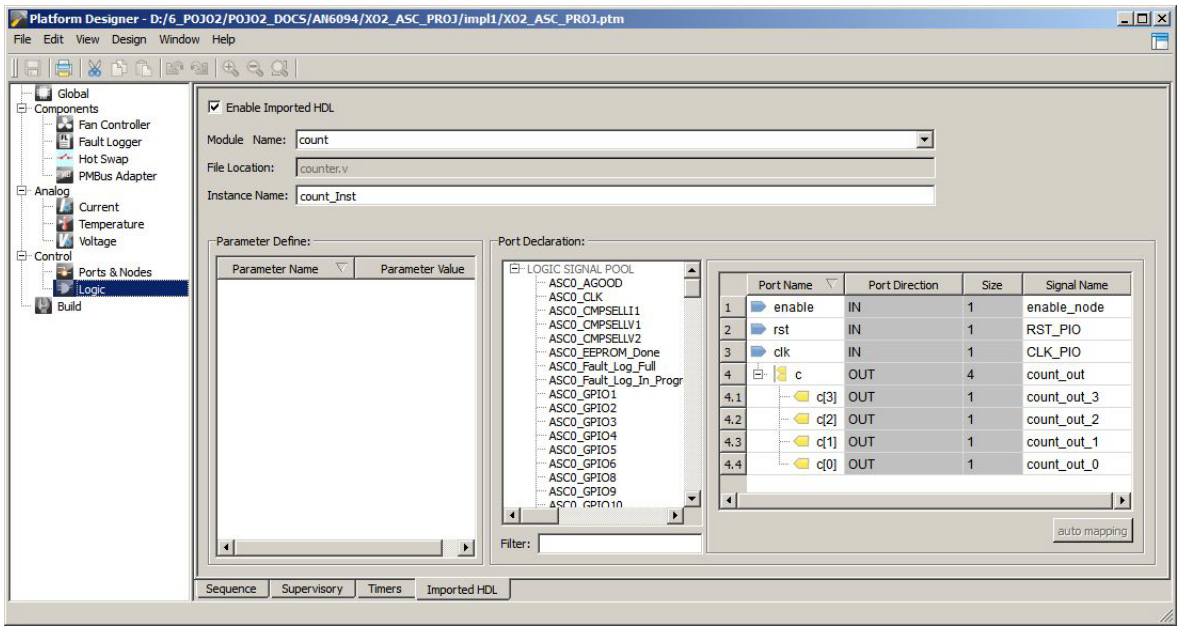
   As shown in Figure 4, the **Imported HDL** tab has *Port Name*, *Port Direction*, and *Size,* which are read only values defined by the HDL code. *Signal Name* is a drag and drop interface used to assign signals from the LOGIC SIGNAL POOL to the input and output connections of the imported HDL module.

   Clicking the **Auto Mapping** button on the **Imported HDL** tab is an alternative method for assigning connections from the LOGIC SIGNAL POOL to the top module. This button automatically renames top level PIO signals (shown in the Ports tab of **Ports & Nodes** view) to match the top module port names and sizes. These updated PIO signals are then assigned to the associated ports in the **Imported HDL** tab.

9. In the main window, select **Build** view to perform DRC, Compile, and Generate JEDEC.

For more details on Importing HDL files, refer to TN1287, Importing HDL Files with Platform Manager 2.

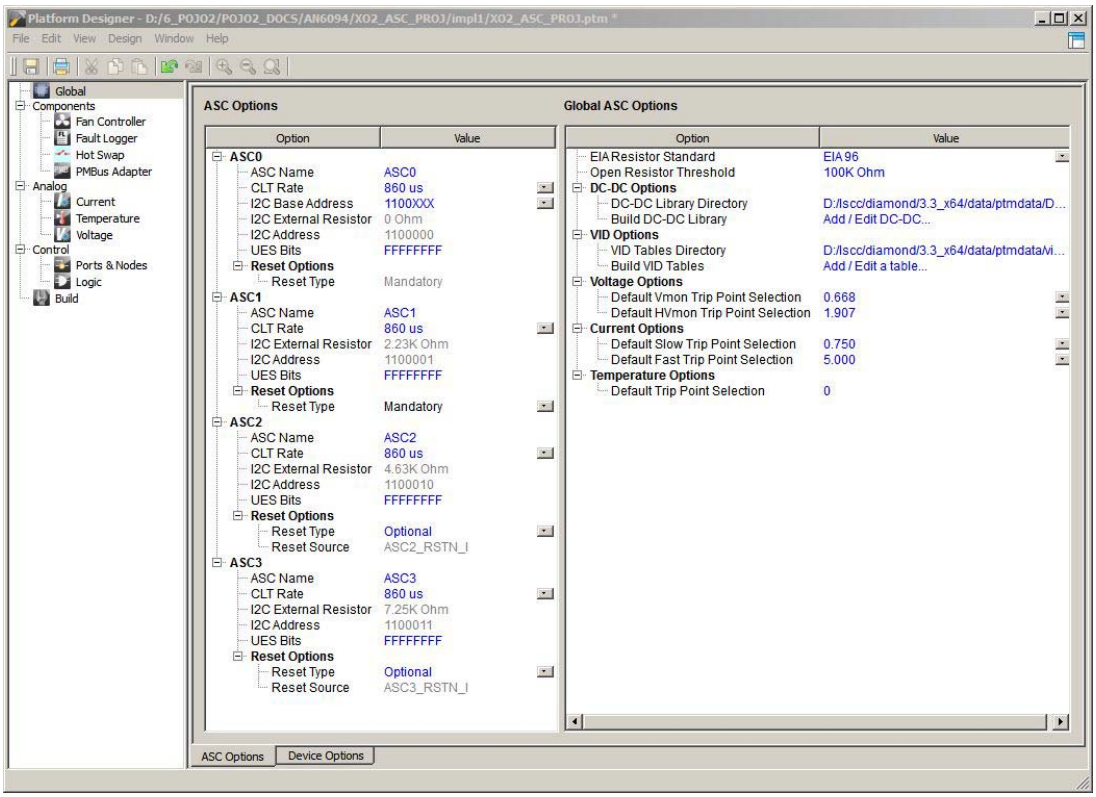
Figure 4. Imported HDL Tab in Logic View



System Connections - MachXO2 or MachXO3 and L-ASC10

The external ASCs for a project can be designated as either Mandatory or Optional. The Mandatory or Optional designation determines external hardware connections between MachXO2 or MachXO3 and L-ASC10. The Mandatory or Optional designation must be specified in the Platform Designer tool. For assigning ASC(s) to be Mandatory or Optional on the Platform Designer tool's main window, select **Global** view and choose the ASC Options tab. This tab displays Reset Options field with Reset Type for each ASC. By clicking the Reset Type drop-down list user can choose it to be either Mandatory or Optional. A Reset Source appears for each Optional Reset Type. The ASC0 is always Mandatory. Figure 5 shows an example setting where ASC0 and ASC1 are Mandatory and ASC2 and ASC3 are Optional.

Figure 5. Assigning Mandatory or Optional ASC(s)



The external ASCs for a project being designated as either Mandatory or Optional determines hardware connections between MachXO2 or MachXO3 and L-ASC10. The required connections include Clock, Reset, ASC Interface (ASC-I/F) and I^2C. Figure 6 and Figure 7 shows system level connections between MachXO2 or MachXO3 and L-ASC10. A Mandatory ASC is required to be present at system start-up but an Optional ASC is not required to be present at system start-up. Each of the connection requirements depending upon Mandatory or optional is described in the Table 3.

Table 3. Hardware connection between MachXO2 or MachXO3 and L-ASC10

Signal/Pin	Mandatory ASC	Optional ASC
RESETb	The ASC0 RESETb and other mandatory ASC resets, ASCx RESETb pins must be connected to a PIOx on MachXO2 or MachXO3 externally. 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 6 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 7 shows the connection for Optional ASC(s).
ASCCLK	The ASC0 ASCCLK will be enabled in the software by default and this pin must be connected to a MachXO2 or MachXO3 primary clock PCLKTx_y input. Other mandatory ASCs (ASC1, ASC2 etc.) ASCCLK pin should be NO CONNECT. Refer to Figure 6, which shows this connection. 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 PCLKTx_y pin on MachXO2 or MachXO3 and this clock should be common to all Mandatory ASCs. The user must specify that an external clock source is being used in software.	The ASCCLK pin for an Optional ASC should be NO CONNECT and left open. Refer Figure 7, 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 MachXO2 or MachXO3 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 6 and Figure 7. 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, 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 TN1225, Platform Manager 2 Hardware Checklist for more details.	
I^2C (SCL/SDA)	For MachXO2 or MachXO3 projects the user MUST connect the SDA and SCL pins on all external ASC devices to the SDA and SCL pins on the MachXO2 or MachXO3 device. The I^2C 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 on the SDA and SCL pins of MachXO2 or MachXO3.	
I^2C Write Protect	For using the I^2C write protect feature in the design, the user MUST connect the ASCx GPIO1 pin to a unique PIOx pin on the MachXO2 or MachXO3. Note that this feature is optional.	

I²C Address Pin

Each ASC connected in a system is identified by a unique 7-bit address. The 3-bit LSB of the slave address for ASC(s) are 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 4 shows the relationship between the resistor values and the three LSB of the I²C Address/ASC device number.

Table 4. Raddr Value vs. ASC Device Number

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

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.

FPGA PIO Pins

The PIO pins of the FPGA section support various logic standards, both single-ended and differential. There are four PIO banks and each bank has an associated set of VCCIO power supply 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. LVCMOS25 outputs must reside in a bank which has VCCIO of 2.5 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 at the FPGA PIO pin must not exceed 3.6 V. The average DC current drawn by FPGA PIO pins should not exceed 8 mA per pin. The FPGA PIOs that interface to the ASC must be LVCMOS33 and powered by 3.3 V.

Figure 6. Adding Mandatory ASC(s)

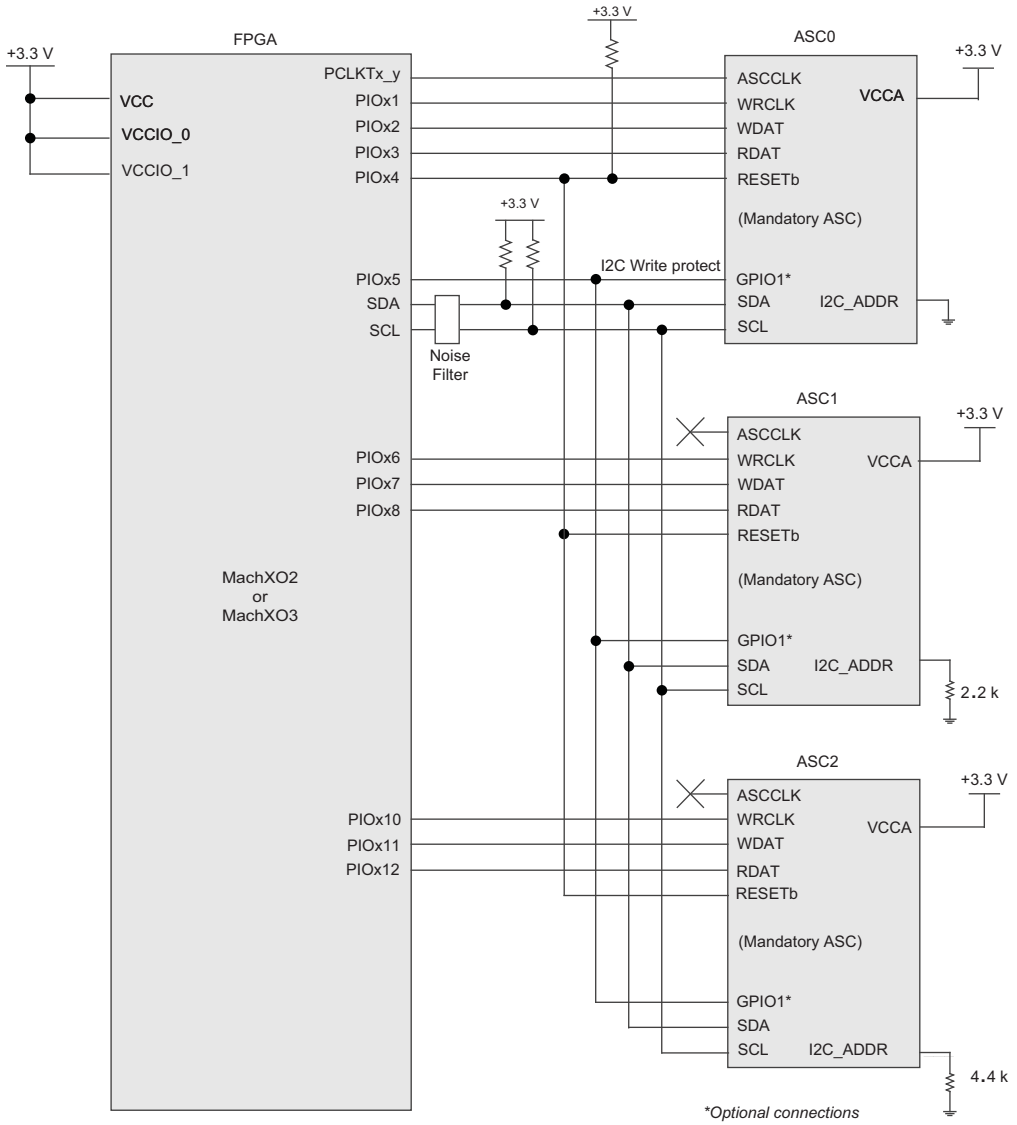
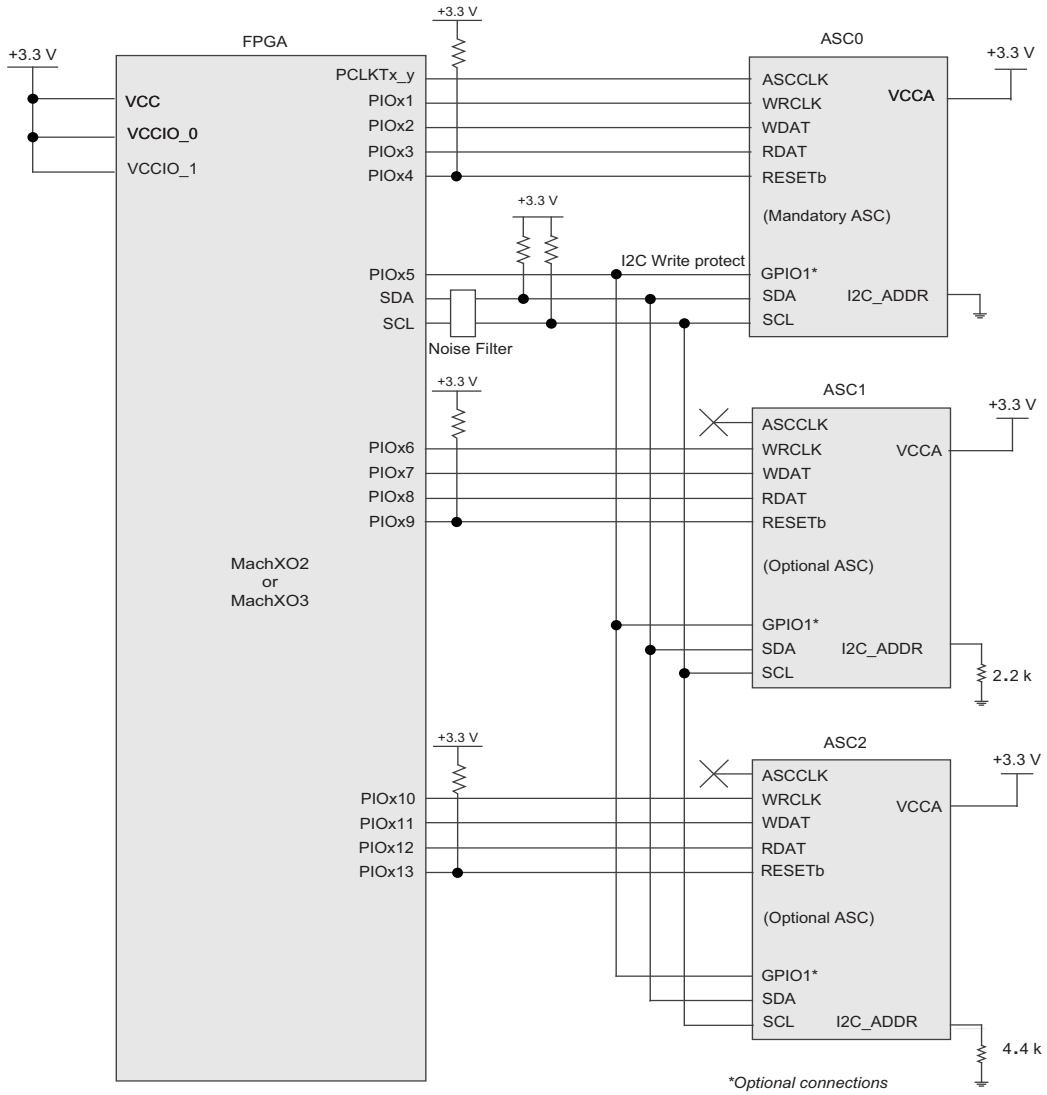


Figure 7. Adding Optional and Mandatory ASC(s)



Summary

MachXO2 or MachXO3 FPGA and L-ASC10 combined together enable designers to create a hardware system having analog interface with digital control. The Platform Designer tool has features to easily interface L-AS10 on MachXO2 or MachXO3 designs and build a hardware system for power and thermal management. The capability of the hardware system can be easily scaled up by adding more L-ASC10 devices to the MachXO2 or MachXO3 FPGA. However, the MachXO2 or MachXO3 device density and available GPIOs should be carefully considered and the system should be analyzed before adding L-ASC10. It is recommended to read the Related Literature before designing a system using MachXO2 or MachXO3 and L-ASC10.

Related Literature

- DS1042, L-ASC10 Data Sheet
- DS1035, MachX02 Family Data Sheet
- DS1047, MachX03 Family Data Sheet
- Platform Designer User Guide
- TN1225, Platform Manager 2 Hardware Checklist
- TN1287, Importing HDL Files with Platform Manager 2

Technical Support Assistance

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

Revision History

Date	Version	Change Summary
October 2019	1.2	Changed document number from AN6094 to FPGA-AN-02011
		Updated values and added footnotes to Table 4.
		Added Disclaimers section.
June 2017	1.1	Added MachXO3.
		Updated Technical Support Assistance information.
March 2015	1.0	Initial release.

Appendix A. Adding L-ASC10 Device to a 4-Bit Counter Design on MachXO2

This section describes an example of adding a single L-ASC10 to a 4-bit counter design on LCMXO2-1200HC device. The counter code is listed below.

```
// 4 bit counter with asynchronous reset with enable
module counter(count  , clk, rst, enable);
input clk, rst, enable;
output [3:0]count;
reg [3:0]count;
wire clk;
wire rst;
wire enable;
always @(posedge clk or posedge rst)
     begin
        if (rst)
           count = 4'b0000;
          else
       begin
           if(enable)
            count = count + 1;
               else
           count=count;
      end
       end
endmodule
```

Consider the following two cases. The first case is before adding L-ASC10 in parallel to a 4-bit counter design in LCMXO2-1200HC and the second is adding L-ASC10 to interface to the 4-bit counter design.

CASE A: Adding L-ASC10 in Parallel to the 4-Bit Counter in LCMXO2-1200HC

Design Description: This design starts with a simple 4-bit up-counter. The counter has an active high reset and active high enable signal. The count value starts incrementing only when enable is high. When enable is low, the count value remains at the last count. The count increments at every rising clock edge when enabled and not in reset. All the control signals for the counter are connected to PIOs on the LCMXO2-1200HC device.

Figure 8. Block Level Implementation and GPIO Connections of the 4-Bit Counter on LCMXO2-1200HC Fabric

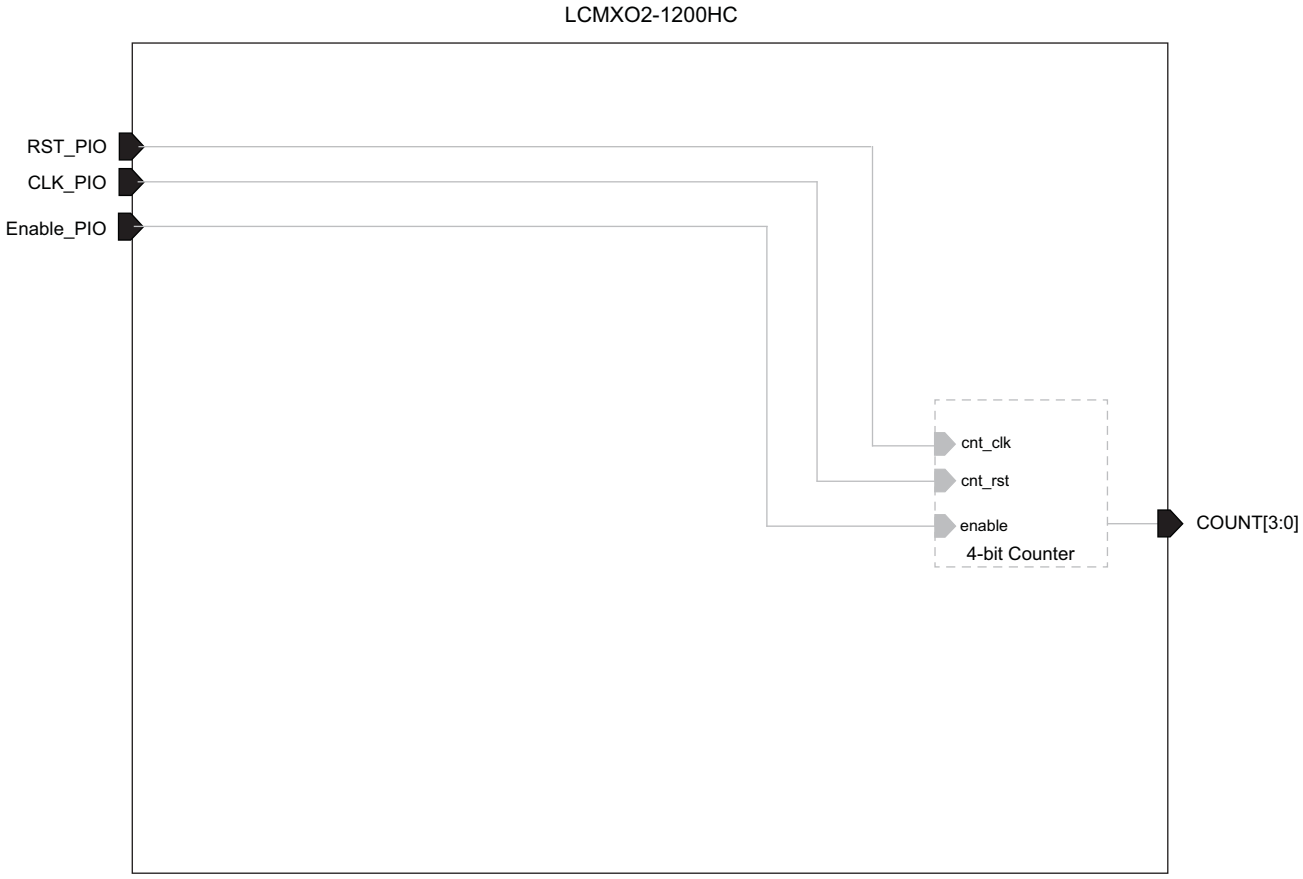


Table 5. Resource Utilization for the 4-Bit Counter Design on LCMXO2-1200HC Using Lattice Diamond Software

Resource Type	Consumption
LUT4s	4
EBR	0
PIOs	7
ASCs	0

Now the design is expanded to include an L-ASC10. The VMON1 voltage monitor on L-ASC10 is used to monitor 3.3 V supply. Using Platform Designer tool, the ASC interface and the 4-bit counter are instantiated in MachXO2. The ASC interface is designed with the control logic such that only when supply exceeds 3.36 V the ASC0_GPIO1 becomes high, else it remains low. The 4-bit counter design (HDL) and the ASC control logic work independently from each other.

Figure 9. Block Level Implementation with Internal Connections and GPIO for the 4-Bit Counter and ASC Interface on LCMXO2-1200HC Fabric.

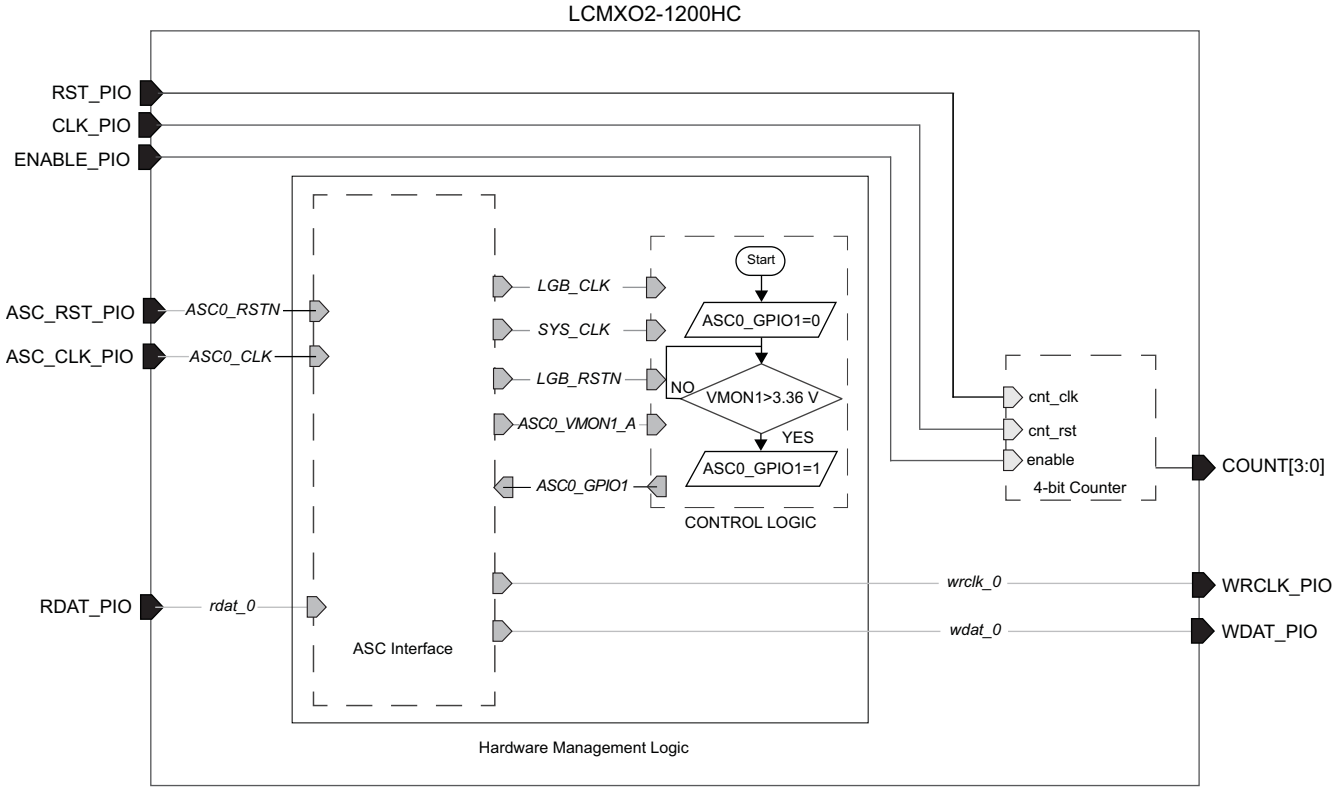


Table 6. Resource Utilization for the 4-Bit Counter Design with ASC Interface on LCMXO2-1200HC Using Platform Designer Tool of Lattice Diamond Software

Resource Type	Consumption
LUT4s	163
EBR	2
PIOs	12
ASC	1

CASE B: Adding L-ASC10 to Interface to the 4-Bit Counter (HDL) Design in LCMXO2-1200HC

This example highlights a situation where L-ASC10 is required to control functionality in the HDL. This design shows interconnection between L-ASC10 and the HDL design.

Design Description: Using the Platform Designer tool the ASC interface and the 4-bit counter are instantiated in MachXO2. In this system L-ASC10 Voltage Monitor is used to monitor 3.3 V supply. The ASC is designed with the control logic such that only when the supply exceeds 3.36 V it will enable the 4-bit counter. The external interface established between L-ASC10 and MachXO2 device manages data communication between both of them such that design objective is realized.

Figure 10. Block Level Implementation with Internal Connections and PIO for the 4-Bit Counter and the ASC Interface on LCMXO2-1200HC Fabric

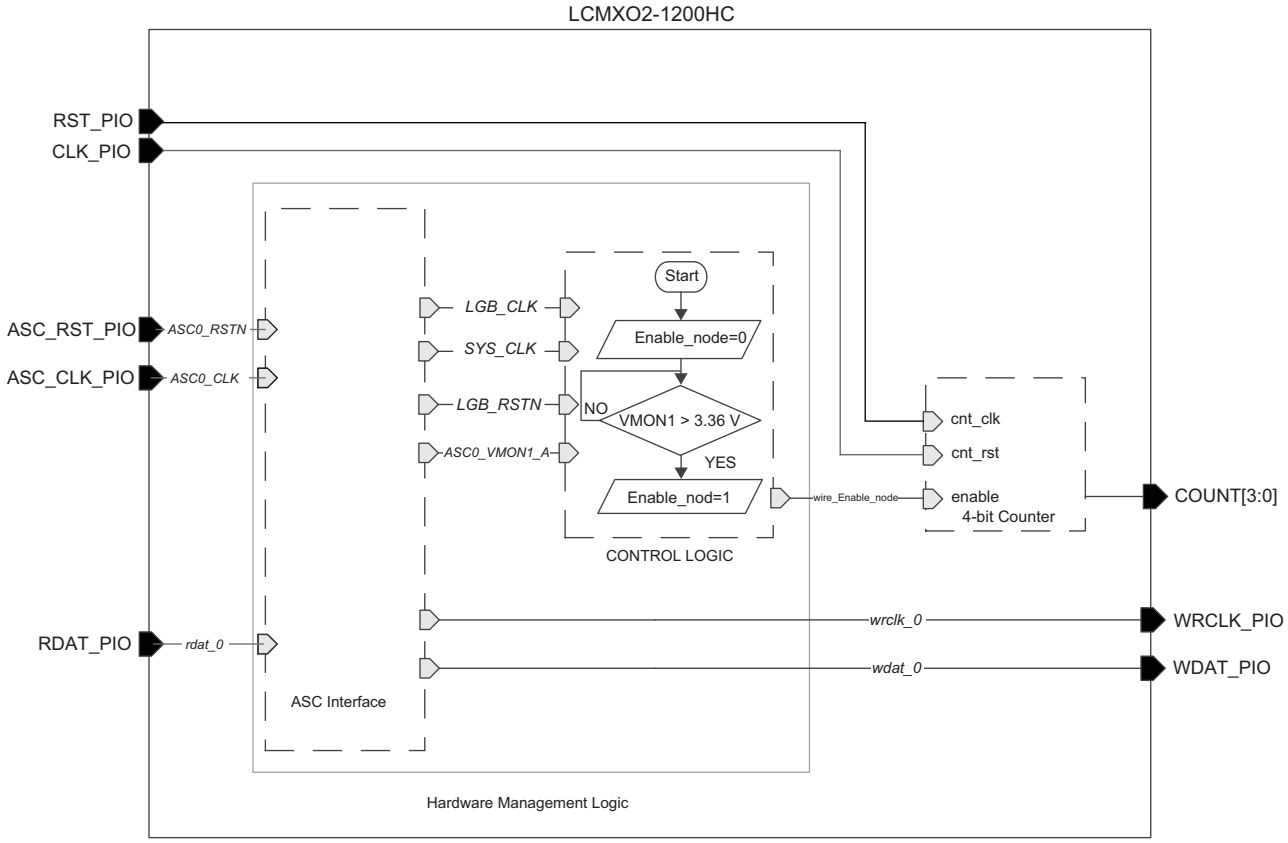


Table 7. Resource Utilization for the 4-bit Counter Design Interconnected with ASC Interface on LCMXO2-1200HC Using Platform Designer Tool of Lattice Diamond Software

Resource Type	Consumption
LUT4s	162
EBR	2
PIOs	11
ASC	1