UltraScale Architecture System Monitor

User Guide

UG580 (v1.10.1) September 15, 2021

XILINX

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/15/2021	1.10.1	Editorial updates only. No technical content updates.
08/25/2020	1.10	In External Analog Inputs, updated description of auxiliary analog inputs preconfiguration, and SYSMONE4 and auxiliary channels using Common-N. Added sentence about auxiliary analog inputs to last paragraph of Analog Inputs. Added IOSTANDARD setting for shared N side of a common N pin to Auxiliary Analog Inputs. Added note about supply rails to Power Supply Sensor. Updated number of registers in Configuration Registers (40h to 44h). Removed page address 24h from Table 3-17. Updated SYSMONE4 subheadings in Table 3-20. Added sentence explaining how supply sensor alarms are enabled to Supply Sensor Alarms. Added description of hysteresis and window modes to Over Temperature Automatic Shutdown.
02/25/2019	1.9.1	Updated design file location information in Example Design Instantiation.
03/29/2018	1.9	Revised Differences from Previous Generations and Thermal Management. Revised Table 1-2 and Table 4-5. Revised description for V_{REFN} in Table 3-1.
06/15/2017	1.8	Updated Figure 1-3. Revised the description for package pin V_{CC_PSADC} in Table 1-2. Updated Temperature Sensor, including deleting Equations 1-2 through 1-5. Revised Table 3-13. Updated Automatic Channel Sequencer, ADC Channel Settling Time (4Eh, 4Fh), ADC Channel Averaging (47h, 4Ah, and 4Bh), and Thermal Management.
12/20/2016	1.7	Revised last paragraph in SYSMON Attributes. Updated Temperature Sensor in Chapter 1. Added Figure 3-12 and Table 3-12. Added PMBus Examples. Revised Figure 1-2, Figure 2-5, Figure 2-6, Figure 3-1, Figure 3-2, Figure 3-7, Figure 3-17, and Figure 3-18. Added note for Table 3-13. Added important note for Table 3-14. Updated Table 1-1, Table 3-4, Table 3-5, Table 3-20, Table 4-1, Table 4-3, Table 4-8, Table 4-10, and Table 4-11. Updated Temperature Sensor in Chapter 2, including revising Equation 2-9 and Equation 2-10, and adding Equation 2-11 and Equation 2-12. Revised DRP I2C Interface. Deleted Table 3-18 SYSMONE1 I2C DRP Write Label Descriptions. Updated important notes in Continuous Sequence Mode (Slow Sequence - SYSMONE4). Revised important note in Example Design Test Bench.
05/26/2016	1.6	Updated dual sequence to slow sequence throughout. Updated Equation 1-3, Equation 2-9, Equation 2-10, Equation 4-1, Equation 4-2, and Equation 4-3. Updated Figure 3-2, Figure 3-7, Figure 4-1, and Figure 4-2. Updated Table 1-2, Table 1-3, Table 3-5, Table 4-2, Table 4-3, Table 4-5, and Table 4-6. Updated Default Mode, Continuous Sequence Mode (Slow Sequence - SYSMONE4), and Thermal Management. Standardized figure format.
11/24/2015	1.5	Added UltraScale+, Zynq UltraScale+, Virtex UltraScale+, and Kintex UltraScale+ FPGA information throughout. Added SYSMONE1 and SYSMONE4 information throughout.
07/11/2015	1.4	Updated last paragraph in SYSMON Overview. Updated Equation 1-2. Updated Temperature Sensor. Updated first paragraph in Chapter 3, SYSMON Register Interface. Updated equations in Thermal Management. Revised second paragraph in Anti-Alias Filters. Updated values for Temp upper alarm trigger, OT upper alarm limit, Temp lower alarm reset, and OT lower alarm reset in Verilog instantiation in Example Design Instantiation. Updated Example Design Test Bench.

Date	Version	Revision
02/20/2015	1.3	Updated Table 1-2 notes. Updated External Analog Inputs, Auxiliary Analog Inputs, I2C Slave Address Assignment, and Example Design Test Bench. Updated Temperature Sensor, page 28 to differentiate between using an external or on-chip reference. Updated Temperature Sensor, page 40 and Thermal Management, page 91. Updated Figure 2-6, Figure 4-3, and Figure 4-4.
09/19/2014	1.2	Updated first sentence in SYSMON Overview. Updated placement of ferrite bead in Figure 1-3, Figure 3-19, and Figure 5-1. Added Equation 2-7, Equation 2-8, Equation 2-14, Equation 2-15, Equation 2-17, Equation 2-18, Equation 2-20, and Equation 2-21. Removed timing information from Figure 3-3. Updated SYSMON DRP JTAG Write Operation. Updated first paragraph in I2C Read/Write Transfers. Updated Supply Sensor Alarms and Thermal Management. Added information on TCL file in Example Design Instantiation. Clarified Note 2 in Table 1-6.
07/17/2014	1.1	Updated SYSMON Overview, External Analog Inputs, Adjusting the Acquisition Settling Time, I2C Slave Address Assignment, External Multiplexer Operation, Reference Inputs (VREFP and VREFN), Anti-Alias Filters, and References. Updated Figure 1-3, Figure 3-5, Figure 3-8, Figure 3-18, and Figure 5-3. Added calibration coefficients and notes to Figure 3-1. For ports I2C_SDA_IN and I2C_SCLK_IN, corrected port name by removing _IN. Updated Table 1-2, Table 1-4, Table 3-2, Table 3-5, and Table 3-9. Added IBUF_ANALOG. Removed references to A_{VCC}, A_{VTT}, and $MGTV_{CCAUX}$. Added I2C Addr Meas and Reserved status registers to Table 3-1. Updated note in DRP JTAG Interface. Updated first paragraph in Chapter 4, SYSMON Operating Modes. Updated SYSMON Verilog example design in Example Design Instantiation.
12/10/2013	1.0	Initial Xilinx release.

Table of Contents

Revision History 2

Chapter 1: Overview and Quick Start

Introduction to the UltraScale Architecture 6
SYSMON Overview 7
SYSMON Pinout Requirements 12
Instantiating the SYSMON 18

Chapter 2: Basic Functionality

ADC Transfer Functions 29
Analog Inputs 32

Chapter 3: SYSMON Register Interface

Dynamic Reconfiguration Port (DRP) Timing 47
Status Registers 48
Control Registers 54
DRP Arbitration 59
DRP JTAG Interface 59
DRP I2C Interface 64

Chapter 4: SYSMON Operating Modes

Single Channel Mode 80
Automatic Channel Sequencer 80
Sequencer Modes 86
External Multiplexer Mode 88
Automatic Alarms 90

Chapter 5: Application Guidelines

Reference Inputs (VREFP and VREFN) 95
Analog Power Supply and Ground 96
External Analog Inputs 99
SYSMON Software Support 100

Appendix A: Additional Resources and Legal Notices

Xilinx Resources 111
Solution Centers 111
Documentation Navigator and Design Hubs 111
References 112
Please Read: Important Legal Notices 113

Overview and Quick Start

Introduction to the UltraScale Architecture

The Xilinx® UltraScale™ architecture is the first ASIC-class architecture to enable multi-hundred gigabit-per-second levels of system performance with smart processing, while efficiently routing and processing data on-chip. UltraScale architecture-based devices address a vast spectrum of high-bandwidth, high-utilization system requirements by using industry-leading technical innovations, including next-generation routing, ASIC-like clocking, 3D-on-3D ICs, multiprocessor SoC (MPSoC) technologies, and new power reduction features. The devices share many building blocks, providing scalability across process nodes and product families to leverage system-level investment across platforms.

Virtex® UltraScale+™ devices provide the highest performance and integration capabilities in a FinFET node, including both the highest serial I/O and signal processing bandwidth, as well as the highest on-chip memory density. As the industry's most capable FPGA family, the Virtex UltraScale+ devices are ideal for applications including 1+Tb/s networking and data center and fully integrated radar/early-warning systems.

Virtex UltraScale devices provide the greatest performance and integration at 20 nm, including serial I/O bandwidth and logic capacity. As the industry's only high-end FPGA at the 20 nm process node, this family is ideal for applications including 400G networking, large scale ASIC prototyping, and emulation.

Kintex® UltraScale+ devices provide the best price/performance/watt balance in a FinFET node, delivering the most cost-effective solution for high-end capabilities, including transceiver and memory interface line rates as well as 100G connectivity cores. Our newest mid-range family is ideal for both packet processing and DSP-intensive functions and is well suited for applications including wireless MIMO technology, Nx100G networking, and data center.

Artix® UltraScale+ devices provide high serial bandwidth and signal compute density in a cost-optimized device for critical networking applications, vision and video processing, and secured connectivity. Coupled with the innovative InFO packaging, which provides excellent thermal and power distribution, Artix UltraScale+ devices are perfectly suited to applications requiring high compute density in a small footprint.

Kintex UltraScale devices provide the best price/performance/watt at 20 nm and include the highest signal processing bandwidth in a mid-range device, next-generation

transceivers, and low-cost packaging for an optimum blend of capability and cost-effectiveness. The family is ideal for packet processing in 100G networking and data centers applications as well as DSP-intensive processing needed in next-generation medical imaging, 8k4k video, and heterogeneous wireless infrastructure.

Zynq® UltraScale+ MPSoC devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing. Integrating an Arm®-based system for advanced analytics and on-chip programmable logic for task acceleration creates unlimited possibilities for applications including 5G Wireless, next generation ADAS, and Industrial Internet-of-Things.

This chapter provides a brief overview of the SYSMON functionality with key information to allow a basic understanding of the SYSMON block. This introduction describes the pinout requirements and how to instantiate basic functionality in designs. Subsequent chapters provide more detailed descriptions of the SYSMON functionality.

This user guide describes the UltraScale architecture system monitor and is part of the UltraScale architecture documentation suite available at: www.xilinx.com/documentation.

SYSMON Overview

The SYSMON includes an analog-to-digital converter (ADC) as well as on-chip sensors that can be used to sample external voltages and monitor on-die operating conditions, such as temperature and supply voltage levels. The ADC and sensors are fully tested and specified (see *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 7], *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922) [Ref 7], *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893) [Ref 7], *Virtex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS923) [Ref 7], and *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925) [Ref 7]). The ADC supports differential sampling of unipolar and bipolar analog input signals (see Chapter 2, Basic Functionality) and provides a wide range of operating modes to convert up to 17 external analog input channels (see Chapter 4, SYSMON Operating Modes). Status registers store the ADC's sampled data, which can be accessed:

- Directly through the dynamic reconfiguration port (DRP) ports
- Through an external JTAG interface
- Through an I2C interface
- Through the power management bus (PMBus) for UltraScale+ devices
- Through the advanced peripheral bus (APB) for Zynq UltraScale+ MPSoC devices

The SYSMON interface can be simplified to drive a series of alarm ports based on user-defined operating conditions, such as die temperature levels or power supply thresholds. Figure 1-1 shows a block diagram of the SYSMON (SYSMONE1 for UltraScale and SYSMONE4 for UltraScale+ devices).

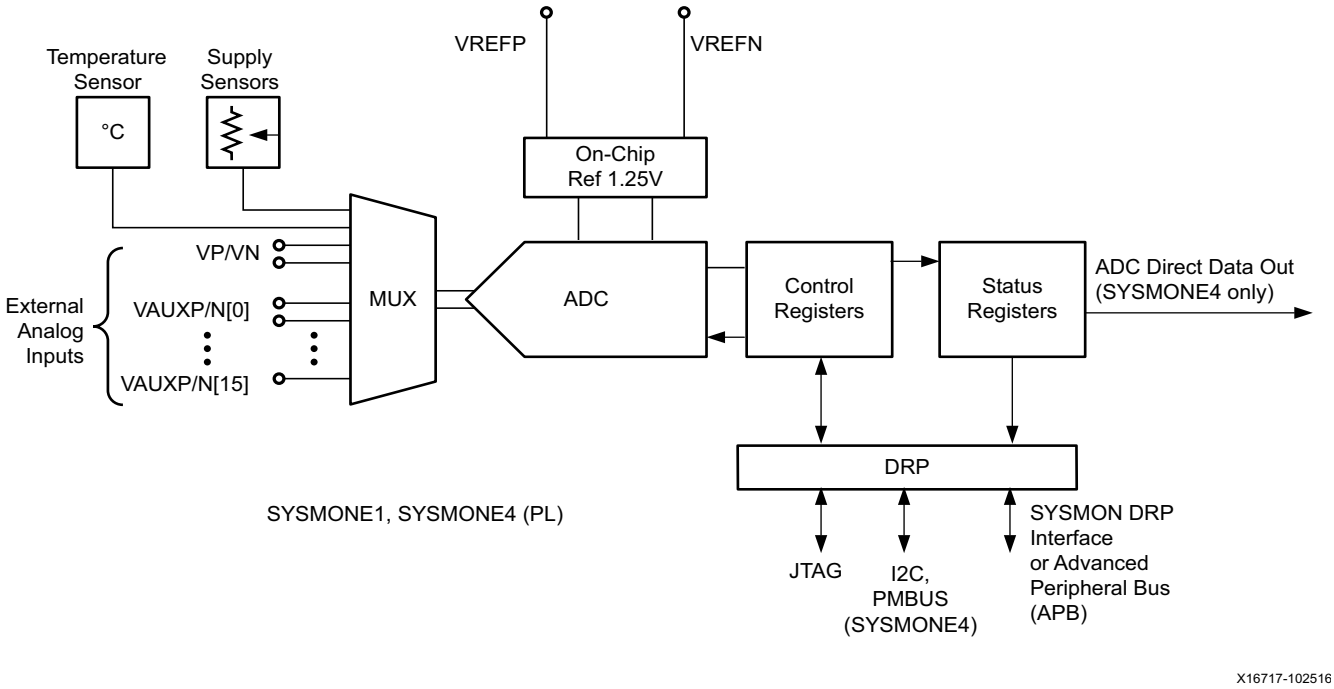


Figure 1-1: **SYSMONE1, SYSMONE4 (PL) Block Diagram**

For the Zynq UltraScale+ MPSoC, the processing system (PS) block contains an additional SYSMON block that is similar to SYSMONE4 in the programmable logic (PL) block (see Figure 1-2). However, the SYSMON block provides a higher sampling frequency of 1000 kSPS along with dedicated sensors for monitoring the PS die temperature and supply references. The PS SYSMON block also contains built-in logic that allows access to both the PS and PL SYSMONE4 blocks that can be used for power management. When SYSMONE4 connects to the APB as a slave, the DRP interface is used and might interrupt existing DRP transactions. For systems that do not want to limit DRP access to SYSMONE4, the PS block can additionally use the standard DRP interface. For additional information on SYSMON within the PS block, see the *Zynq UltraScale+ MPSoC Technical Reference Manual* (UG1085) [Ref 10].

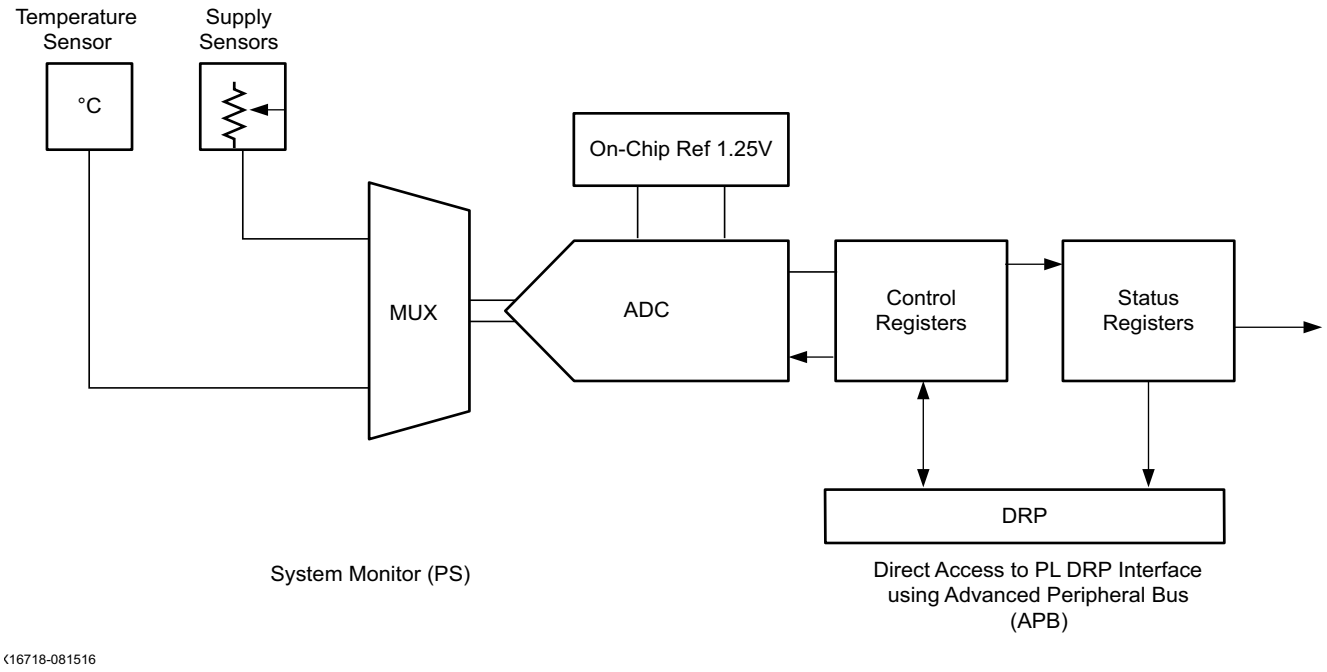


Figure 1-2: **System Monitor (PS) Block Diagram**

If the SYSMON is not instantiated in a design, the device operates in a predefined default mode that monitors on-chip temperature and supply voltage. The SYSMON has numerous operating modes that are user-defined by writing to the control registers, which can be accessed through DRP, JTAG, or I2C. It is also possible to initialize these register contents when the SYSMON is instantiated in a design using the block attributes.

TIP: *When SYSMON is not needed, it can be permanently disabled for a device. SYSMON can be powered down and disabled using this command in an XDC file:*

```
set_property BITSTREAM.GENERAL.SYSMONPOWERDOWN <DISABLE|ENABLE> [current_design]
```

For 3D ICs based on stacked silicon interconnect (SSI) technology, each super logic region (SLR) has one system monitor to provide for monitoring supply voltages within the SLR. I2C DRP and JTAG DRP access is limited to the SYSMON located in the master SLR only. The system monitors can be placed in the bottom SLR0 (SYSMONE1_X0Y0) and then consecutively in the upper SLR increasing Y locations (SYSMONE1_X0Y1 for SLR1, SYSMONE1_X0Y2 for SLR2, etc.). Monitoring across SLR boundaries is not possible. Each SYSMON can only access banks within the SLR. Temperature, V_{CCINT}, V_{CCAUX}, V_{CCBRAM} measurements are specific to an individual SLR. For information on which banks belong to each SLR, see the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1] and the associated ASCII package files.

UltraScale+ 3D ICs add I2C access to the slave SLRs as described in DRP I2C Interface.

For the UltraScale FPGAs SYSMONE1, the System Management Wizard provides I2C functionality to the slave SLRs using the DRP port and additional logic. See the *System*

Management Wizard LogiCORE IP Product Guide (AXI) (PG185) [Ref 9] for additional information.

For UltraScale+ devices, SYSMONE4 adds the ADC_DATA port, which allows access to the measured data. Additionally, for Zynq UltraScale+ MPSoCs, SYSMONE4 monitors the $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, and V_{CC_PSAUX} supplies for the PS.

IMPORTANT: *While the dedicated I2C pins directly connected to SYSMON only support the I2C connection to the master SLR, the System Management Wizard provides I2C functionality that can be used to access the SYSMON blocks within the slave SLRs. Be aware that the additional I2C functionality for 3D ICs in the System Management Wizard uses DRP ports and restrictions might apply.*

Differences from Previous Generations

The SYSMON was designed with the same functionality as the 7 series XADC except for the functional differences described in this section. Because of these functional differences, all XADC designs must be redesigned to the SYSMONE1 primitive. SYSMONE4 adds to the SYSMONE1 functionality.

IMPORTANT: *The SYSMON contains only a single 10-bit 0.2 MSPS ADC. Consequently, the sequencer for SYSMON does not support simultaneous sampling mode or independent ADC mode.*

- 10-bit 0.2 MSPS single-channel analog-to-digital converter
- Any single I/O bank can be selected to include external analog inputs (up to two I/O banks for SYSMONE4)
- Eight additional alarm outputs (16 total alarms)
- Status and control registers extended to 256 addresses
- Simultaneous sampling mode and independent ADC mode are no longer supported

SYSMONE4 for Zynq UltraScale+ MPSoCs, Kintex UltraScale+ FPGAs, and Virtex UltraScale+ FPGAs adds these features to SYSMONE1:

- Direct access to measured data through ADC_DATA port
- Monitoring of PS supplies ($V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, V_{CC_PSAUX} in Zynq UltraScale+ MPSoCs)
- Additional system monitor within PS can operate up to 1 MSPS in Zynq UltraScale+ MPSoCs
- Slow sequence
- SMBALERT for PMBus applications
- Common-N reduces package pins for auxiliary analog inputs by sharing a single N for single ended

SYSMONE4 expands the functionality of SYSMONE1. As a result, migrating to SYSMONE4 allows for settings that were not previously available. Use the SYSMONE4 primitive and add the ADC_DATA and SMBALERT ports. To achieve successful functional simulation, add the SIM_DEVICE attribute.

Table 1-1 lists the differences between the 7 series XADC primitive versus the UltraScale architecture SYSMON primitives.

Table 1-1: **7 Series XADC Migration to UltraScale Architecture Portfolio System Monitor**

Feature	XADC 7 Series FPGAs and Zynq-7000 SoC	SYSMONE1 Kintex UltraScale and Virtex UltraScale FPGAs	SYSMONE4 Kintex UltraScale+ and Virtex UltraScale+ FPGAs and Zynq UltraScale+ MPSoC (PL only)	SYSMON (PS) Zynq UltraScale+ (inside PS block)
Resolution	12-bit	10-bit	10-bit	10-bit
Sample rate	1.0 MSPS	0.2 MSPS	0.2 MSPS	1.0 MSPS
Analog-to-digital converters	2	1	1	1
Auxiliary analog inputs	16	16	16	N/A
Banks supporting external analog inputs	1	All	All	N/A
Control registers	40h to 7Fh	40h to 7Fh	40h to 7Fh, D0h, D1h	40h to 7Fh, D0h, D1h
Status registers	00h to 3Fh	00h to 3Fh, 80h to 8Fh	00h to 3Fh, 80h to 8Fh	00h to 3Fh, 80h to 8Fh
Alarm outputs	8: ALM[7:0]	16: ALM[15:0]	16: ALM[15:0]	16: ALM[15:0]
Temperature sensors	1	1+	1+ (PL)	1 (PS)
System supply sensors	V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, V_{CCPINT}, V_{CCPAUX}, V_{CCO_DDR}	V_{CCINT}, V_{CCAUX}, V_{CCBRAM}	V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, V_{CC_PSAUX}	V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, V_{CC_PSAUX}
USER supply sensors	0	4	4	0
Reconfiguration interfaces	DRP, JTAG	DRP, I2C, JTAG	JTAG, DRP or dedicated PS DRP, I2C and PMBus	JTAG or APB

Table 1-1: **7 Series XADC Migration to UltraScale Architecture Portfolio System Monitor *(Cont'd)***

Feature	XADC 7 Series FPGAs and Zynq-7000 SoC	SYSMONE1 Kintex UltraScale and Virtex UltraScale FPGAs	SYSMONE4 Kintex UltraScale+ and Virtex UltraScale+ FPGAs and Zynq UltraScale+ MPSoC (PL only)	SYSMON (PS) Zynq UltraScale+ (inside PS block)
Sequencer modes	Default, single pass, continuous, single channel, simultaneous sampling, independent ADC	Default, single pass, continuous, single channel	Default, single pass, continuous, single channel, slow sequence	N/A
Sampling modes	Differential sampling	Differential sampling	Differential sampling, single-ended sampling with Common-N	N/A

SYSMON Pinout Requirements

Dedicated Package Pins

Figure 1-3 shows the basic pinout requirements for the SYSMON. There are two recommended configurations. On the left, the SYSMON is powered from V_{CCAUX} (1.8V) and uses an external 1.25V reference source. The external reference delivers the best performance in terms of accuracy and thermal drift. A ferrite bead is used to isolate the ground reference for the analog circuits and system ground. An additional low pass filter for the V_{CCAUX} supply similarly improves the ADC performance (see Chapter 5, Application Guidelines). Shared or common ground impedance is the most common way to introduce unwanted noise into analog circuits.

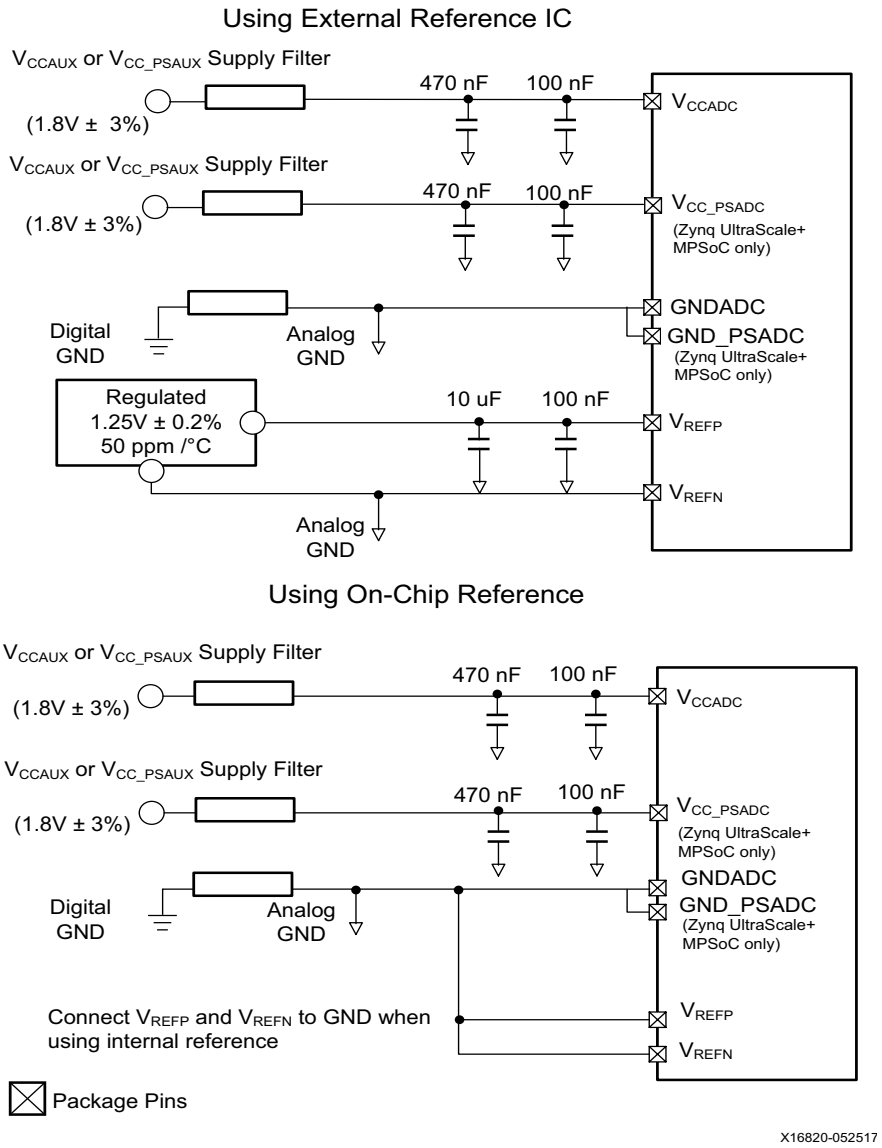


Figure 1-3: **SYSMON Pinout Requirements**

It is also possible to use an on-chip reference for the ADC for SYSMONE1 or SYSMONE4. For Zynq UltraScale+ MPSoC, the PS always uses the on-chip reference. To enable the on-chip reference source for the SYSMONE1 or PL SYSMONE4, the V_{REFP} pin must be connected to GND as shown in Figure 1-3. When only basic on-chip thermal and supply monitoring is required, using the on-chip reference provides good performance. Consult the UltraScale device data sheets to see the accuracy specifications when using the external and on-chip reference sources. Table 1-2 lists the pins associated with the SYSMON and the recommended connectivity.

IMPORTANT: *It is also important to place the 100 nF decoupling capacitors as close as possible to the V_{CCADC_0}, V_{GNDADC_0}, V_{REFP_0} (optional), and V_{REFN_0} (optional) package balls to minimize inductance between the decoupling and package balls.*

Table 1-2: **SYSMON Package Pins**

Package Pin	Type	Description
V_{CCADC}	Power supply	This is the analog supply pin for the ADC and other analog circuits in the SYSMON. The pin can be tied to the 1.8V V_{CCAUX} supply. See Analog Power Supply and Ground for more information. This pin should never be tied to GND. The pin should be tied to V_{CCAUX} even when the SYSMON is not being used.
V_{CC_PSADC}(1)	PS power supply	This is the analog supply pin for the PS ADC and other analog circuits in the SYSMON. The pin can be tied to the 1.8V V_{CC_PSAUX} supply. See Analog Power Supply and Ground for more information. This pin should never be tied to GND. The pin should be tied to V_{CC_PSAUX} or V_{CCAUX} even when the SYSMON is not being used.
V_{CCADC}	Power supply	This is the analog supply pin for the PL ADC and other analog circuits in the SYSMON. The pin can be tied to the 1.8V V_{CCAUX} supply via a low pass filter. See Analog Power Supply and Ground for more information. This pin should never be tied to GND. The pin should be tied to V_{CCAUX} even when the SYSMON is not being used.
GNDADC	Power supply	This is the ground reference pin for the ADC and other analog circuits in the SYSMON. It can be tied to the system ground with an isolating ferrite bead as shown in Figure 1-3. In a mixed-signal system this pin should be tied to an analog ground plane, if available, in which case the ferrite bead is not required. See Analog Power Supply and Ground for more information. This pin should always be tied to GND even if the SYSMON is not being used.
GND_PSADC(1)	PS power supply	This is the ground reference pin for the PS ADC and other analog circuits in the SYSMON. It can be tied to the system ground with an isolating ferrite bead as shown in Figure 1-3. In a mixed-signal system this pin should be tied to an analog ground plane, if available, in which case the ferrite bead is not required. See Analog Power Supply and Ground for more information. This pin should always be tied to GND even if the SYSMON is not being used.

Table 1-2: **SYSMON Package Pins *(Cont'd)***

Package Pin	Type	Description
GNDADC	Power supply	This is the ground reference pin for the PL ADC and other analog circuits in the SYSMON. It can be tied to the system ground with an isolating ferrite bead as shown in Figure 1-3. In a mixed-signal system this pin should be tied to an analog ground plane, if available, in which case the ferrite bead is not required. See Analog Power Supply and Ground for more information. This pin should always be tied to GND even if the SYSMON is not being used.
V_{REFP}	Reference voltage input	This pin can be tied to an external 1.25V accurate reference IC for best performance of the ADC. It should be treated as an analog signal that together with the V_{REFN} signal provides a differential 1.25V voltage. By connecting this pin to GNDADC (see Figure 1-3) an on-chip reference source is activated. This pin should be connected to GNDADC if an external reference is not supplied. See Reference Inputs (VREFP and VREFN) for more information.
V_{REFN}	Reference voltage input	This pin should be tied to ground pin of an external 1.25V accurate reference IC for best performance of the ADC. It should be treated as an analog signal that together with the V_{REFP} signal provides a differential 1.25V voltage. This pin should always be connected to GNDADC even if an external reference is not supplied. See Reference Inputs (VREFP and VREFN) for more information.
V_P	Dedicated analog input	This is the positive input terminal of the dedicated differential analog input channel (V_P/V_N). The analog input channel is very flexible and supports multiple analog input signal types. For more information, see Analog Inputs. This pin should be connected to GNDADC if not used.
V_N	Dedicated analog input	This is the negative input terminal of the dedicated differential analog input channel (V_P/V_N). The analog input channel is very flexible and supports multiple analog input signal types. For more information, see Analog Inputs. This pin should be connected to GNDADC if not used.
_AD0P_ to _AD15P(2)(3)	Auxiliary analog inputs/digital I/O	These are multifunction pins that can support analog inputs or can be used as regular digital I/O (see Figure 1-1). These pins support up to 16 positive input terminals of the differential auxiliary analog input channels (VAUXP/VAUXN). The analog input channels are very flexible and support multiple analog input signal types. For more information, see Analog Inputs. When not being used as analog input, these pins can be treated like any other digital I/O. ***Note:*** The PS SYSMON block does not contain any auxiliary analog input pins.
_AD0N to _AD15N (2)(3)	Auxiliary analog inputs/digital I/O	These are multifunction pins that can support analog inputs or can be used as regular digital I/O (see Figure 1-1). These pins support up to 16 negative input terminals of the differential auxiliary analog input channels (VAUXP/VAUXN). The analog input channels are very flexible and support multiple analog input signal types. For more information, see Analog Inputs. When not being used as analog input these pins can be treated like any other digital I/O. ***Note:*** The PS SYSMON block does not contain any auxiliary analog input pins.

Table 1-2: **SYSMON Package Pins *(Cont'd)***

Package Pin	Type	Description
TCK	Dedicated JTAG input	IEEE Std 1149.1 (JTAG) Test Clock Clock for all devices on a JTAG chain. Connect to the TCK pin of the Xilinx cable header. Treat as a critical clock signal and buffer the cable header TCK signal as necessary for multiple device JTAG chains. If the TCK signal is buffered, connect the buffer input to an external weak (for example, 10 kΩ) pull-up resistor to maintain a valid High when no cable is connected.
TMS	Dedicated JTAG input	JTAG Test Mode Select Mode select for all devices on a JTAG chain. Connect to the TMS pin of the Xilinx cable header. Buffer the cable header TMS signal as necessary for multiple device JTAG chains. If the TMS signal is buffered, connect the buffer input to an external weak (for example, 10 kΩ) pull-up resistor to maintain a valid High when no cable is connected.
TDI	Dedicated JTAG input	JTAG Test Data Input JTAG chain serialized data input. For an isolated device or for the first device in a JTAG chain, connect to the TDI pin of the Xilinx cable header. Otherwise, when the UltraScale device is not the first device in a JTAG chain, connect to the TDO pin of the upstream JTAG device in the JTAG scan chain.
TDO	Dedicated JTAG output	JTAG Test Data Output JTAG chain serialized data output. For an isolated device or for the last device in a JTAG chain, connect to the TDO pin of the Xilinx cable header. Otherwise, when the UltraScale device is not the last device in a JTAG chain, connect to the TDI pin of the downstream JTAG device in the JTAG scan chain.
I2C_SDA[(4)]	Multifunction SYSMON I2C I/O	Multifunction pin that can be used to support the I2C DRP interface for SYSMON. I2C_SDA is the data pin for the DRP I2C interface. See DRP I2C Interface for more information.  **IMPORTANT:** *I2C is a bidirectional interface that is active prior to configuration. During this time, these pins should only be used for I2C access.*
I2C_SCLK[(4)]	Multifunction SYSMON I2C I/O	Multifunction pin that can be used to support the I2C DRP interface for SYSMON. I2C_CLK is the clock pin for the DRP I2C interface. See DRP I2C Interface for more information.  **IMPORTANT:** *I2C is a bidirectional interface that is active prior to configuration. During this time, these pins should only be used for I2C access.*

Table 1-2: **SYSMON Package Pins *(Cont'd)***

Package Pin	Type	Description
SMBALERT[4]	Multifunction SYSMON output	Optional PMBus alert. When Low indicates a system fault that must be cleared using PMBus commands. Connect to SMBALERT_TS. See DRP I2C Interface and Figure 3-11 "SYSMON I2C DRP Interface" for more information. **IMPORTANT:** *SMBALERT is active prior to configuration.*

1. Applicable to Zynq UltraScale+ MPSoCs.
2. I/Os that are analog input-enabled contain the _ADxP_ and _ADxN_ designation in the package file name, for example, IO_L1P_T0_AD0P_35 is the input pin for analog auxiliary channel VAUXP[0]. IO_L1N_T0_AD0N_35 is the input pin for analog auxiliary channel VAUXN[0]. For more information, see the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1].
3. Due to the reduced number of available package pins, HD I/O banks support either 12 auxiliary analog inputs (VAUXP[11:0/VAUXN[11:0]) or 8 auxiliary analog inputs(VAUXP[11:8,3:0], VAUXN[11:8,3:0]).
4. In some devices, these pins reside in an HP bank and have a maximum V_{CCO} value of 1.8V. External circuitry (such as external level shifters) might be required.

IMPORTANT: *Consult Chapter 5, Application Guidelines before commencing any PC board layout. Board layout and external component choices can greatly impact the performance of the ADC. For additional PCB layout guidelines, see XADC Layout Guidelines (XAPP554) [Ref 2].*

External Analog Inputs

Apart from a single dedicated analog input pair (V_P/V_N), SYSMON supports up to 16 external analog inputs (auxiliary analog inputs). Because the auxiliary analog inputs are supported in dual-purpose I/Os, only the auxiliary analog inputs used by a design are connected. When package pins are used as auxiliary analog inputs, they cannot also be used as digital I/O.

Due to the reduced number of available package pins, HD I/O banks support either 12 auxiliary analog inputs (VAUXP[11:0/VAUXN[11:0]) or 8 auxiliary analog inputs (VAUXP[15:8], VAUXN[15:8]). See *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1]. The auxiliary analog inputs are enabled by connecting the analog inputs on the SYSMONE1 or SYSMONE4 primitive to the top level of the design and setting the I/O standard to ANALOG or ANALOG_SE (when using Common-N inputs for SYSMONE4). For example, AD0P and AD0N are connected to inputs at the top level of the design. Vivado® synthesis infers an IBUF_ANALOG input primitive for each input.

IBUF and IBUF_ANALOG are allowed for the auxiliary analog inputs but are not required. Differential input buffers such as the IBUFDS primitive are not supported.

To enable the auxiliary analog inputs preconfiguration, write `0001h` to DRP address `02h` through the JTAG interface. The auxiliary analog inputs cannot be enabled through the PS interface. Preconfiguration, the auxiliary analog inputs are limited to bank 66.

The SYSMONE1 allows any single I/O bank to support the auxiliary analog input connections. The SYSMONE4 allows up to two banks (in multi-SLR devices in the one SLR) to support the auxiliary analog input connections (the auxiliary analog inputs should be connected to bank(s) in same SLR). Within a given bank, up to 16 differential package pin pairs can connect to the SYSMON's differential analog input circuitry. Analog input voltages cannot exceed the I/O bank supply (V_{CCO}). Analog inputs must set IOSTANDARD = ANALOG or ANALOG_SE (when using Common-N inputs for SYSMONE4). To assign an auxiliary analog input to a particular bank, assign the input to a valid analog input as designated by _AD[15:0]P_<BANK #> or _AD[15:0]N_<BANK #>. The Vivado pin planner can be used to help identify allowable pins for each bank. For example, _AD0P_<BANK #> should be assigned to the input connected to VAUXP[0] port for the SYSMONE1 instantiation.

IMPORTANT: *All auxiliary analog inputs must connect to the appropriate pin and the SYSMON port number must align to the pin's reference number. For example, _AD0P_ must only be connected to the VAUXP[0] SYSMON port.*

All analog input channels are differential and require two inputs. For SYSMONE1, both inputs must come from package balls. SYSMONE4 optionally supports Common-N inputs allowing a single N package ball to be shared among a number of auxiliary analog inputs. Auxiliary channels that use Common-N to share a common N package ball must all reside in the same bank. See the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1] for more information. See Analog Inputs for more information.

Instantiating the SYSMON

It is not necessary to instantiate the SYSMON in a design to access the on-chip monitoring capability. However, if the SYSMON is not instantiated in a design, the only way to access this information is by using either the JTAG TAP or I2C. To allow access to the status registers (measurement results) from the interconnect logic, the SYSMON must be instantiated. These subsections give a brief overview of the SYSMONE1 and SYSMONE4 primitives (ports and attributes).

SYSMON Ports

Figure 1-4 shows the ports on the SYSMONE1 primitive and Figure 1-5 shows the ports on the SYSMONE4 primitive. Table 1-3 describes the functionality of the ports.

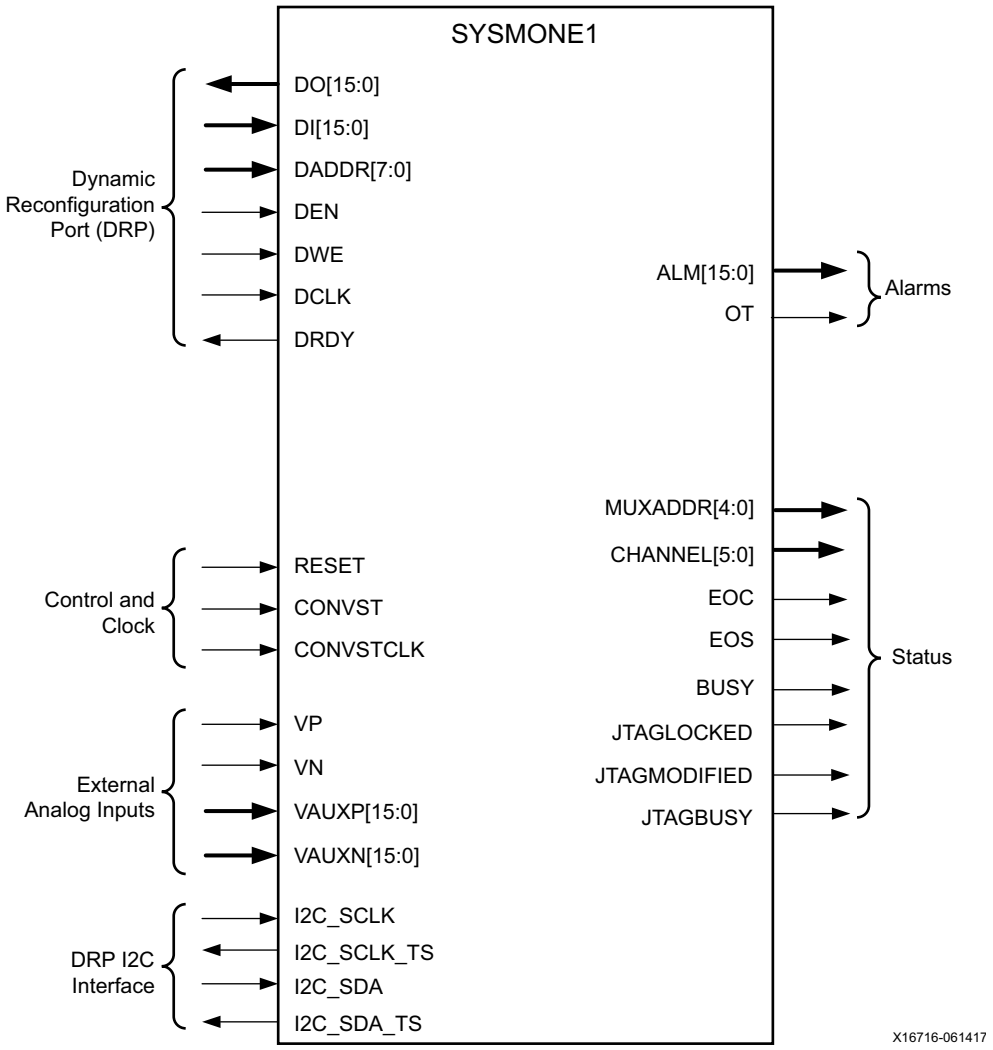


Figure 1-4: **SYSMONE1 Primitive Ports**

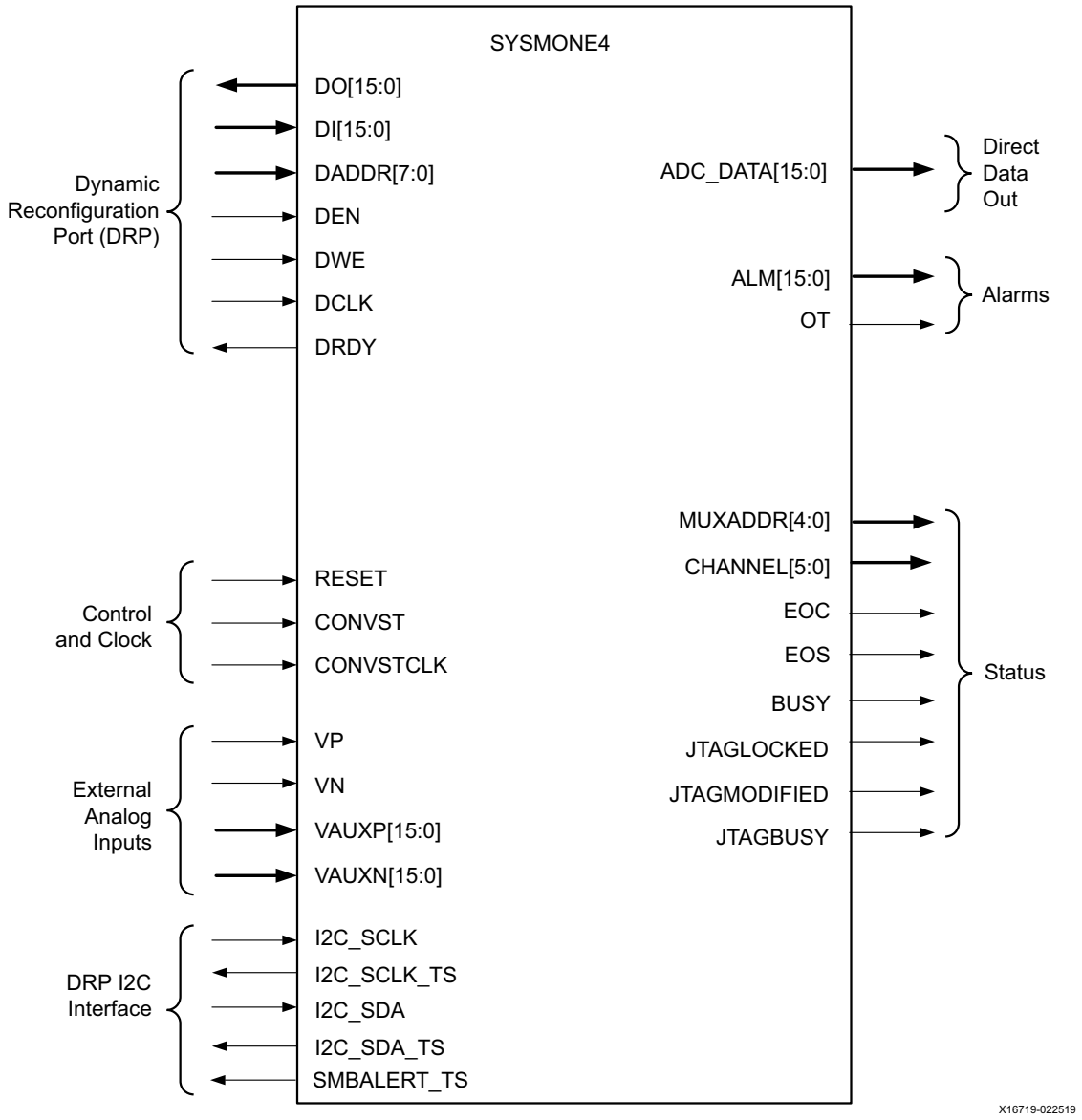


Figure 1-5: **SYSMONE4 Primitive Ports**

Table 1-3: **SYSMON Port Descriptions**

Port	I/O	Description
ADC_DATA[15:0]	Output	(SYSMONE4 only) Direct data output. Measurement results updated every conversion (EOC). To decode what channel the data corresponds to, use with CHANNEL.
DI[15:0]	Input	Input data bus for the DRP.(1)
DO[15:0]	Output	Output data bus for the DRP.(1)
DADDR[7:0]	Input	Address bus for the DRP.(1)
DEN	Input	Enable signal for the DRP.(1)
DWE	Input	Write enable for the DRP.(1)
DCLK	Input	Clock input for the DRP.(1)
DRDY	Output	Data ready signal for the DRP.(1)
RESET	Input	Asynchronous reset signal for the SYSMON control logic. RESET is deasserted synchronously to DCLK or internal configuration when DCLK is stopped.
CONVST	Input	Convert start input. This input controls the sampling instant on the ADC(s) input and is only used in event driven sampling mode timing. This input comes from the general-purpose interconnect. (See Adjusting the Acquisition Settling Time).
CONVSTCLK	Input	Convert start clock input. This input is connected to a clock net. Like CONVST, this input controls the sampling instant on the ADC(s) inputs and is only used in event driven sampling mode timing. This input comes from the local clock distribution network. Thus, for the best control over the sampling instant (delay and jitter), a global clock input can be used as the CONVSTCLK source. (See Adjusting the Acquisition Settling Time).
V_P, V_N	Input	One dedicated analog input pair. The SYSMON has one pair of dedicated analog input pins that provides a differential analog input. When designing with the SYSMON feature without using the dedicated external channel of V_P and V_N, connect both V_P and V_N to analog ground.
VAUXP[15:0], VAUXN[15:0]	Input	Sixteen auxiliary analog input pairs. In addition to the dedicated differential analog input, the SYSMON can access 16 differential analog inputs by configuring digital I/O as analog inputs. For auxiliary analog inputs using the shared Common-N inputs, only VAUXP must be connected. These inputs can also be enabled preconfiguration with the JTAG port (see DRP JTAG Interface).
ALM[0]	Output	Temperature sensor alarm output. When High, measured data violates alarm thresholds.
ALM[1]	Output	V_{CCINT} sensor alarm output. When High, measured data violates alarm thresholds.
ALM[2]	Output	V_{CCAUX} sensor alarm output. When High, measured data violates alarm thresholds.
ALM[3]	Output	V_{CCBRAM} sensor alarm output. When High, measured data violates alarm thresholds.

Table 1-3: **SYSMON Port Descriptions *(Cont'd)***

Port	I/O	Description
ALM[4]	Output	$V_{CC\_PSINTLP}$. (SYSMONE4 only) sensor alarm output. When High, measured data violates alarm thresholds.
ALM[5]	Output	V_{CC_INTFP}. (SYSMONE4 only) sensor alarm output. When High, measured data violates alarm thresholds.
ALM[6]	Output	V_{CC_PSAUX}. (SYSMONE4 only) sensor alarm output. When High, measured data violates alarm thresholds.
ALM[7]	Output	Logic OR of bus ALM[6:0]. Can be used to flag the occurrence of any alarm in this group.
ALM[11:8]	Output	Alarms of user-selected sources USER[3:0]. When ALM[8] is High, the measured USER0 data violates alarm thresholds (see Power and User Supply Sensors.)
ALM[15]	Output	Logic OR of buses ALM[11:8] and ALM[6:0]. Can be used to flag the occurrence of any alarm in this group.
OT	Output	Over-Temperature alarm output.
MUXADDR[4:0]	Output	These outputs are used in external multiplexer mode. They indicate the address of the next channel in a sequence to be converted. They provide the channel address for an external multiplexer (see External Multiplexer Mode).
CHANNEL[5:0]	Output	Channel selection outputs. The ADC input MUX channel selection for the current ADC conversion is placed on these outputs at the end of an ADC conversion.
EOC	Output	End of conversion signal. This signal transitions to active High at the end of an ADC conversion when the measurement is written to the status registers.
EOS	Output	End of sequence. This signal transitions to active High when the measurement data from the last channel in an automatic channel sequence is written to the status registers.
BUSY	Output	ADC busy signal. This signal transitions High during an ADC conversion. This signal also transitions High for an extended period during an ADC or sensor calibration.
JTAGLOCKED	Output	Indicates that a DRP port lock request has been made by the JTAG interface (see DRP JTAG Interface). This signal is also used to indicate that the DRP is ready for access (when Low).
JTAGMODIFIED	Output	Used to indicate that a JTAG write to the DRP has occurred.
JTAGBUSY	Output	Used to indicate that a JTAG DRP transaction is in progress.
I2C_SDA	Input	Input for I2C_SDA. Required for DRP I2C interface. For SYSMONE1, the I2C_SDA and I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in DRP I2C Interface. In SYSMONE4, the ports can also connect to internal logic.
I2C_SDA_TS	Output	Output for I2C_SDA. Required for DRP I2C interface. For SYSMONE1, the I2C_SDA and I2C_SDA_TS ports must be connected to the dedicated I2C_SDA package pin as described in DRP I2C Interface. In SYSMONE4, the ports can also connect to internal logic.

Table 1-3: **SYSMON Port Descriptions *(Cont'd)***

Port	I/O	Description
I2C_SCLK	Input	Input for I2C_SCLK. Required for DRP I2C interface. For SYSMONE1, the I2C_SCLK and I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in DRP I2C Interface. In SYSMONE4, the ports can also connect to internal logic.
I2C_SCLK_TS	Output	Output for I2C_SCLK. Required for DRP I2C interface. For SYSMONE1, the I2C_SCLK and I2C_SCLK_TS ports must be connected to the dedicated I2C_SCLK package pin as described in DRP I2C Interface. In SYSMONE4, the ports can also connect to internal logic.
SMBALERT_TS	Output	(SYSMONE4 only) output control signal for SMBALERT. Connect to SMBALERT. See Figure 3-11 "SYSMON I2C DRP Interface".

Notes:

1. The DRP is the interface between the SYSMON and the device. All SYSMON registers can be accessed from the interconnect logic using this interface. Not available when the dedicated PS interface is being used to connect directly to SYSMONE4 from the PS block.

SYSMON Attributes

The block diagram in Figure 1-1 shows the 16-bit control registers that define the operation of the SYSMON. These registers can be read and written using the DRP, JTAG, or I2C ports. It is also possible to initialize the contents of these registers during the configuration using attributes for the SYSMONE1 primitive. The attributes (Table 1-4) are called INIT_xx, where *xx* corresponds to the hexadecimal address of the register on the DRP. For example, INIT_40 corresponds to the first control register at address `40h` on the DRP. The control registers and the INIT_xx values are described in detail in Figure 3-2.

Table 1-4: **SYSMON Primitive Attributes**

Attribute	Type	Allowed Values	Description
SIM_MONITOR_FILE	String	–	Simulation analog entry file name.
SIM_DEVICE	String	ULTRASCALE_PLUS, ZYNQ_ULTRASCALE	(SYSMONE4 only) Target device. Simulation models use SIM_DEVICE to determine the channels used for the default mode. Use ULTRASCALE_PLUS when using either Kintex UltraScale+ or Virtex UltraScale+ FPGAs.
INIT_40 to INIT_7F	Integer	`0000h to FFFFh`	Initialization values for control register addresses `40h` to `7Fh`. See Table 3-4.
SYSMON_VUSER[3:0]_BANK	Integer	Specific to device and package	SYSMON_VUSER[3:0]_BANK and SYSMON_VUSER[3:0]_MONITOR are both required for selecting a power supply to be measured by V_{USER}. Restrictions apply. Use the System Management Wizard for selecting.

Table 1-4: **SYSMON Primitive Attributes *(Cont'd)***

Attribute	Type	Allowed Values	Description
SYSMON_VUSER[3:0]_MONITOR	String	V_{CCO}[1], V_{CCO_TOP}[2], V_{CCO_BOT}[2], V_{CCINT}, V_{CCAUX}	SYSMON_VUSER[3:0]_BANK and SYSMON_VUSER[3:0]_MONITOR are both required for selecting a power supply to be measured by V_{USER}. Restrictions apply. Use the System Management Wizard for selecting.
COMMON_N_SOURCE	Integer	`0h` to `fh`	(SYSMONE4 only) Sets the auxiliary analog input that is used for the Common-N input. For example, if COMMON_N_SOURCE = `0h`, VAUXN[0] is used.

Notes:

1. Only supported in HP I/O and HD I/O banks.
2. Only supported in HR I/O banks. In some devices and packages, some of the HR I/O banks are split into smaller 26-pin banks. These banks do not support V_{CCO_TOP} and V_{CCO_BOT}. See *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1].

The SYSMONE1 primitive also has the SIM_MONITOR_FILE attribute that points to the analog stimulus file. This attribute is required to support simulation. This attribute points to the path and file name of a text file that contains analog information (for example, temperature and voltage). UNISIM and SIMPRIM models use this text file during simulation. This is the only way analog signals can be introduced into a simulation of the SYSMON. For more information, see SYSMON Software Support.

SYSMON_VUSER[3:0]_BANK and SYSMON_VUSER[3:0]_MONITOR attributes must be used together to select the on-chip user supply monitor. For example, if V_{USER0} is used to measure the V_{CCO} in bank 66, SYSMONE1 must be set to SYSMON_VUSER0_bank(66) and SYSMON_VUSER0_MONITOR(VCCO). UltraScale architecture-based devices support V_{CCO} supplies differently in HR I/O banks and HP I/O banks. In HP I/O banks, SYSMON_VUSER[3:0]_MONITOR must be set to V_{CCO} when the V_{CCO} supply is being measured. In UltraScale architecture-based devices, HR I/O banks are split into either the top or bottom of the HR I/O bank. HR I/O banks must be set to either V_{CCO_TOP} or V_{CCO_BOT}. Every on-chip user supply can be independently set. Because partial reconfiguration can limit V_{USER} resources, it is recommended that V_{USER} readings are ignored and that all V_{USER} alarms are disabled until partial reconfiguration is complete. Due to routing restrictions, the System Management Wizard should be used for setting the on-chip user supplies.

IBUF_ANALOG

Figure 1-6 shows the ports on the IBUF_ANALOG primitive, and Table 1-5 describes the functionality of the ports. The IBUF_ANALOG is used to indicate the dedicated analog routing to the SYSMON block.

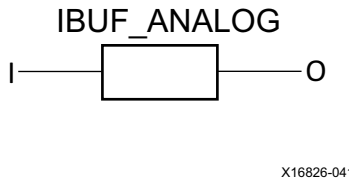


Figure 1-6: **Auxiliary Analog Inputs (IBUF_ANALOG)**

Table 1-5: **Port Descriptions**

Port	I/O	Description
I	Input	Input connection. Directly connect to the top-level input port of the design.
O	Output	Output connection. Directly connect to the auxiliary analog inputs of the SYSMONE1 or SYSMONE4 primitive.

This primitive is used to connect the external auxiliary analog inputs to the SYSMONE1 or SYSMONE4 component. When using the VAUXP/VAUXN pins of the SYSMONE1 component, this primitive allows for a proper connection to the top-level port in the design.

IMPORTANT: *The IBUF_ANALOG primitive is not a buffer.*

The IBUF_ANALOG primitive is automatically inserted (inferred) by the synthesis tool when connecting auxiliary analog inputs from the SYSMONE1 and SYSMONE4 primitive to the top-level input port of the design.

Inferencing Examples

The Verilog and VHDL examples in this section show how two IBUF_ANALOG primitives are inferred for each auxiliary analog input as shown in Figure 1-7.

Verilog

```
module ug580 (
    output EOS,
    input AD0P,
    input AD0N
    );
wire [15:0] vauxp, vauxn;

assign vauxp = {15'h0000, AD0P};
assign vauxn = {15'h0000, AD0N};

SYSMONE1 SYSMON_INST (
    .EOS (EOS),
    .VAUXP (vauxp),
    .VAUXN (vauxn)
);
endmodule
```

VHDL

```
  library IEEE;
  use IEEE.STD_LOGIC_1164.ALL;
  library UNISIM;
  use UNISIM.VComponents.all;

  entity ug580_ibuf_test is
      Port (
          AD0P: in std_logic;
          AD0N: in std_logic;
          EOS: out std_logic
          );
  end ug580_ibuf_test;

  architecture Behavioral of ug580_ibuf_test is
     signal vauxp: std_logic_vector(15 downto 0);
     signal vauxn: std_logic_vector(15 downto 0);
  begin

  vauxp <= "000000000000000" & AD0P;
  vauxn <= "000000000000000" & AD0N;
  SYSMONE1_inst : SYSMONE1
        port map (
           ALM => open,
           OT => open,
           DO => open,
           DRDY => open,
           BUSY => open,
           CHANNEL => open,
           EOC => open,
           EOS => EOS,
           JTAGBUSY => open,
```

```
            JTAGLOCKED => open,
            JTAGMODIFIED => open,
            MUXADDR => open,
            VAUXN => vauxn,
            VAUXP => vauxp,
            CONVST => '0',
            CONVSTCLK => '0',
            RESET => '0',
            VN => '0',
            VP => '0',
            DADDR => X"00",
            DCLK => '0',
            DEN => '0',
            DI => X"0000",
            DWE => '0',
            I2C_SCLK => '0',
            I2C_SDA => '0'
        );
end Behavioral;
```

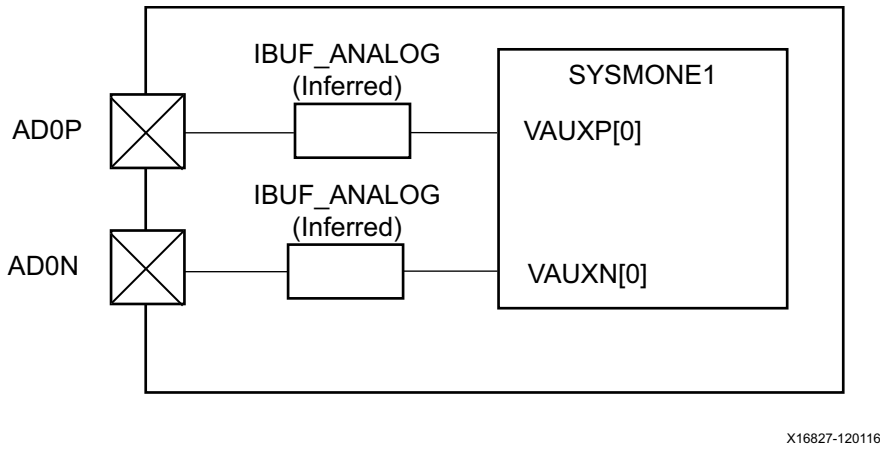


Figure 1-7: **Inferred IBUF_ANALOG Primitives for Auxiliary Analog Input**

ADC and Sensors

For more comprehensive information on the operation of the ADCs and on-chip sensors, see Chapter 2, Basic Functionality. This section provides a brief overview on how to quickly interpret data read from the status registers and verify the operation of the SYSMON.

Analog-to-Digital Converter

The ADC has a nominal analog input range from 0V to 1V. In unipolar mode (default), the analog inputs of the ADC produce a full scale code of `3FFh` (10 bits) when the input is 1V. Thus, an analog input signal of 200 mV in unipolar mode produces an output code of:

$$((0.2/1.0) \times 3FFh) = 204 \text{ or } CCh \qquad \textit{Equation 1-1}$$

In bipolar mode, the ADC uses two's complement coding and produces a full scale code of `1FFh` with +0.5V input and `200h` with –0.5V input.

Temperature Sensor

The transfer function for temperature depends on the architecture and reference source. See Equation 2-5 through Equation 2-12 for transfer functions specific to the reference used and the architecture.

Power and User Supply Sensors

The SYSMON power supply sensors have a transfer function that generates a full scale ADC output code of `3FFh` with a 3V input voltage. This voltage is outside the allowed supply range, but the device supply measurements map into this measurement range. Thus, V_{CCINT} = 1V generates an output code of 1/3 x 1024 = 341 = `155h`. The SYSMON monitors V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, and V_{CC_PSAUX}. The measurement results are stored in status registers `01h`, `02h`, `06h`, `0Dh`, `0Eh`, and `0Fh`, respectively.

Furthermore, the SYSMON allows four additional supplies (VUSER[3:0]) to be measured in status registers `80h`, `81h`, `82h`, and `83h`. The System Management Wizard connects VUSER[3:0] to a bank's V_{CCO}, V_{CCO_TOP}, V_{CCO_BOT}, V_{CCINT}, or V_{CCAUX} supply pins. The four measured supplies can be located in different banks. The System Management Wizard provides the allowable connections. Because the user supplies can be used with HR I/O banks and HD I/O banks, a wider input range is required. As a result, the user supplies can have a full scale ADC output code of `3FFh` with a 6V input voltage for HR I/O banks. See Power Supply Sensor for more information.

Basic Functionality

The SYSMON block contains a 10-bit, 0.2 MSPS ADC. The ADC can be used with both external analog inputs and on-chip sensors. Several predefined operating modes are available that cover the most typical use cases for the ADC. These operating modes are described in Chapter 4. This chapter focuses on the detailed operation of the ADC and the on-chip sensors. The various input configurations for the external analog inputs are also covered. All operating modes of the ADC, sensors, and analog inputs are configured using the SYSMON control registers. A detailed description of the control registers is covered in Chapter 3.

ADC Transfer Functions

The ADC has transfer functions as shown in Figure 2-1 and Figure 2-2. These transfer functions reflect unipolar and bipolar operating modes, respectively. All on-chip sensors use the unipolar mode of operation for the ADC. External analog input channels can operate in unipolar or bipolar modes (see Analog Inputs and ADC Channel Analog-Input Mode (4Ch, 4Dh)).

IMPORTANT: *For the ADC to function as specified, the power supplies and reference options must be configured correctly.*

The required package ball connections are shown in Figure 1-3. The PCB layout and external component selection are important for ensuring optimal ADC performance and are covered in Chapter 5.

RECOMMENDED: *Read Chapter 5 before the board design is started.*

TIP: *The ADC always produces a 16-bit conversion result, and the full 16-bit result is stored in the 16-bit status registers. The 10-bit transfer functions shown in this section correspond to the 10 MSBs (most significant or left-most bits) in the 16-bit status registers. The six LSBs can be used to minimize quantization effects or improve resolution through averaging or filtering.*

Unipolar Mode

Figure 2-1 shows the 10-bit unipolar transfer function for the ADC. The nominal analog input range to the ADC is 0V to 1V in this mode. The ADC produces a zero code (000h) when 0V is present on the ADC input and a full scale code of all 1s (3FFh) when 1V is present on the input.

The ADC output coding in unipolar mode is straight binary. The designed code transitions occur at successive integer LSB values such as one LSB, two LSBs, and three LSBs (and so on). The LSB size in volts is equal to $1V/2^{10}$ or 1V/1024 = 977 µV. The analog input channels are differential and require both the positive (V_P) and negative (V_N) inputs of the differential input to be driven. For more information, see the Analog Inputs section.

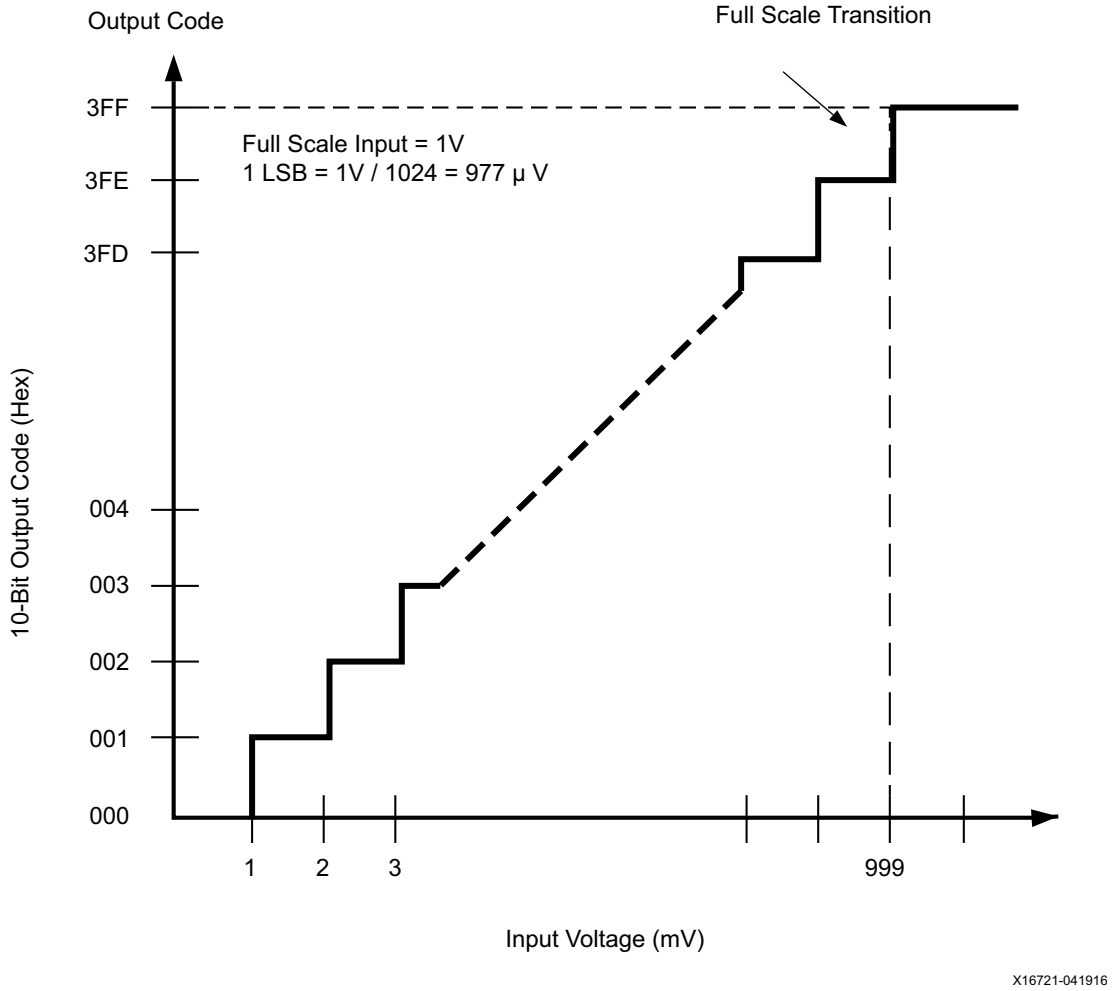


Figure 2-1: **Unipolar Transfer Function**

Bipolar Mode

When the external analog input channels of the ADC are configured as bipolar, they can accommodate true differential and bipolar analog signal types (see the Analog Inputs section). When dealing with differential signal types, it is useful to have both sign and magnitude information about the analog input signal. Figure 2-2 shows the ideal transfer function for bipolar mode operation. The output coding of the ADC in bipolar mode is two's complement and indicates the sign of the input signal on V_P relative to V_N. The designed code transitions occur at successive integer LSB values, that is, one LSB, two LSBs, three LSBs, etc. The LSB size in volts is equal to $1V/2^{10}$ or 1V/1024 = 977 µV.

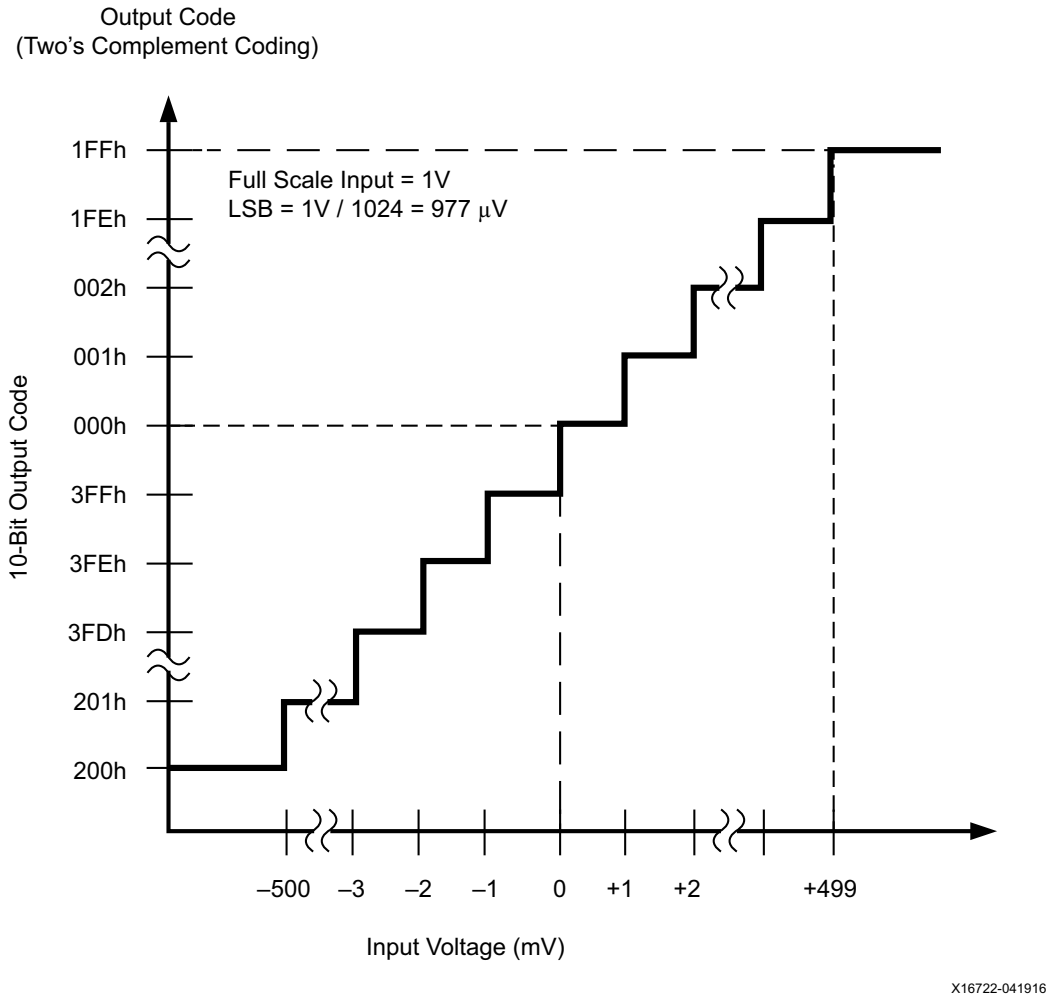


Figure 2-2: **Bipolar Transfer Function**

Analog Inputs

The analog inputs of the ADC use a differential sampling scheme to reduce the effects of common-mode noise signals. This common-mode rejection improves the ADC performance in noisy digital environments. Figure 2-3 shows the benefits of a differential sampling scheme. Common ground impedances (R_G) easily couple noise voltages (switching digital currents) into other parts of a system. These noise signals can be 100 mV or more. For the ADC, this noise voltage is equivalent to hundreds of LSBs, thus inducing large measurement errors. The differential sampling scheme samples both the signal and any common mode noise voltages at both analog inputs (V_P and V_N). The common mode signal is effectively subtracted because the track-and-hold amplifier captures the difference between V_P and V_N or V_P minus V_N. To take advantage of the high common mode rejection, connect V_P and V_N in a differential configuration.

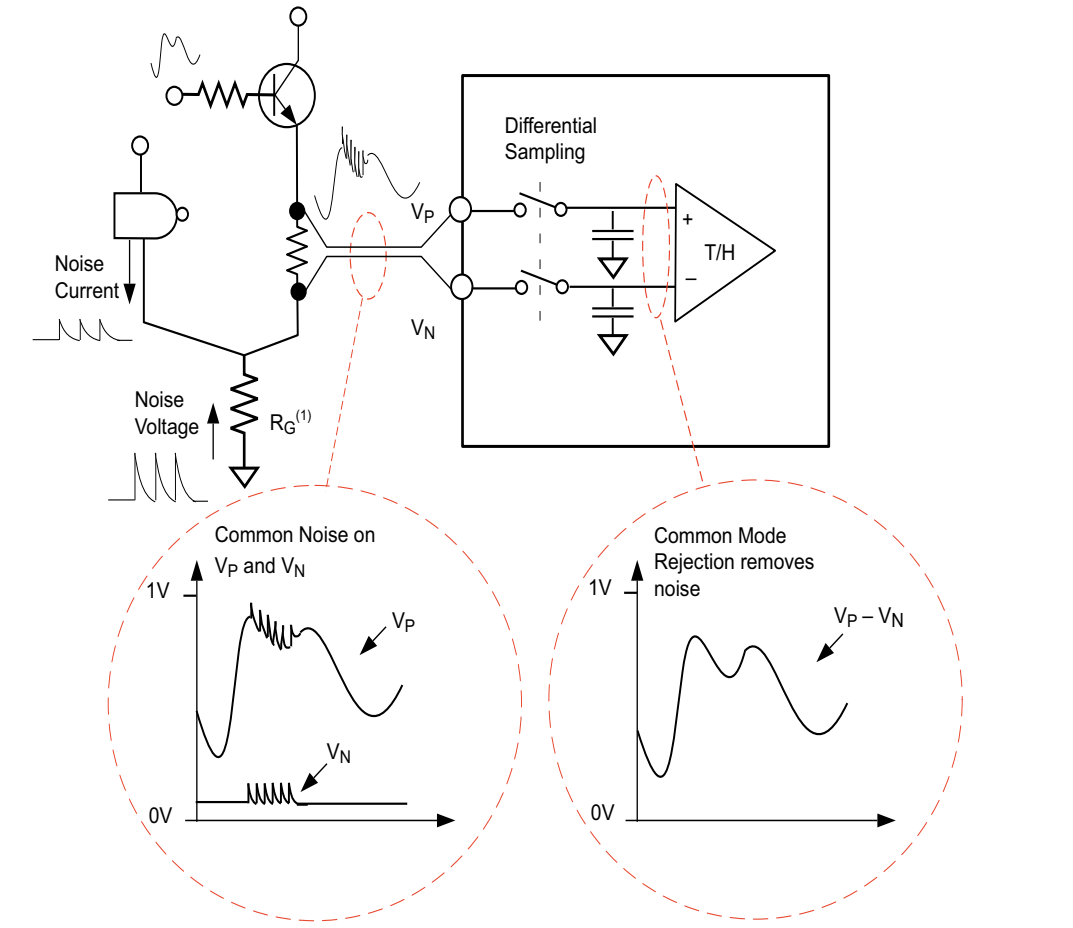


Figure 2-3: **Common Mode Noise Rejection**

To free up package pins for SYSMONE4, the auxiliary analog inputs also support single-ended sampling when using the Common-N mode. When using the Common-N mode, the number of package pins required to support 16 auxiliary analog inputs is reduced from 32 package pins to 17 package pins as shown in Figure 2-4. Because the common ground noise is not being compensated for, the single-ended sampling performance degrades. To compensate for the reduced accuracy of the single-ended sampling, averaging should be used. Auxiliary analog inputs that leverage Common-N must all reside in the same bank.

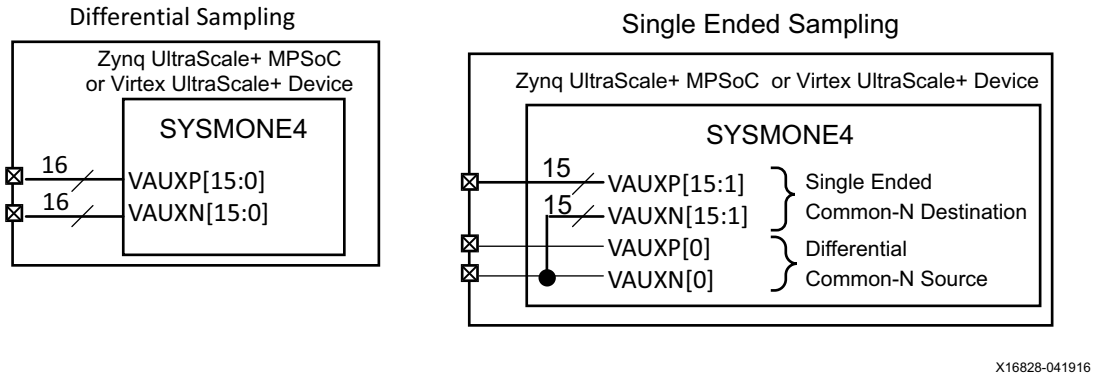


Figure 2-4: **Reducing Package Pins Using Common-N**

Auxiliary Analog Inputs

The auxiliary analog inputs (VAUXP[15:0] and VAUXN[15:0]) are analog inputs that are shared with regular digital I/O package balls. Only the auxiliary inputs connected in a design are enabled as analog inputs. Not all packages can support all banks fully. These partially populated banks can have 0, 8, or 12 auxiliary channels for a given package. The SYSMON auxiliary inputs pins are labeled in the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1] by appending *_ADxP_* and *_ADxN_* to the I/O name, where *x* is the auxiliary pair number. For example, the auxiliary input VAUXP[15] could be designated IO_LxxP_xx_AD15P_xx in the pinout specification.

When designated as analog inputs, these inputs are unavailable for use as digital I/Os. If the I/O is used as a digital I/O, it is subject to the specifications of the configured I/O standard.

IMPORTANT: *If the I/O is used as an analog input, the input voltage must adhere to the specifications given in the analog-to-digital converter table in the UltraScale device data sheets.*

Additionally, the I/O standard should be set to ANALOG. As an example, to assign VAUXP0 and VAUXN0 to the ANALOG I/O standard:

```
set_property PACKAGE_PIN value [get_ports VAUXP0]
set_property IOSTANDARD ANALOG [get_ports VAUXP0]
```

For the SYSMONE4 auxiliary analog inputs that use the Common-N configuration, the I/O standard should be set to ANALOG_SE. For example, when VAUXP[1] uses Common-N:

```
set_property PACKAGE_PIN value [get_ports VAUXP1]
set_property IOSTANDARD ANALOG_SE [get_ports VAUXP1]
```

The shared N side of a common N pin should have its IOSTANDARD set to ANALOG:

```
set_property PACKAGE_PIN value [get_ports VAUXN2]
set_property IOSTANDARD ANALOG [get_ports VAUXN1]
```

The SYSMONE4 primitive must also set the COMMON_N_SOURCE attribute. For example, VAUXN0 is used as the source when COMMON_N_SOURCE = `0h`. It is possible to enable up to 16 auxiliary analog inputs in an I/O bank and use the remaining as digital I/Os. If there is a mixture of analog and digital I/Os in a bank, the I/O bank must be powered by a supply required to meet the specifications of the digital I/O standard in used. The analog input signal should not exceed the I/O bank supply voltage (V_{CCO}) in this case.

Adjusting the Acquisition Settling Time

SYSMON supports two modes for sampling the analog channels, continuous sampling mode or event-driven sampling mode.

IMPORTANT: *For both sampling modes, make sure that the acquisition settling time is sufficient to support the conversion times and the clock frequencies used.*

Continuous Sampling Mode

In the continuous sampling mode, the ADC continuously performs conversions. Separate operating mode settings determine which analog channel is selected. In this mode, 26 ADCCLK cycles are required to acquire an analog signal and perform a conversion. The maximum conversion rate specified for the ADC is 0.2 MSPS or a conversion time of 5 µs. This implies a maximum ADCCLK frequency of 5.2 MHz. See Figure 2-5.

Note: The ADCCLK is an internal clock that is only available to the ADC. The ADCCLK cannot be accessed. See Table 3-10, page 58 to determine the ratio between DCLK and ADCCLK.

If the ACQ (see Control Registers) bit has not been set, four ADCCLKs or 769 ns is allowed for the final stages of the acquisition. This *settling* time ensures that the analog input voltage is acquired to a 10-bit accuracy. The settling time can be increased by reducing the ADCCLK frequency or setting the ACQ bit (single channel, `40h`) or the associated ACQ bit for the sequencer (SEQACQ[2:0], `4Eh`, `4Fh`). In the latter case, assuming 5.2 MHz clock, the settling time is increased to 1923 ns (10 ADCCLK cycles), and the conversion rate would be reduced to 162 kSPS for the same ADCCLK frequency.

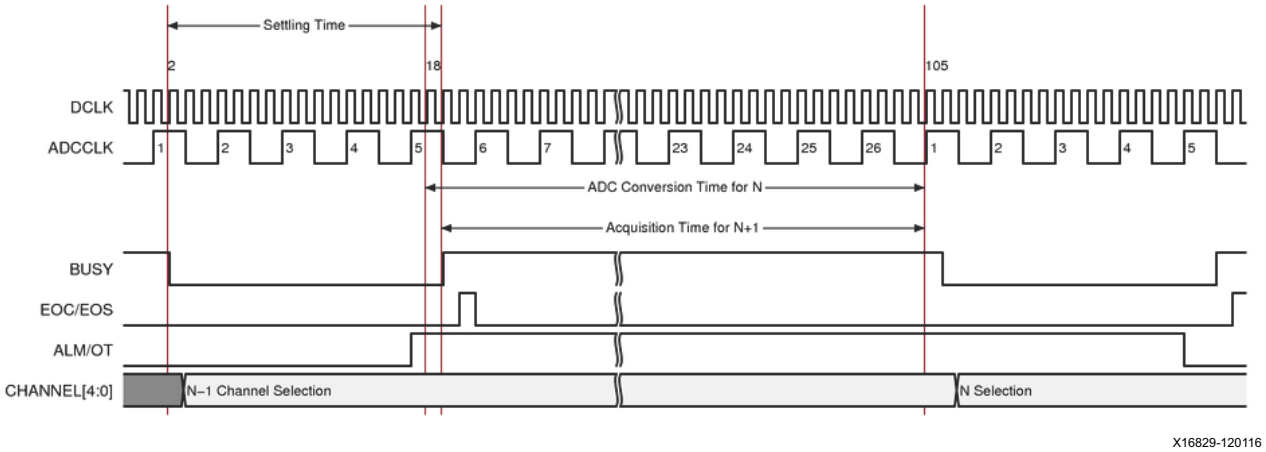


Figure 2-5: **Continuous Sampling Mode**

Event-Driven Sampling Mode

In the event-driven sampling mode, the CONVST or CONVSTCLK signal starts the conversion process. Consequently, the acquisition time varies based on when the previous conversion is completed, as shown in Figure 2-6.

Note: In event-driven sampling mode, the ACQ bit has no meaning because the sampling instant is controlled by CONVST/CONVSTCLK.

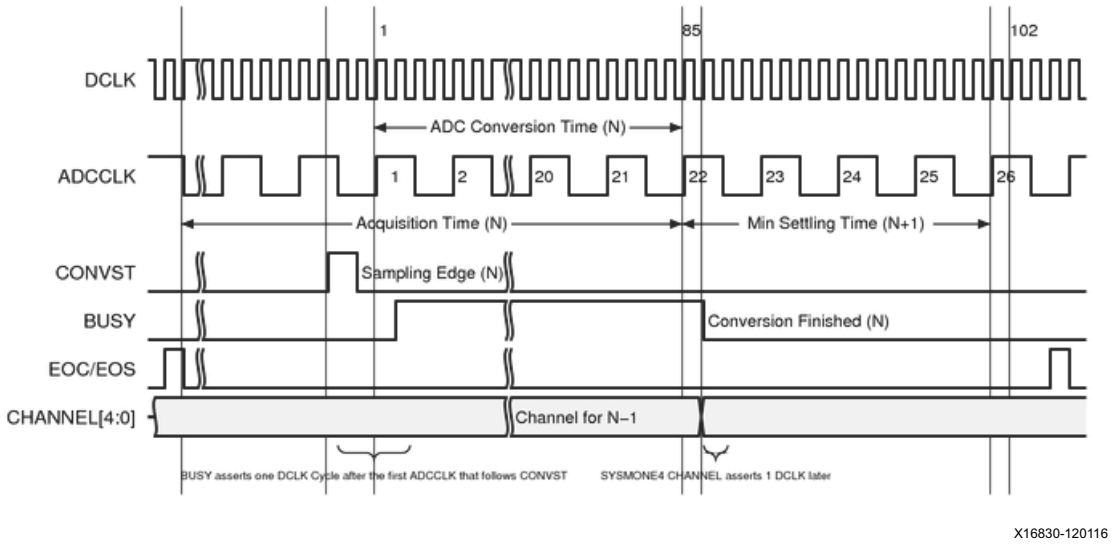


Figure 2-6: **Event-Driven Sampling Mode**

The event-driven sampling mode must provide enough time for the acquisition phase, the time between a channel change and the rising edge of CONVST or CONVSTCLK (the sampling time). The T/H starts to acquire the voltage on the next channel as soon as BUSY goes High and the conversion starts.

CONVST and CONVSTCLK are logically ORed within the SYSMON. If asynchronous, the SYSMON automatically resynchronizes the conversion process to the ADCCLK. The ADC cannot be interrupted until the conversion is completed and BUSY goes Low. 16 DCLK cycles after BUSY goes Low, EOC pulses High for one DCLK cycle when the conversion result has been transferred to the channel's status register.

EOS indicates the end of a sequence that depends on the automatic channel sequencer settings and averaging settings. If the automatic channel sequencer is used, then EOS matches the last channel enabled (see Table 4-1, page 81). When averaging is used, EOS only pulses High after all the sequences or samples have been completed (16, 64, and 256). The number of samples is set in configuration reg 0 (40H) by the values of AVG0, AVG1 as shown in Table 3-7, page 57.

CONVST/CONVSTCLK starts a single conversion. When using the automatic channel sequencer or averaging, the number of conversions are the product of the number of channels in a sequence and the number of samples being averaged.

Analog Input Description

Figure 2-7 illustrates an equivalent analog input circuit for the external analog input channels in both unipolar and bipolar configurations. The analog inputs consist of a sampling switch and a sampling capacitor used to acquire the analog input signal for conversion. During the ADC acquisition phase, the sampling switch is closed, and the sampling capacitor is charged up to the voltage on the analog input. The time needed to charge this capacitor to its final value (±0.5 LSBs at 10 bits) is determined by the capacitance of the sampling capacitor (C_{SAMPLE}), the resistance of the analog multiplexer circuit (R_{MUX}), and any external (source) impedance.

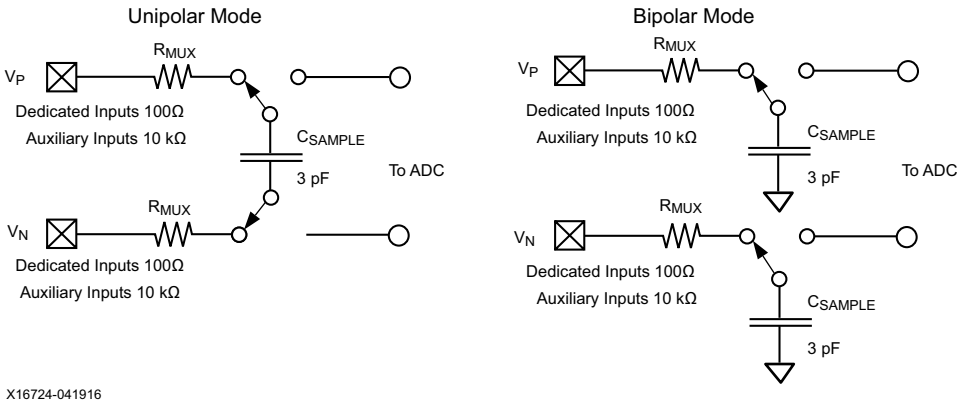


Figure 2-7: **Equivalent Analog Input Circuits (shown as SYSMONE1)**

The required 10-bit acquisition time (assuming no additional external or source resistance) in bipolar mode for example is approximated by:

$$t_{ACQ} = 7.6 \times R_{MUX} \times C_{SAMPLE} \quad \textit{Equation 2-1}$$

The time constant 7.6 is derived from TC = Ln $2^{(N + m)}$, where N = 10 for a 10-bit system and m = 1 additional resolution bit. The required 10-bit acquisition time in unipolar mode is approximated:

$$t_{ACQ} = 7.6 \times (R_{MUX} + R_{MUX}) \times C_{SAMPLE} \quad \textit{Equation 2-2}$$

For the dedicated channel (V_P/V_N), the minimum acquisition time (bipolar mode) required is given by:

$$t_{ACQ} = 7.6 \times 100 \times 3 \times 10^{-12} = 2.3\ ns \quad \textit{Equation 2-3}$$

The auxiliary analog channels (such as, VAUXP[15:0] and VAUXN[15:0]) have a much larger R_{MUX} resistance that is approximately equal to 10 kΩ. Equation 2-4 shows the minimum acquisition time in bipolar mode.

$$t_{ACQ} = 7.6 \times (10 \times 10^{3}) \times (3 \times 10^{-12}) = 230\ ns \quad \textit{Equation 2-4}$$

Table 2-1 summarizes the different input configurations and the results tACQ values.

Table 2-1: **Analog Input Configurations**

Analog Input Configuration	R_{MUX} [Ohms]	C_{SAMPLE} [F]	t_{ACQ} [sec]
Dedicated inputs, Unipolar (Kintex UltraScale FPGA and Virtex UltraScale FPGA)	100	$3x10^{-12}$	2.3×10^{-9}
Dedicated inputs, Bipolar (Kintex UltraScale FPGA and Virtex UltraScale FPGA)	100	$3x10^{-12}$	2.3×10^{-9}
Auxiliary inputs, Unipolar (Kintex UltraScale FPGA and Virtex UltraScale FPGA)	10000	$3x10^{-12}$	230×10^{-9}
Auxiliary inputs, Bipolar (Kintex UltraScale FPGA and Virtex UltraScale FPGA)	10000	$3x10^{-12}$	230×10^{-9}
Dedicated inputs, Unipolar (SYSMONE4)	100	$2x10^{-12}$	1.5×10^{-9}
Dedicated inputs, Bipolar (SYSMONE4)	100	$2x10^{-12}$	1.5×10^{-9}
Auxiliary inputs, Unipolar (SYSMONE4)	3000	$2x10^{-12}$	46×10^{-9}
Auxiliary inputs, Bipolar (SYSMONE4)	3000	$2x10^{-12}$	46×10^{-9}

Any additional external resistance, such as the anti-alias filter or resistor divider, increases the acquisition time requirement due to the increased R_{MUX} value in Equation 2-1. To calculate the new acquisition time, convert any external resistance to a series equivalent resistance value and add to the R_{MUX} resistance specified in Equation 2-3 and Equation 2-4. For more information and design considerations for driving the ADC inputs, see *Driving the Xilinx Analog-to-Digital Converter* (XAPP795) [Ref 5].

Unipolar Input Signals

When measuring unipolar analog input signals, the ADC must operate in a unipolar input mode. This mode is selected by writing to configuration register 0 (see Control Registers). When unipolar operation is enabled, the differential analog inputs (V_P and V_N) have an input range of 0V to 1.0V. In this mode, the voltage on V_P (measured with respect to V_N) must always be positive. Figure 2-8 shows a typical application of unipolar mode. The V_N input should always be driven by an external analog signal. V_N is typically connected to a local ground or common mode signal. The common mode signal on V_N can vary from 0V to +0.5V (measured with respect to GNDADC). Because the differential input range is from 0V to 1.0V (V_P to V_N), the maximum signal on V_P is 1.5V. Figure 2-8 shows the maximum signal levels on V_N and V_P in unipolar mode, measured with respect to analog ground (GNDADC package ball).

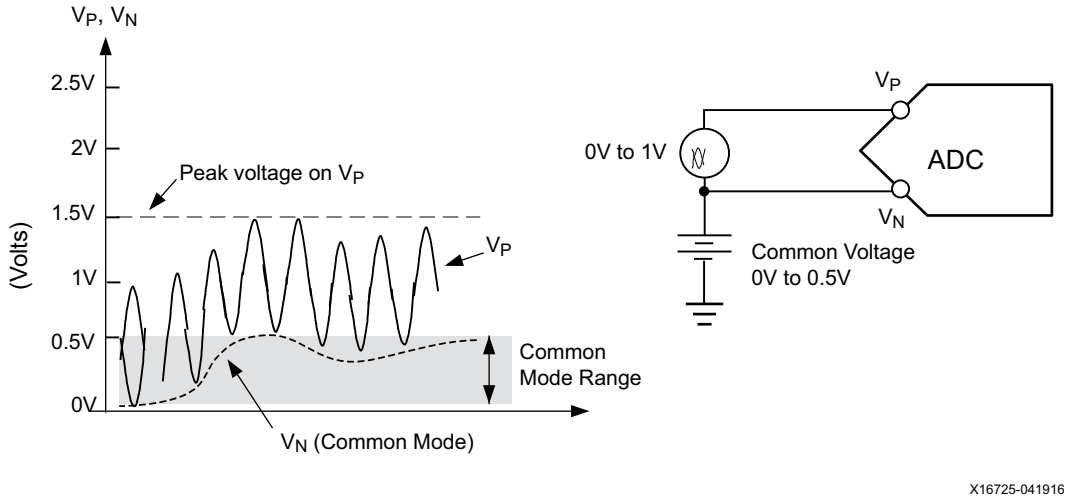


Figure 2-8: **Unipolar Input Signal Range**

Bipolar Input Signals

The analog inputs can accommodate analog input signals that are positive and negative with respect to a common mode or reference. To accommodate these types of signals, the analog input must be configured to bipolar mode. Bipolar mode is selected by writing to configuration register 0 (see Control Registers). All input voltages must be positive with respect to analog ground (GNDADC).

When bipolar operation is enabled, the differential analog input (V_P – V_N) can have a maximum input range of ±0.5V. The common mode or reference voltage should not exceed 0.5V in this case (see Figure 2-9).

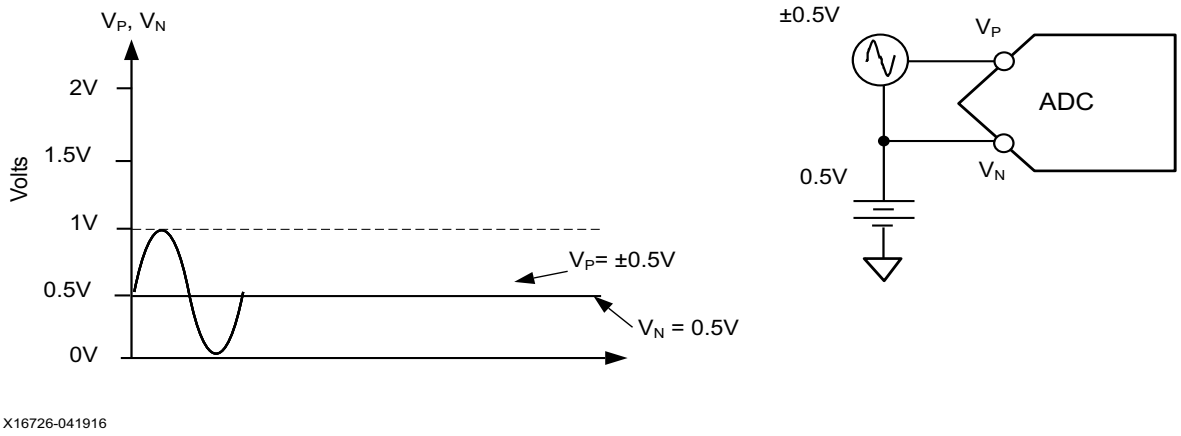


Figure 2-9: **Bipolar Input Signal Range**

The bipolar input mode also accommodates inputs signals driven from a true differential source, for example, a balanced bridge. In this case, V_N and V_P can swing positive and negative relative to a common mode or reference voltage (see Figure 2-10). The maximum differential input (V_P – V_N) is ±0.5V. With maximum differential input voltages of ±0.5V and assuming balanced inputs on V_N and V_P, the common mode voltage must lie in the range 0.25V to 0.75V.

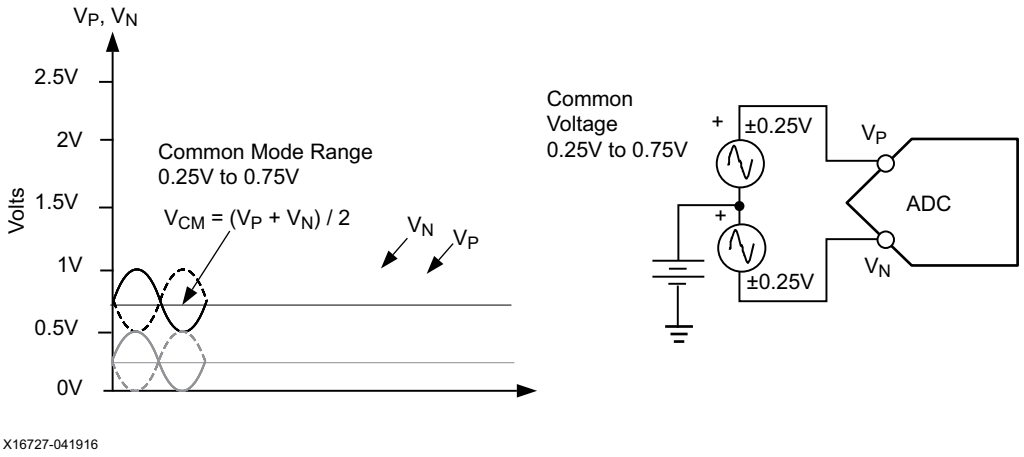


Figure 2-10: **Differential Input Signal Range**

Temperature Sensor

The SYSMON contains a temperature sensor that produces a voltage output proportional to the die temperature.

Voltage = 10 x kT/q x ln(10)

Where:

k = Boltzmann's constant = 1.3806 x 10^{-23} J/K

T = Temperature K (Kelvin) = °C + 273.15

q = Charge on an electron = 1.6022 x 10^{-19} C

SYSMONE1

In practice, the actual SYSMONE1 temperature transfer function depends on the architecture and reference source. For SYSMONE1, when using an external reference, the temperature sensor has a transfer function as shown in Equation 2-5.

$$Temperature(C) = \frac{ADC \times 502.9098}{2^{bits}} - 273.8195 \qquad \textit{Equation 2-5}$$

To calculate the ADC values for a given temperature value using the external reference, the equivalent calculation is shown in Equation 2-6.

$$ADC = \frac{(T + 273.8195) \times (2^{bits})}{502.9098} \qquad \textit{Equation 2-6}$$

For example, ADC code 608 (`260h`) = 25°C using the 10 MSBs. Using all 16 bits, this translates to 38940 (`981Ch`) when using the external reference. The temperature sensor result is found in the status register `00h`. For SYSMONE1, when using the on-chip reference, the transfer function is as shown in Equation 2-7.

$$Temperature(C) = \frac{ADC \times 501.3743}{2^{bits}} - 273.6777 \qquad \textit{Equation 2-7}$$

To calculate the ADC values for a given temperature value using the on-chip reference, the equivalent calculation is shown in Equation 2-8.

$$ADC = \frac{(T + 273.6777) \times (2^{bits})}{501.3743} \qquad \textit{Equation 2-8}$$

SYSMONE4

For SYSMONE4, when using an external reference, the temperature sensor has a transfer function as shown in Equation 2-9.

$$Temperature\,(C) = \left(\frac{ADC\_code \times 507.5921310}{2^{bits}}\right) - 279.42657680 \qquad \textit{Equation 2-9}$$

To calculate the ADC values for a given temperature value using the external reference, the equivalent calculation is shown in Equation 2-10.

$$ADC\_code = \frac{(T + 279.42657680) \times (2^{bits})}{507.5921310} \quad \text{Equation 2-10}$$

For SYSMONE4 (both PS and PL), when using an internal reference, the temperature sensor has a transfer function as shown in Equation 2-11.

$$Temperature\,(C) = \left(\frac{ADC\_code \times 509.3140064}{2^{bits}}\right) - 280.23087870 \quad \text{Equation 2-11}$$

To calculate the ADC values for a given temperature value using the internal reference, the equivalent calculation is shown in Equation 2-12.

$$ADC\_code = \frac{(T + 280.23087870) \times (2^{bits})}{509.3140064} \quad \text{Equation 2-12}$$

For SYSMONE4, this means that ADC code 615 (267h) = 25°C using the 10 MSBs with an external reference. Using all 16 bits, 25°C translates to 39305 (9989h).

Power Supply Sensor

The SYSMON also includes on-chip sensors that allow monitoring of the device power-supply voltages using the ADC. The sensors sample and attenuate the power supply voltages VUSER[3:0], V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, and V_{CC_PSAUX}. Supply voltages are attenuated by a factor of three. The exception is when V_{USER} is connected to a V_{CCO} supply in an HR I/O bank and the voltage is attenuated by a factor of six.

IMPORTANT: *The SYSMON measures supply rails at the die level, while the data sheet supply requirements are given at the package ball. Because the DC resistance through the package can cause a supply's level to drop after it reaches the SYSMON sensor, IR drop should be accounted for when setting alarm thresholds. For details on how to account for IR drop, see Xilinx Answer Record 75358.*

Figure 2-11 shows the power-supply sensor transfer function after digitizing by the ADC. The power supply sensor can be used to measure voltages in the range 0V to V_{CCAUX} + 3% with a resolution of approximately 2.93 mV. The transfer function for the supply sensor is shown in Equation 2-13.

$$Voltage = \frac{ADC\ Code}{1024} \times 3V \quad \text{Equation 2-13}$$

The transfer function can also be expressed as a 16-bit value:

$$Voltage = \frac{16\ bit\ ADC\ Code}{2^{16}} \times 3V \quad \text{Equation 2-14}$$

$$Voltage = \frac{16\ bit\ ADC\ Code}{65536} \times 3V \quad \text{Equation 2-15}$$

The power-supply measurement results for V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, and V_{CC_PSAUX} are stored in the status registers at DRP addresses `01h`, `02h`, `06h`, `0Dh`, `0Eh`, and `0Fh`, respectively.

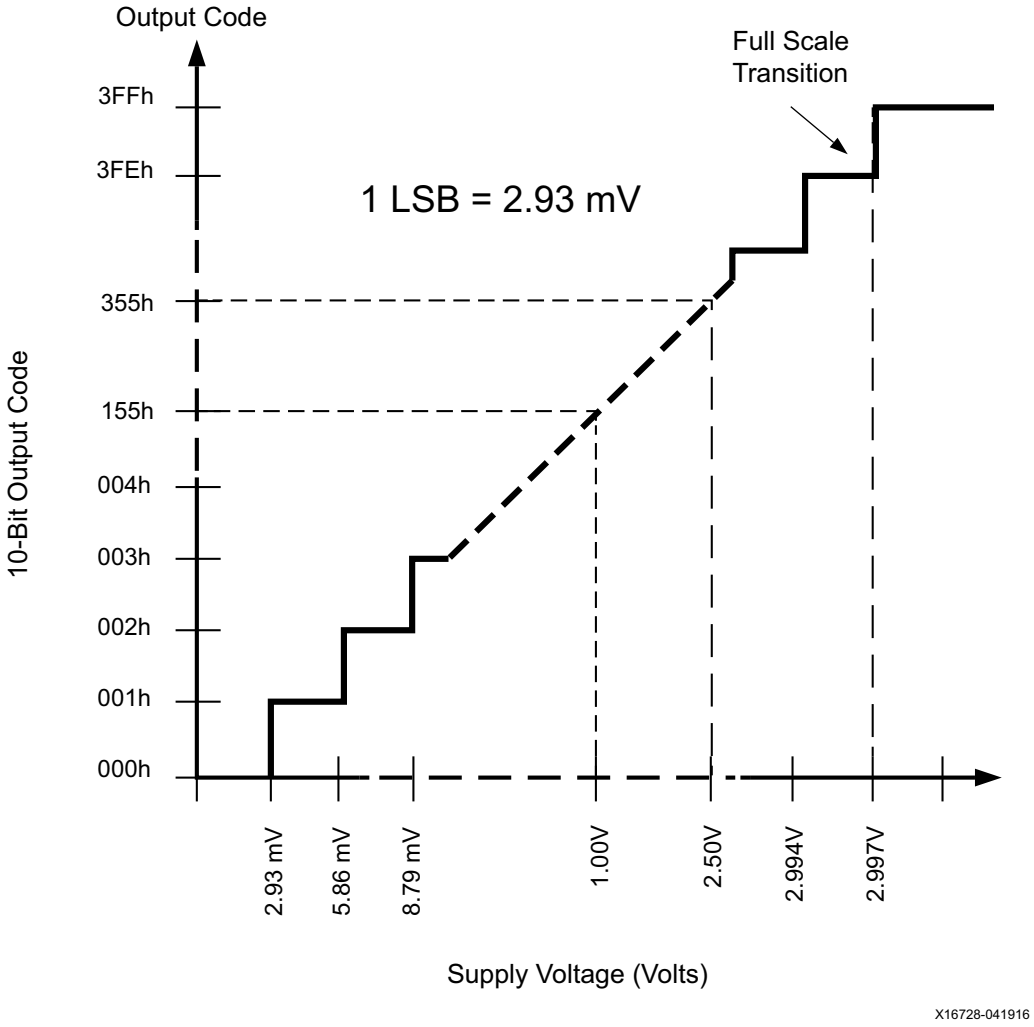


Figure 2-11: **Ideal Power Supply Transfer Function (All Supplies Excluding HR I/O Banks (SYSMONE1), HD I/O Banks (SYSMONE4), and V_{CCO_PSIO} (Zynq UltraScale+ MPSoC PS Block))**

The power supply measurement results for V_{USER0}, V_{USER1}, V_{USER2}, and V_{USER3} are stored in the status registers at DRP addresses `80h`, `81h`, `82h`, and `83h`, respectively. When the V_{USER} supply is attached to an HP I/O bank, the transfer function is:

$$Voltage = \frac{ADC\ Code}{1024} \times 3V \quad \textit{Equation 2-16}$$

The transfer function can also be expressed as a 16-bit value:

$$Voltage = \frac{16\ bit\ ADC\ Code}{2^{16}} \times 3V \quad \textit{Equation 2-17}$$

$$Voltage = \frac{16\ bit\ ADC\ Code}{65536} \times 3V \quad \textit{Equation 2-18}$$

To support wider voltage ranges, some of the voltages have been further attenuated. When the V_{USER} supply is attached to HR I/O banks (SYSMONE1) or HD I/O banks (SYSMONE4), or when the PS block is measuring V_{CCO_PSIO} (Zynq UltraScale+ MPSoC), the transfer function is attenuated, as shown in Equation 2-19. See Figure 2-12.

$$Voltage = \frac{ADC\ Code}{1024} \times 6V \quad \textit{Equation 2-19}$$

The transfer function can also be expressed as a 16-bit value:

$$Voltage = \frac{16\ bit\ ADC\ Code}{2^{16}} \times 6V \quad \textit{Equation 2-20}$$

$$Voltage = \frac{16\ bit\ ADC\ Code}{65536} \times 6V \quad \textit{Equation 2-21}$$

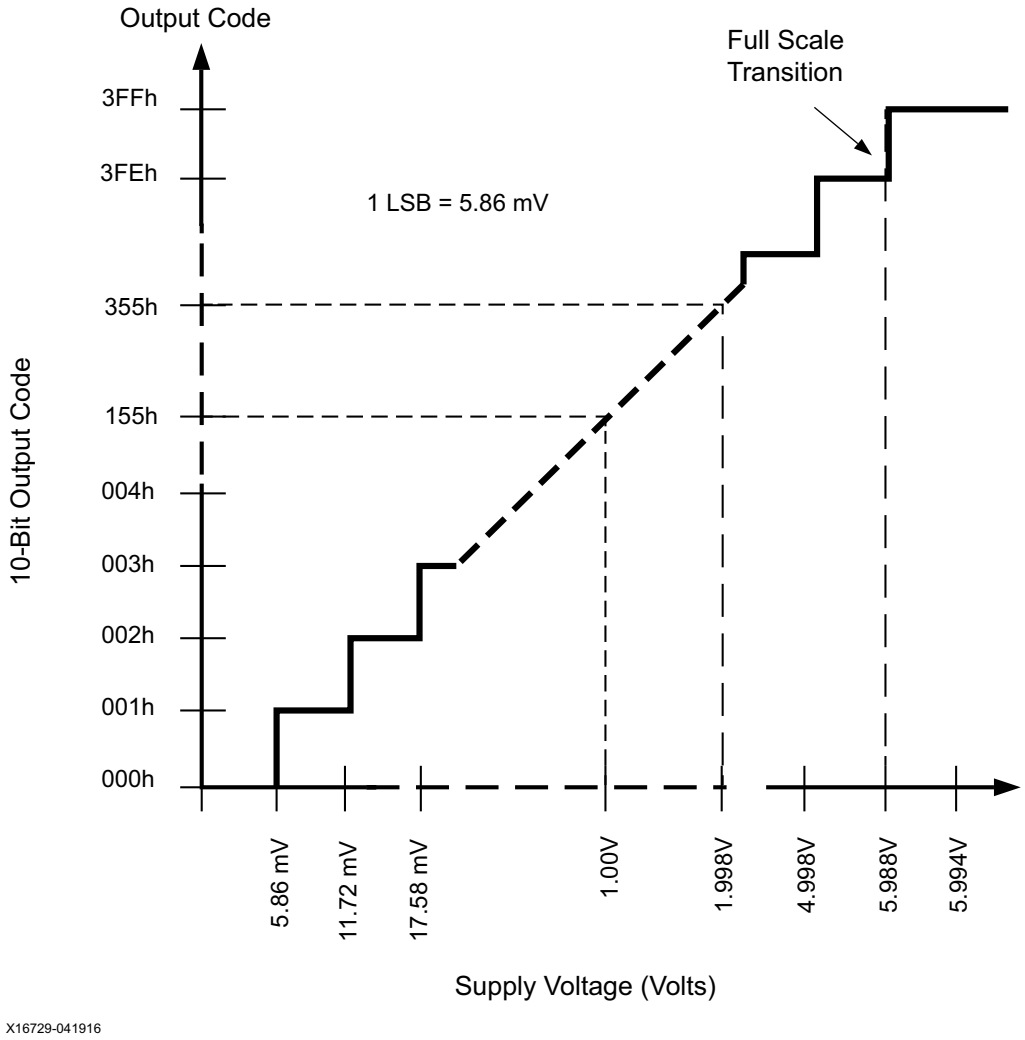


Figure 2-12: **Ideal Power Supply Transfer Function for USER Supply (HR I/O Banks (SYSMONE1), HD I/O Banks (SYSMONE4), and V_{CCO_PSIO} (Zynq UltraScale+ MPSoC PS Block))**

SYSMON Register Interface

Figure 3-1 shows the SYSMONE1 register interface and Figure 3-2 shows the SYSMONE4 register interface. All registers in the register interface are accessible through the dynamic reconfiguration port (DRP). The DRP can be accessed by the SYSMONE1 DRP interface, the I2C interface, or the JTAG TAP. Access is governed by an arbitrator (see DRP Arbitration). The DRP allows access up to 256 16-bit registers (DADDR[7:0] = `00h` to `FFh`). The access locations DADDR[7:0] = `00h` to `3Fh` and DADDR[7:0] = `80h` to `FFh` are read-only and contain the ADC measurement data. These registers are status registers. The control registers are located at addresses `40h` to `7Fh` and are readable or writable through the DRP.

IMPORTANT: *Not all registers apply to all blocks. For example, the SYSMON block within the PS does not have access to any of the auxiliary analog inputs.*

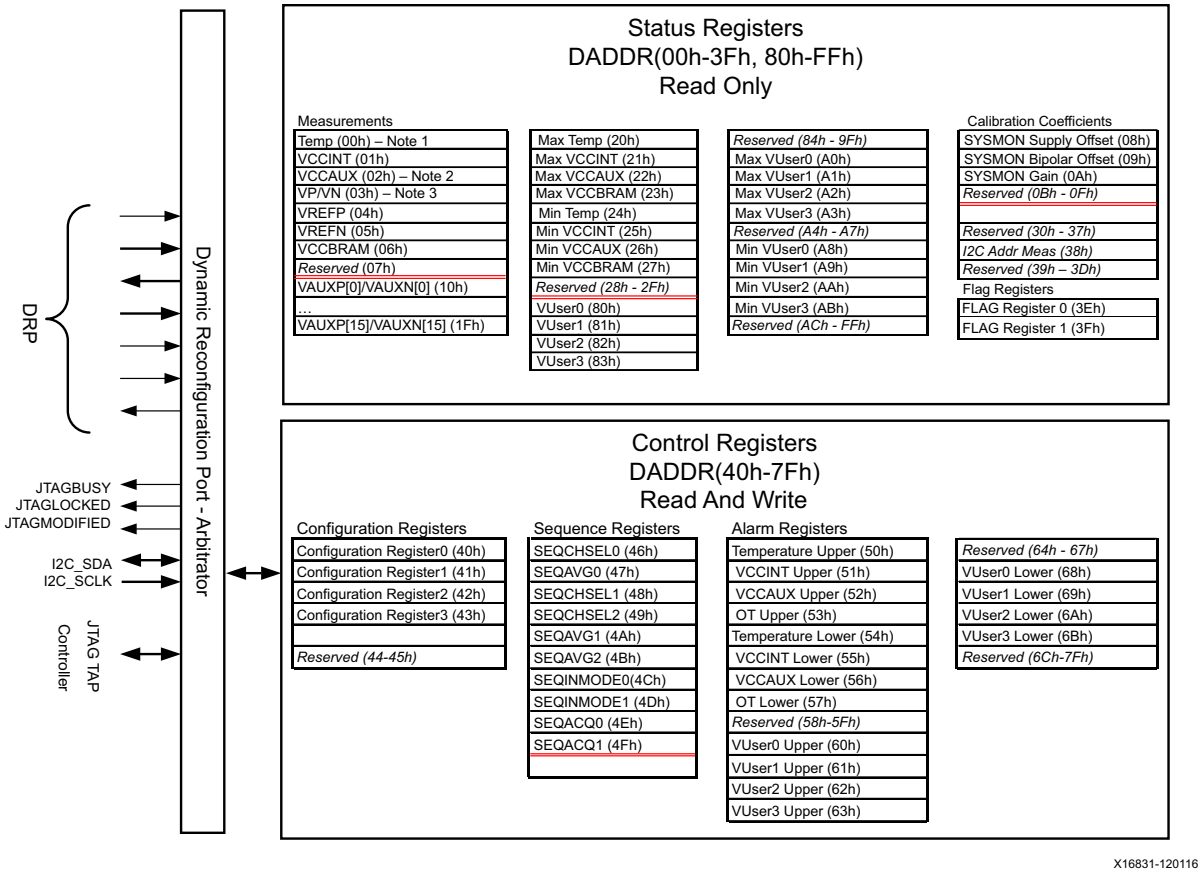


Figure 3-1: **SYSMONE1 Register Interface**

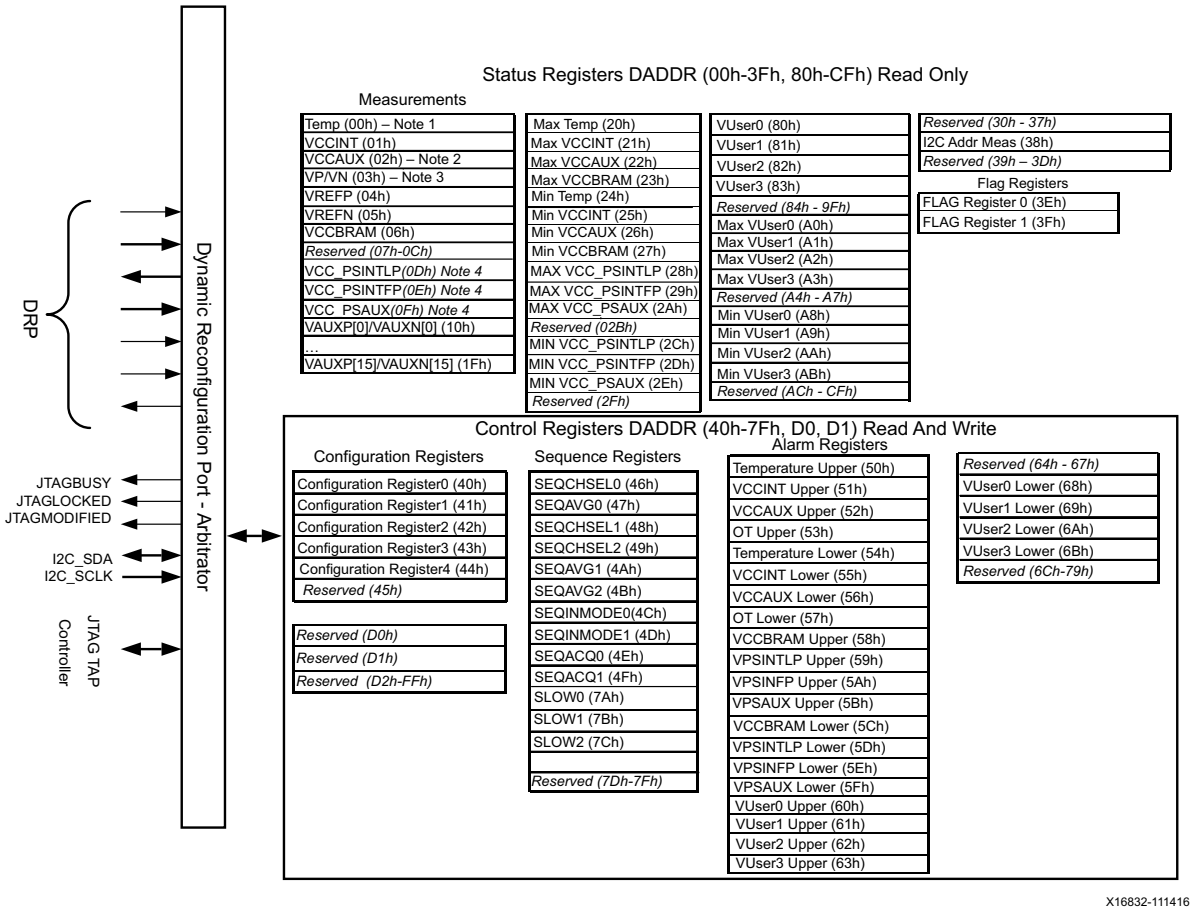


Figure 3-2: **SYSMONE4 Register Interface**

Notes relevant to Figure 3-1 and Figure 3-2:

1. Status Register `00h` is a shared address. Writing `0000h` to DADDR (`00h`) resets the JTAGLOCKED signal. Writing `0001h` to `0000h` activates the JTAGLOCKED signal.
2. Status Register `02h` is a shared address. Writing `0001h` to DADDR (`02h`) enables auxiliary channels before configuration (preconfiguration).
3. Status Register `03h` is a shared address. Writing any value (`xxxxh`) to DADDR (`03h`) has the same effect as pulsing the RESET pin.
4. For Zynq UltraScale+ MPSoC devices that support PS and the associated processor supplies.

For a detailed description of the DRP timing, see Dynamic Reconfiguration Port (DRP) Timing. For more information on the JTAG DRP interface, see DRP JTAG Interface.

Dynamic Reconfiguration Port (DRP) Timing

Figure 3-3 illustrates a DRP read and write operation. When the DEN is pulsed High for a single clock cycle, the DRP address (DADDR) and write enable (DWE) inputs are captured on the next rising edge of DCLK. DEN should only transition High for one DCLK period.

If DWE is a logic Low, a DRP read operation is carried out. The data for this read operation is valid on the DO bus when DRDY transitions High. Thus, DRDY should be used to capture the DO bus. For a write operation, the DWE signal is a logic High and the DI bus and DRP address (DADDR) is captured on the next rising edge of DCLK. The DRDY signal transitions to a logic High when the data has been successfully written to the DRP register. A new read or write operation cannot be initiated until the DRDY signal transitions Low.

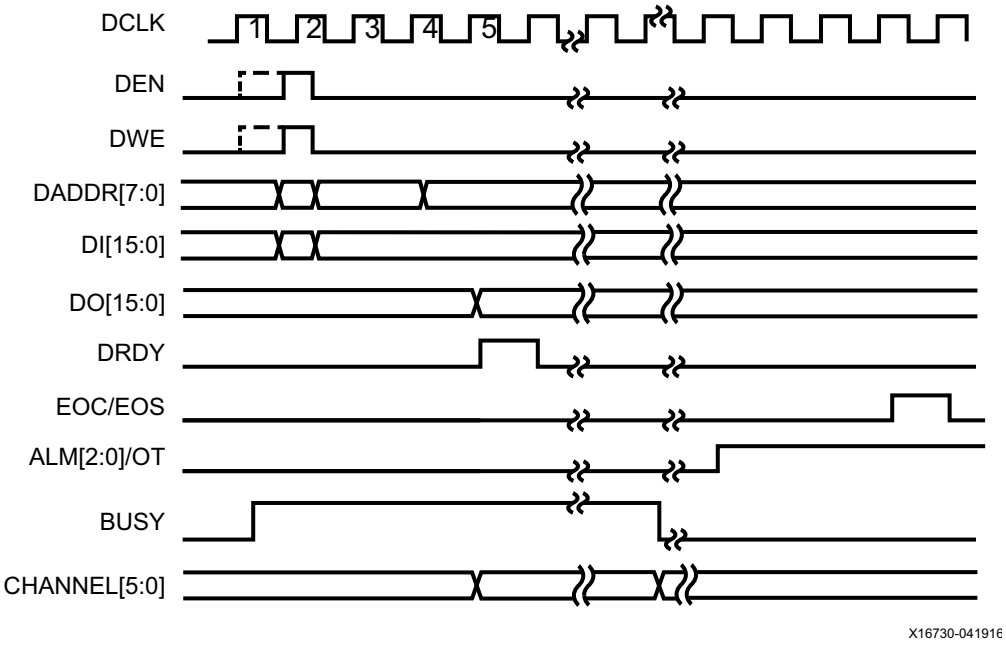


Figure 3-3: **DRP Detailed Timing**

Status Registers

The status registers (00h-3Fh, 80h-BFh) contain the measurement results of the analog-to-digital conversions, the flag registers, and the calibration coefficients as shown in Table 3-1.

Table 3-1: **Status Registers (Read Only)**

Name	Address	Description
Temperature	00h	The result of the on-chip temperature sensor measurement is stored in this location. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the temperature sensor transfer function shown in Figure 2-11.
V_{CCINT}	01h	The result of the on-chip V_{CCINT} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the supply sensor transfer function shown in Figure 2-11.
V_{CCAUX}	02h	The result of the on-chip V_{CCAUX} data supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the supply sensor transfer function shown in Figure 2-11.
V_P/V_N	03h	The result of a conversion on the dedicated analog input channel is stored in this register. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the transfer function shown in Figure 2-8 or Figure 2-9 depending on analog input mode settings.
V_{REFP}	04h	The result of a conversion on the reference input V_{REFP} is stored in this register. The 10 MSBs correspond to the ADC transfer function shown in Figure 2-11. The data is MSB justified in the 16-bit register. The supply sensor is used when measuring V_{REFP}.
V_{REFN}	05h	The result of a conversion on the reference input V_{REFN} is stored in this register. This channel is measured in bipolar mode with a two's complement output coding as shown in Figure 2-2. By measuring in bipolar mode, small positive and negative offset around 0V (V_{REFN}) can be measured. The supply sensor is used when measuring V_{REFN} so this channel has a bipolar range of ±1.5V.
V_{CCBRAM}	06h	The result of the on-chip V_{CCBRAM} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the supply sensor transfer function shown in Figure 2-11.
Reserved	07h	This location is reserved.
Supply Offset	08h	The calibration coefficient for the supply sensor offset using ADC is stored at this location. (SYSMONE1 only. Not used for SYSMONE4.)
ADC Offset	09h	The calibration coefficient for the ADC offset is stored at this location. (SYSMONE1 only. Not used for SYSMONE4.)
ADC Gain	0Ah	The calibration coefficient for the ADC gain error is stored at this location. (SYSMONE1 only. Not used for SYSMONE4.)
Undefined	0Bh to 0Ch	These locations are unused and contain invalid data.
$V_{CC\_PSINTLP}$	0Dh	On-chip $V_{CC\_PSINTLP}$ supply monitor measurement (Zynq UltraScale+ MPSoC).
$V_{CC\_PSINTFP}$	0Eh	On-chip $V_{CC\_PSINTFP}$ supply monitor measurement (Zynq UltraScale+ MPSoC).
V_{CC_PSAUX}	0Fh	On-chip V_{CC_PSAUX} supply monitor measurement (Zynq UltraScale+ MPSoC).

Table 3-1: **Status Registers (Read Only) *(Cont'd)***

Name	Address	Description
VAUXP[15:0]/ VAUXN[15:0]	10h to 1Fh	The results of the conversions on auxiliary analog input channels are stored in this register. The data is MSB justified in the 16-bit register. The 10 MSBs correspond to the transfer function shown in Figure 2-1 or Figure 2-2 depending on analog input mode settings.
Max Temp	20h	Maximum temperature measurement recorded since power-up or the last SYSMON reset.
Max V_{CCINT}	21h	Maximum V_{CCINT} measurement recorded since power-up or the last SYSMON reset.
Max V_{CCAUX}	22h	Maximum V_{CCAUX} measurement recorded since power-up or the last SYSMON reset.
Max V_{CCBRAM}	23h	Maximum V_{CCBRAM} measurement recorded since power-up or the last SYSMON reset.
Min Temp	24h	Minimum temperature measurement recorded since power-up or the last SYSMON reset.
Min V_{CCINT}	25h	Minimum V_{CCINT} measurement recorded since power-up or the last SYSMON reset.
Min V_{CCAUX}	26h	Minimum V_{CCAUX} measurement recorded since power-up or the last SYSMON reset.
Min V_{CCBRAM}	27h	Minimum V_{CCBRAM} measurement recorded since power-up or the last SYSMON reset.
MAX $V_{CC\_PSINTLP}$	28h	Maximum $V_{CC\_PSINTLP}$ measurement since power-up, or last SYSMON reset.
MAX $V_{CC\_PSINTFP}$	29h	Maximum $V_{CC\_PSINTFP}$ measurement since power-up, or last SYSMON reset.
MAX V_{CC_PSAUX}	2Ah	Maximum V_{CC_PSAUX} measurement since power-up, or last SYSMON reset.
Reserved	2Bh	Reserved.
MIN $V_{CC\_PSINTLP}$	2Ch	Minimum $V_{CC\_PSINTLP}$ measurement since power-up, or last SYSMON reset.
MIN $V_{CC\_PSINTFP}$	2Dh	Minimum $V_{CC\_PSINTFP}$ measurement since power-up, or last SYSMON reset.
MIN V_{CC_PSAUX}	2Eh	Minimum V_{CC_PSAUX} measurement since power-up, or last SYSMON reset.
Reserved	2Fh to 37h	Reserved.
I2C Addr Meas	38h	V_P/V_N measurement at power-up used for I2C address decoding. D[15:12] determines the default I2C address when I2C_OR is Low. See Table 3-20, page 75.
Reserved	39h to 3Dh	These locations are reserved.
Flag1, Flag0	3Eh to 3Fh	This register contains general status information (see Flag Register).
V_{USER0}	80h	The result of the on-chip V_{USER0} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register.
V_{USER1}	81h	The result of the on-chip V_{USER1} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register.
V_{USER2}	82h	The result of the on-chip V_{USER2} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register.

Table 3-1: **Status Registers (Read Only) *(Cont'd)***

Name	Address	Description
V_{USER3}	83h	The result of the on-chip V_{USER3} supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register.
Max V_{USER0}	A0h	Maximum V_{USER0} measurement recorded since power-up or the last SYSMON reset.
Max V_{USER1}	A1h	Maximum V_{USER1} measurement recorded since power-up or the last SYSMON reset.
Max V_{USER2}	A2h	Maximum V_{USER2} measurement recorded since power-up or the last SYSMON reset.
Max V_{USER3}	A3h	Maximum V_{USER3} measurement recorded since power-up or the last SYSMON reset.
Min V_{USER0}	A8h	Minimum V_{USER0} measurement recorded since power-up or the last SYSMON reset.
Min V_{USER1}	A9h	Minimum V_{USER1} measurement recorded since power-up or the last SYSMON reset.
Min V_{USER2}	AAh	Minimum V_{USER2} measurement recorded since power-up or the last SYSMON reset.
Min V_{USER3}	ABh	Minimum V_{USER3} measurement recorded since power-up or the last SYSMON reset.

Measurement Registers

Measurement results from the analog-to-digital conversions are stored as 16-bit results in the status registers. As shown in Figure 3-4, the 10-bit data corresponds to the 10 MSBs (most significant or left-most bits) in the 16-bit registers. The unreferenced LSBs can be used to minimize quantization effects or improve the resolution through averaging or filtering.

Maximum and minimum measurements are also recorded for the on-chip sensors from the device power-up or the last user reset of the SYSMON. Table 3-1 defines the status registers.

The SYSMON also tracks the minimum and maximum values recorded for the internal sensors since the last power-up or since the last reset of the SYSMON control logic (see Figure 3-1 and Table 3-1 for minimum/maximum register addresses.) On power-up or after reset, all minimum registers are set to FFFFh, and all maximum registers are set to 0000h. Each new measurement generated for an on-chip sensor is compared to the contents of its maximum and minimum registers. If the measured value is greater than the contents of its maximum registers, the measured value is written to the maximum register. Similarly, for the minimum register, if the measured value is less than the contents of its minimum register, the measured value is written to the minimum register. This check is carried out every time a measurement result is written to the status registers.

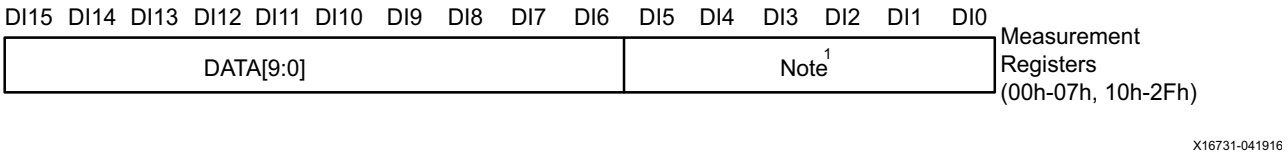


Figure 3-4: **Measurement Registers**

Notes:

1. The ADCs always produce a 16-bit conversion result, and the full 16-bit result is stored in the 16-bit status registers. The 10-bit data correspond to the 10 MSBs (most significant or left-most bits) in the 16-bit status registers. The unreferenced LSBs can be used to minimize quantization effects or improve resolution through averaging or filtering.

Flag Register

The flag register is shown in Figure 3-5. The bit definitions are described in Table 3-2.

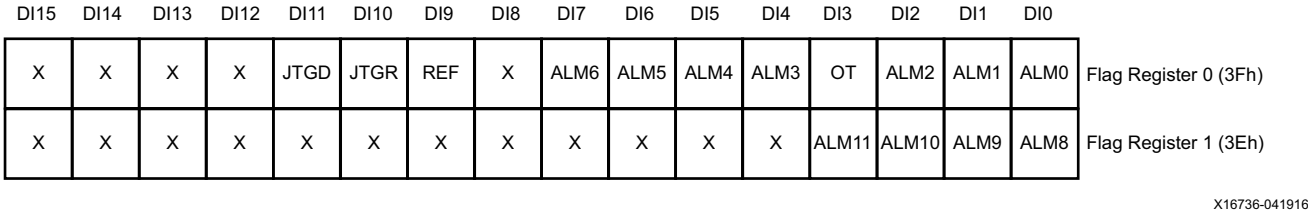


Figure 3-5: **Flag Registers**

Table 3-2: **Flag Register Bit Definitions**

Name	Description
ALM11 to ALM0	Indicates the status of the alarm outputs ALM[11:8, 2:0]
OT	Status of Over Temperature logic output
REF	Indicates system monitor ADC is using the internal voltage reference (High) or external reference (Low)
JTGR	A logic 1 indicates that the bitstream setting BITSTREAM.GENERAL.JTAG_SYSMON = STATUSONLY has been set to restrict access to read only. See DRP JTAG Interface for more information.
JTGD	A logic 1 indicates that the bitstream setting for SYSMON has been set to BITSTREAM.GENERAL.JTAG_SYSMON = DISABLE to disable all JTAG access. See DRP JTAG Interface for more information.

SYSMON Calibration Coefficients (SYSMONE1 only)

The SYSMON can digitally calibrate out any offset and gain errors in the ADC and power supply sensor using the calibration registers (see Figure 3-6). By connecting known voltages (V_{REFP} and V_{REFN} as opposed to the internal reference) to the ADC and the supply sensor, the offset and gain errors can be calculated and correction coefficients generated for optimal performance. The calibration coefficients for SYSMONE1 are stored in status registers `08h` to `0Ah` (see Table 3-1).

IMPORTANT: *The calibration coefficients only apply to SYSMONE1. SYSMONE4 calibration has changed and these coefficients no longer apply.*

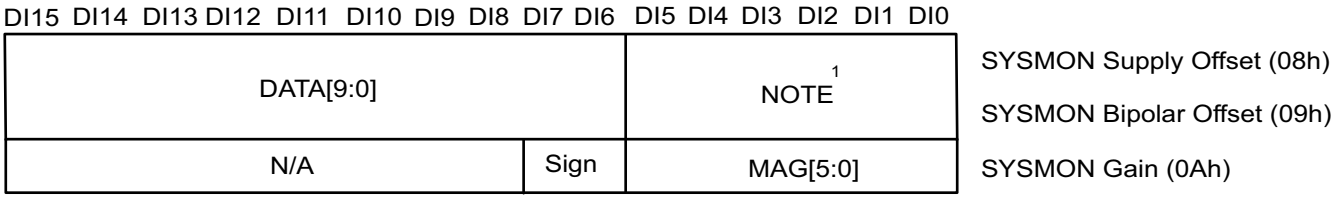


Figure 3-6: **Calibration Registers**

Notes:

1. The ADCs always produce a 16-bit conversion result. The 10-bit data correspond to the 10 MSBs in the 16-bit status registers. The unreferenced LSBs can be used to minimize quantization.

Table 3-3: **Calibration Register Bit Definitions**

Name	Description
CAL_OFFSET[9:0]	Offset correction factor for the supply sensor (unipolar mode) recorded in two's complement.
CAL_BIPOLAR_OFFSET[9:0]	Offset correction for the supply sensor (bipolar mode).
SIGN	Sign bit for calibration. Positive when 1 or negative when 0.
MAG[5:0]	Magnitude of calibration.

The SYSMON has a built-in calibration function that automatically calculates these coefficients. By initiating a conversion on channel 8 (08h), all calibration coefficients are calculated. The SYSMON *default* operating mode automatically uses calibration. When not operating in the default mode, these calibration coefficients are applied to all ADC measurements by enabling the calibration bits (CAL0–3) in configuration register 1 (41h) (see Table 3-6).

BUSY transitions High for the duration of the entire calibration sequence (conversion on channel 8). For the SYSMONE1, the calibration sequence is four times longer than a regular conversion on a sensor channel as offset and gain are measured for the ADC and the power supply sensor. For SYSMONE4, the calibration sequence is ten times longer.

Calibration Coefficients Definition

The offset and gain calibration coefficients are stored in the status registers. This section explains how to interpret the values in these registers. These are read-only registers, and the contents cannot be modified using the DRP.

Offset Coefficients

The offset calibration registers store the offset correction factor for the supply sensor and ADC. The offset correction factor is a 10-bit, two's complement number and is expressed in LSBs. Similar to other status registers, the 10-bit values are MSB justified in the registers. For example, if the ADC has an offset of +10 LSBs (approximately 10 x 977 µV = 9.77 mV), the offset coefficient records –10 LSBs or FF6h (status register 08h). For the supply sensor, the LSB size is approximately 2930 µV, thus a +10 LSB offset is equivalent to 29.3 mV of offset in the supply measurement.

Gain Coefficients

The ADC gain calibration coefficient stores the correction factor for any gain error in the ADC. The correction factor is stored in the seven LSBs of register 0Ah. These seven bits store both sign and magnitude information for the gain correction factor. If the seventh bit is a logic 1, the correction factor is positive. If it is 0, the correction factor is negative. The next six bits store the magnitude of the gain correction factor. Each bit is equivalent to 0.1%.

For example, if the ADC has a positive gain error of +1%, then the gain calibration coefficient records –1% (the –1% correction applied to cancel the +1% error). Because the correction factor is negative, the seventh bit is set to zero. The remaining magnitude bits record 1%, where 1% = 10 x 0.1% and 10 = 001010 binary. The status register 0Ah records 0000 0000 0000 1010. With six bits assigned to the magnitude and a maximum value of 3Fh, the calibration can correct errors in the range of ±0.1% x 63 = ±6.3%.

Control Registers

The SYSMON control registers are used to configure the SYSMON operation. All SYSMON functionality is controlled through these registers.

These control registers are initialized using the SYSMON attributes when the SYSMON is instantiated in a design. This means that the SYSMON can be configured to start in a predefined mode after device configuration.

Table 3-4: **SYSMON Control Registers**

Name	Address	SYSMONE1 Attribute(1)	Description
Configuration Registers	40h to 44h	INIT_40 to INIT_44	These are SYSMON configuration registers (see Configuration Registers (40h to 44h)).
Sequence registers	46h to 4Fh	INIT_46 to INIT_4F	These registers are used to program the channel sequencer function (see SYSMON Operating Modes in Chapter 4).
Alarm registers	50h to 6Fh	INIT_50 to INIT_6F	These are the alarm threshold registers for the SYSMON alarm function (see Automatic Alarms).

Notes:

1. SYSMONE1 attributes set SYSMONE1 operation after configuration is completed.

Configuration Registers (40h to 44h)

The SYSMON configuration registers are the first five registers in the control register block, and are used to configure the SYSMON operating modes. The configuration register bit definitions are listed in Figure 3-7.

IMPORTANT: *Bits shown as **0** should always be set to **0**.*

The configuration registers can be modified through the DRP after the device has been configured. For example, a soft microprocessor or state machine can be used to alter the contents of the SYSMON control registers at any time during normal operation. Table 3-5 through Table 3-7 define the bits for the configuration registers.

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
CAVG	X	AVG1	AVG0	MUX	BU	EC	ACQ	X	X	CH5	CH4	CH3	CH2	CH1	CH0	Config Reg0 (40h)
SEQ3	SEQ2	SEQ1	SEQ0	ALM6	ALM5	ALM4	ALM3	CAL3	CAL2	0	CAL0	ALM2	ALM1	ALM0	OT	Config Reg1 (41h)
CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	X	X	X	0	0	0	0	0	Config Reg2 (42h)
	I2C_A[6]	I2C_A[5]	I2C_A[4]	I2C_A[3]	I2C_A[2]	I2C_A[1]	I2C_A[0]	I2C_EN	X	X	X	ALM11	ALM10	ALM9	ALM8	Config Reg3 (43h)
X	X	X	X	SLOW_EOS[1]	SLOW_EOS[0]	SLOW_SEQ1	SLOW_SEQ0	X	X	X	X	PMBUS_HRIO[3]	PMBUS_HRIO[2]	PMBUS_HRIO[1]	PMBUS_HRIO[0]	Config Reg4 (44h)

X16833-081516

Figure 3-7: **Configuration Registers Bit Definitions**

Table 3-5: **Configuration Register Bit Definitions**

Name	Description
CH5 to CH0	When operating in Single Channel mode or External Multiplexer mode, these bits are used to select the ADC input channel. See Table 3-6.
ACQ	Four ADCCLK cycles (Low) or ten ADCCLK cycles (High). See ADC Channel Settling Time (4Eh, 4Fh) for controlling the acquisition times using the automatic channel sequencer. In default mode, the acquisition time cannot be adjusted.
BU	In Single Channel mode, selects Unipolar (Low) or Bipolar (High) operating mode for the ADC analog inputs (see Analog Inputs).
EC	Selects Continuous (Low) or Event (High) driven sampling mode for the ADC (see Adjusting the Acquisition Settling Time).
MUX	Enables (High) external multiplexer mode. See Chapter 4, SYSMON Operating Modes for more information.
AVG1, AVG0	Sets the amount of sample averaging on selected channels in both Single Channel and Sequence modes (see Table 3-7).
CAVG	Disables (High) averaging for the calculation of the calibration coefficients. Averaging is enabled by default (Low). Averaging is fixed at 16 samples. (SYSMONE1 only, calibration averaging always occurs in SYSMONE4.)
OT	Disables (High) the Over-Temperature signal.
ALM0 to ALM6 and ALM8 to ALM11	Disables (High) individual alarm outputs for the corresponding alarm.
SEQ0 to, SEQ3	Enables (High) the channel-sequencer function (see Table 3-8).

Table 3-5: **Configuration Register Bit Definitions *(Cont'd)***

Name	Description
CAL0, CAL2, CAL3	Enables (High) the application of the calibration coefficients to the ADC and on-chip supply sensor measurements. A logic 1 enables calibration and a logic 0 disables calibration. CAL1 is not supported and must be set to 0. (SYSMONE1 only).
CD7 to CD0	Selects the division ratio between the DRP clock (DCLK) and the lower frequency ADC clock (ADCCLK) used for the ADC (Dynamic Reconfiguration Port (DRP) Timing). See Table 3-10 for DCLK division values.
I2C_EN	I2C Enable. When High allows I2C interface to be used after configuration. (SYSMONE1 only).
I2C_A[6:0]	I2C address used only when I2C_OR is High. For SYSMONE4, I2C_A[2] controls the I2C or PMBus functionality. When High, PMBus is used and when Low, I2C is used.
I2C_OR	I2C address override. When High, I2C address is based on I2C_A[6:0]. When Low, I2C address is determined at power-up (38h) by the four MSBs of the dedicated analog input channel (Vp/Vn) as shown in Table 3-20.
SLOW_SEQ[1:0]	When using slow sequence mode, determines the divided down conversion rate for low-rate channels. (SYSMONE4 only). 00 - every sequence 01 - every 4th sequence 10 - every 16th sequence 11 - every 64th sequence
PMBUS_HRIO[3:0]	Automatically set by the Vivado design tools when SYSMON_VUSER[3:0]_MONITOR is either V_{CCO_TOP} or V_{CCO_BOT}. When High, the LINEAR16 transfer function compensates for the higher voltage ranges of V_{CCO_TOP} or V_{CCO_BOT} (i.e., 0-6V). For all other banks and supplies, leave Low. (SYSMONE4 only).
SLOW_EOS[1:0]	Control generation of EOS relative to sequencer (SEQCHSEL[2:0], 46h, 48h, 49h) and slow sequence (SLOWCHSEL[2:0], 7Ah, 7Bh, 7Ch). (SYSMONE4 only). 00 – End of sequence from SEQCHSEL 01 – End of sequence from SLOWCHSEL 10 – End of sequence from SEQCHSEL and SLOWCHSEL 11 - Default - End of sequence from SEQCHSEL

Table 3-6: **ADC Channel Select**

ADC Channel	CH5	CH4	CH3	CH2	CH1	CH0	Description
0	0	0	0	0	0	0	On-chip temperature
1	0	0	0	0	0	1	Average on-chip V_{CCINT}
2	0	0	0	0	1	0	Average on-chip V_{CCAUX}
3	0	0	0	0	1	1	V_P, V_N - Dedicated analog inputs
4	0	0	0	1	0	0	V_{REFP} (1.25V)
5	0	0	0	1	0	1	V_{REFN} (0V)
6	0	0	0	1	1	0	Average on-chip V_{CCBRAM}
7	0	0	0	1	1	1	Invalid channel selection
8	0	0	1	0	0	0	Carry out a SYSMON calibration
12–9	0	...	...	...	...	...	Invalid channel selection
13	0	0	1	1	0	1	VCC_PSINTLP (Zynq UltraScale+ MPSoC only)
14	0	0	1	1	1	0	VCC_PSINTFP (Zynq UltraScale+ MPSoC only)
15	0	0	1	1	1	1	VCC_PSAUX (Zynq UltraScale+ MPSoC only)
16	0	1	0	0	0	0	VAUXP[0], VAUXN[0] – Auxiliary channel 1
17	0	1	0	0	0	1	VAUXP[1], VAUXN[1] – Auxiliary channel 2
31–18	0	...	...	...	...	...	VAUXP[2:15], VAUXN[2:15] – Auxiliary channels 3 to 16
32	1	0	0	0	0	0	V_{USER0} User Supply 0
33	1	0	0	0	0	1	V_{USER1} User Supply 1
34	1	0	0	0	1	0	V_{USER2} User Supply 2
35	1	0	0	0	1	1	V_{USER3} User Supply 3
36+	1	...	...	...	...	...	Invalid channel selection

Table 3-7: **Averaging Filter Settings**

AVG1	AVG0	Function
0	0	No averaging
0	1	Average 16 samples
1	0	Average 64 samples
1	1	Average 256 samples

Table 3-8: **Sequencer Operation Settings**

SEQ3	SEQ2	SEQ1	SEQ0	Function
0	0	0	0	Default mode
0	0	0	1	Single pass sequence
0	0	1	0	Continuous sequence mode
0	0	1	1	Single channel mode (sequencer off)
1	1	X	X	Default mode

Table 3-9: **Calibration Enables (SYSMONE1 only)**

Name	Description
CAL0	ADC offset correction enable
CAL1	Reserved. Set Low.
CAL2	Supply sensor offset correction enable
CAL3	Supply sensor offset and gain correction enable

Table 3-10: **DCLK Division Selections**(1)

CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	Division
0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

Notes:
1. Minimum division ratio is 2, for example, ADCCLK = DCLK/2.

Channel Sequencer Registers (46h to 4Fh)

These registers are used to program the channel sequencer functionality. For more information, see Automatic Channel Sequencer.

Alarm Registers (50h to 6Fh)

These registers are used to program the alarm thresholds for the automatic alarms. For more information, see Automatic Alarms.

DRP Arbitration

Because the DRP registers are accessed from three different ports (SYSMONE1 DRP interface, I2C, and JTAG TAP), an arbitrator is implemented to manage potential conflicts. Arbitration is managed on a per transaction basis (a transaction is a single read/write operation to the DRP).

Three status signals help manage access through the interconnect when the JTAG or I2C port is also being used: JTAGBUSY, JTAGMODIFIED, and JTAGLOCKED.

DRP JTAG Interface

The SYSMON uses a full JTAG interface extension to the DRP interface. This allows read/write access to the SYSMON DRP through the existing on-chip JTAG infrastructure. No instantiation is required to access the DRP interface over JTAG. A boundary-scan instruction (6-bit instruction = `110111`) called SYSMON_DRP, added to UltraScale architecture-based devices, allows access to the DRP through the JTAG TAP. For Zynq UltraScale+ MPSoC, the boundary-scan instruction is 12-bit = `111111110111` (`FF7h`). All SYSMON JTAG instructions are 32 bits wide. For more information on the boundary-scan instructions and usage, see the *UltraScale Architecture Configuration User Guide* (UG570) [Ref 4]. Read and write operations using the SYSMON JTAG DRP interface are described in the next sections.

RECOMMENDED: *If you are unfamiliar with basic JTAG functionality, you should become familiar with the JTAG standard (IEEE standard 1149.1) before proceeding.*

IMPORTANT: *JTAG access can be limited to read only or completely disabled. To adjust the JTAG access, add the following to an XDC file:*

set_property BITSTREAM.GENERAL.JTAG_SYSMON <ENABLE|DISABLE|STATUSONLY> [current_design]

See the Vivado Design Suite User Guide: Programming and Debugging (UG908) [Ref 3] for more information on device configuration bitstream settings.

SYSMON DRP JTAG Write Operation

Figure 3-8 shows a timing diagram for a write operation to the SYSMON DRP through the JTAG TAP. The DRP is accessed through the SYSMON data register (SYSMON DR). Before the SYSMON DR is accessed, the instruction register (IR) must first be loaded with the SYSMON instruction. The controller is placed in the IR-scan mode, and the SYSMON instruction is shifted to the IR.

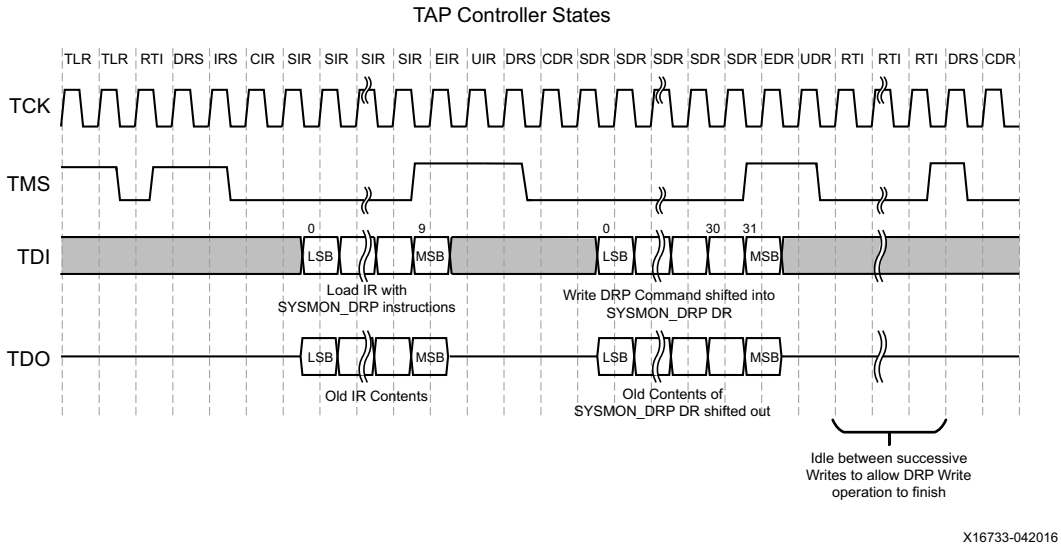


Figure 3-8: **SYSMON JTAG DRP Write**

After the SYSMON instruction is loaded, all data register (DR)-scan operations are carried out on the SYSMON DR. When the data shifted into SYSMON DR is a JTAG DRP write command, the SYSMON DRP arbitrator carries out a DRP write. The format of this write command is described in JTAG DRP Commands. The SYSMON DR contents are transferred to the SYSMON DRP arbitrator during the Update-DR state. After the Update-DR state, the arbitrator manages the new data transfer to the SYSMON DRP register. This takes up to 18 DRP clock (DCLK) cycles if a DRP access from the interconnect logic is already in progress.

During the Capture-DR phase (just before data is shifted into the SYSMON DR), DRP data is captured from the arbitrator. Depending on the last JTAG DRP command, this data can be old data, previously written to the DRP, or requested new read data (see SYSMON DRP JTAG Read Operation). This captured data is shifted out (LSB first) on DO as the new JTAG DRP command is shifted in. The 16 LSBs of this 32-bit word contain the JTAG DRP data. The 16 MSBs are set to zero.

If multiple writes to the SYSMON DR are occurring, it might be necessary to idle the TAP controller for several TCK cycles by adding RTI states before advancing to the next write operation (see Figure 3-6). Equation 3-1 shows the calculation to determine the required number of RTI states. From Equation 3-1, to ensure that RTI idle states are not needed: $F_{DRP\_DCLK} > 6 \times F_{JTAG\_TCK}$.

$$\frac{F_{DRP\_DCLK}}{18} > \frac{F_{JTAG\_TCK}}{3 + RTI} \qquad \textit{Equation 3-1}$$

Such that:

RTI = Required number of additional RTI states to ensure arbitration has fully resolved

F_{JTAG_TCK} = Frequency of TCK used for JTAG

F_{DRP_DCLK} = Frequency of DCLK used for SYSMON DRP interface

SYSMON DRP JTAG Read Operation

Figure 3-9 shows the timing for an SYSMON DR read operation. The IR should contain the DR-scan operation (SYSMON_DRP instruction). A JTAG read from the SYSMON DRP is a two-step operation.

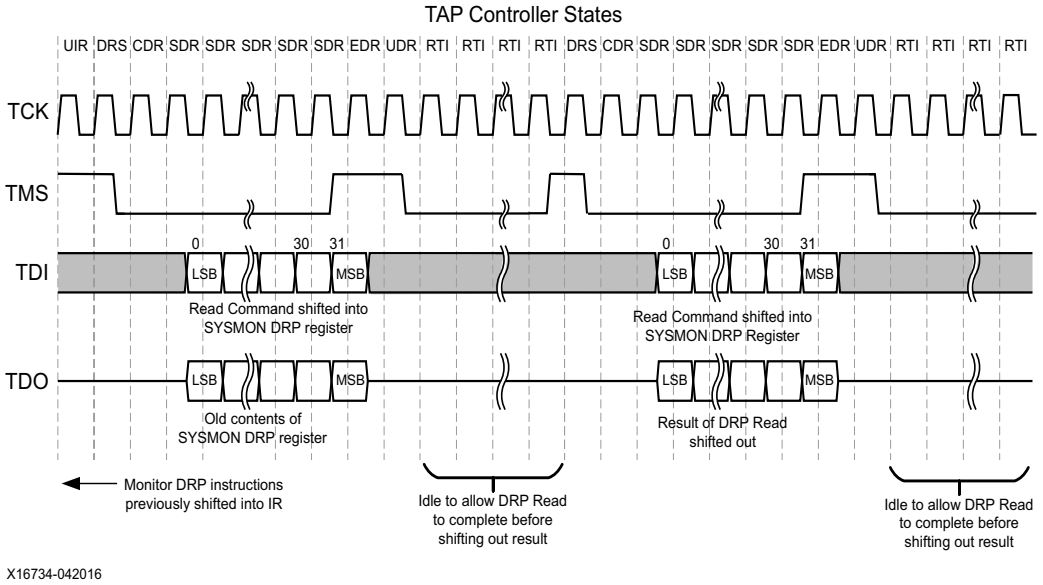


Figure 3-9: **SYSMON JTAG DRP Read**

First, the SYSMON DR is loaded with the read DRP instruction. This instruction is transferred to the arbitrator during the Update-DR state. Then the arbitrator reads the selected DRP register and stores the newly read 16-bit data. This operation takes several DCLK cycles to complete.

During the DR-Capture phase of the next DR-scan operation, newly read data is transferred from the arbitrator to the SYSMON DR. This 16-bit data (stored in the 16 LSBs of the 32-bit word) is then shifted out on TDO during the subsequent shift operation (see Figure 3-9). The timing diagram shows several idle states at the end of the first DR-scan operation, allowing the arbitrator enough time to fetch the SYSMON DRP data.

However, if the DCLK frequency is significantly faster than the TCK, these idle states might not be required.

Implementing a DR-scan operation before the arbitrator has completed the DRP-read operation results in old DRP data being transferred to the SYSMON DR during the DR-capture phase.

To ensure reliable operation over all operating clock frequencies, a minimum of 10 run-test-idle (RTI) states should be inserted. Multiple read operations can be pipelined, as shown in Figure 3-9. Thus, as the result of a read operation is being shifted out of the SYSMON DR, an instruction for the next read can be shifted in.

JTAG DRP Commands

The data shifted into the 32-bit SYSMON DR during a DR-scan operation instructs the arbitrator to carry out a write, read, or no operation on the SYSMON DRP. Figure 3-10 shows the data format of the JTAG DRP command loaded into the SYSMON DR. The first 16 LSBs of SYSMON DR [15:0] contain the DRP register data. For both read and write operations, the address bits SYSMON DR [25:16] hold the DRP target register address. The command bits SYSMON DR [29:26] specify a read, write, or no operation (see Table 3-11).

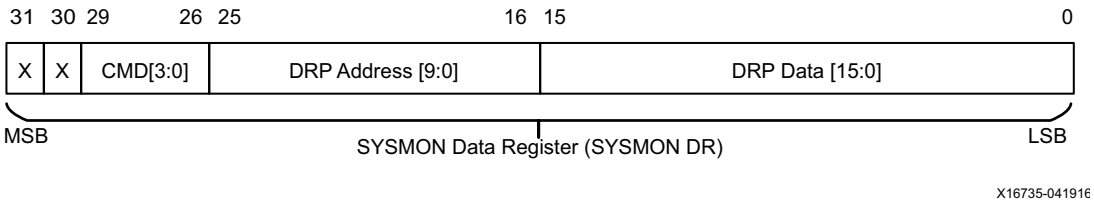


Figure 3-10: **SYSMON JTAG DRP Command**

Table 3-11: **JTAG DRP Commands**

CMD[3:0]				Operation
0	0	0	0	No operation
0	0	0	1	DRP read
0	0	1	0	DRP write
-	-	-	-	Not defined

It is also possible to enable the auxiliary analog input channel preconfiguration of the device, allowing external analog voltages (on the PCB) to be monitored using the JTAG TAP before configuration. The auxiliary channels are enabled by writing `0001h` to DRP address `02h`. This address lies within the read-only status register address space and normally holds the result of a V_{CCAUX} measurement. However, a write to this address enables the auxiliary inputs. This function only works prior to configuration. After configuration, these inputs must be explicitly instantiated in the design.

JTAGBUSY

JTAGBUSY becomes active during the update phase of a DRP transaction through the JTAG TAP. This signal resets when the JTAG SYSMON DR transaction is completed. Each read/write to the SYSMON DR is treated as an individual transaction. If DRP access initiates through the interconnect port when JTAGBUSY is High, then the arbitrator queues this request for a read/write through the interconnect logic. DRDY does not transition active until JTAGBUSY transitions Low and the interconnect transaction is completed. A second DRP access through the interconnect logic must not be initiated until the DRDY for the initial access becomes active and indicates the read/write was successful. If an interconnect access is in progress when a JTAG DRP transaction initiates, the interconnect access is completed before the JTAG transaction.

JTAGMODIFIED

Whenever there is a JTAG write (JTAG reads typically occur more often) to any register in the DRP, the application (device) must be notified about the potential change of configuration. Thus, the JTAGMODIFIED signal transitions High after a JTAG write. A subsequent DRP read/write resets the signal.

JTAGLOCKED

When JTAG is used, in some cases it is simpler to take DRP ownership for a period by locking out access through the interconnect. This is useful in a diagnostic situation where a large number of DRP registers are modified through the JTAG TAP. When a JTAGLOCKED request is made, the JTAGLOCKED signal transitions to the active-High state. The signal remains High until the port is unlocked again. No read or write access is possible via the DRP port when the JTAGLOCKED signal is High. The JTAGLOCKED signal is activated by writing `0001h` to DRP address `00h`. The JTAGLOCKED signal is reset by writing `0000h` to DRP address `00h`.

JTAGLOCKED is also used to indicate when the DRP is ready for a read or write when the DCLK is first connected or when DCLK becomes active again after a period of inactivity. It can take up to 18 DCLK cycles for JTAGLOCKED to deassert Low after DCLK becomes active.

TIP: *The SYSMON automatically switches over to an on-chip clock oscillator if a missing DCLK is detected.*

SYSMON JTAG Reset

A user reset of the SYSMON can also be initiated using the JTAG interface. The SYSMON is reset by writing `xxxxh` (any 16-bit value) to DRP address `03h`. The JTAG reset has the same effect as pulsing the RESET pin.

DRP I2C Interface

SYSMONE1 can be addressed as an I2C slave device allowing read/write access to the SYSMONE1 DRP interface. I2C is a standardized 2-wire bus that is commonly used by device manufacturers.

IMPORTANT: *I2C pins are active preconfiguration to allow access to SYSMON. Because I2C_SCLK and I2C_SDA are bidirectional, all I2C signals might drive out before the device has been configured.*

SYSMONE1 supports transfers up to 400 Kb/s, Standard-mode (Sm) and Fast-mode (Fm). For slow interfaces, clock stretching is supported at the bit level, which means that the I2C_SCLK low pulse will be extended if the I2C_SDA setup times are not met.

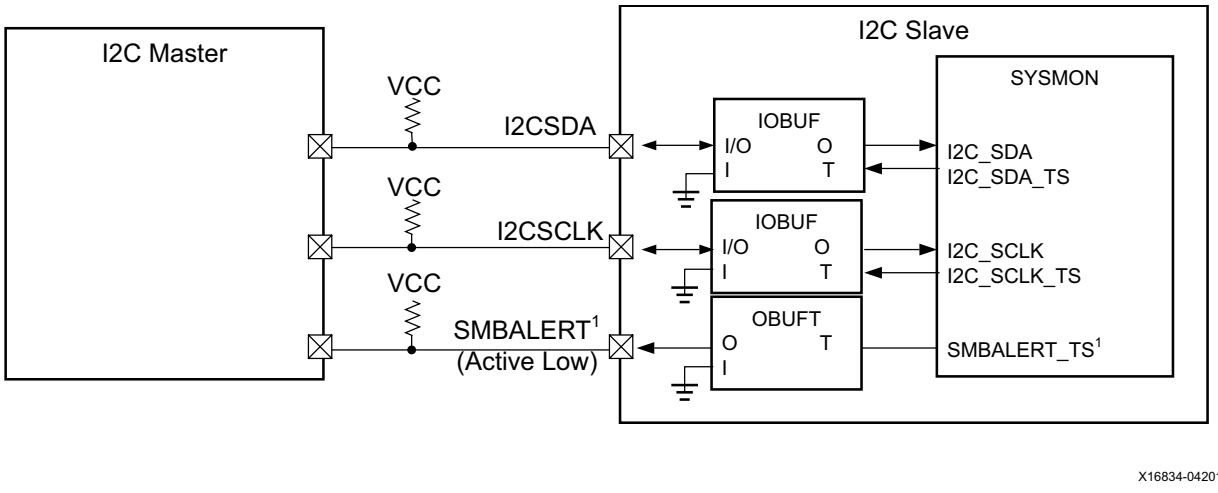


Figure 3-11: **SYSMON I2C DRP Interface**

Notes:
1. SMBALERT is an optional alert signal for PMBus (only available for SYSMONE4).

As shown in Figure 3-11, two package pins are required for the I2C serial clock (I2C_SCLK) and I2C serial data (I2C_SDA) lines. For Kintex UltraScale and Virtex UltraScale devices, SYSMONE1 uses dedicated connections for I2C_SCLK and I2C_SDA. The I2C_SCLK, I2C_SCLK_TS, I2C_SDA, and I2C_SDA_TS must be connected to bidirectional buffers in the design. These dedicated I2C connections are not accessible to the FPGA logic. For the SYSMONE1 block in UltraScale devices, assertion of the asynchronous reset either internally or through the assertion of PROGRAM_B_0 can cause SCL to assert Low for up to 500 ns.

For SYSMONE4, I2C_SCLK, I2C_SCLK_TS, I2C_SDA, I2C_SDA_TS, and SMBALERT_TS can either route to FPGA logic or use the dedicated routing.

IMPORTANT: *For SYSMONE4, because the I2C connections use GPIO, all I2C and PMBus transactions are reset after configuration is completed.*

For applications supporting the PMBus power system protocol specification, SYSMONE4 adds the SMBALERT output as described in the PMBus specification [Ref 11]. This optional pin provides an interrupt output and supports alert response address (ARA) functionality as defined by the PMBus specification.

IMPORTANT: *The SMBALERT continues to be asserted while the failing condition exists.*

For postconfiguration use of the DRP I2C interface, the System Management Wizard should be used for SYSMONE1 to ensure the dedicated I2C interface is properly connected. The I2C_SDA (bidirectional) and I2C_SCLK (bidirectional) ports on the SYSMONE4 should be connected at the top level. The System Management Wizard sets the I2C_EN bit of the control register `43h` High.

When not used for I2C, the dual-purpose package pins for I2C_SDA, I2C_SCLK, and SMBALERT (SYSMONE4 only) can be used as general purpose I/O. The I2C_EN bit of the control register `43h` must be set Low.

See the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575) [Ref 1] for pin location.

Power Management Bus Transfers (SYSMONE4)

In the SYSMONE4 block, external access includes support for the PMBus protocol. Figure 3-12 and Table 3-12 list the available command sequences and the command descriptions, respectively.

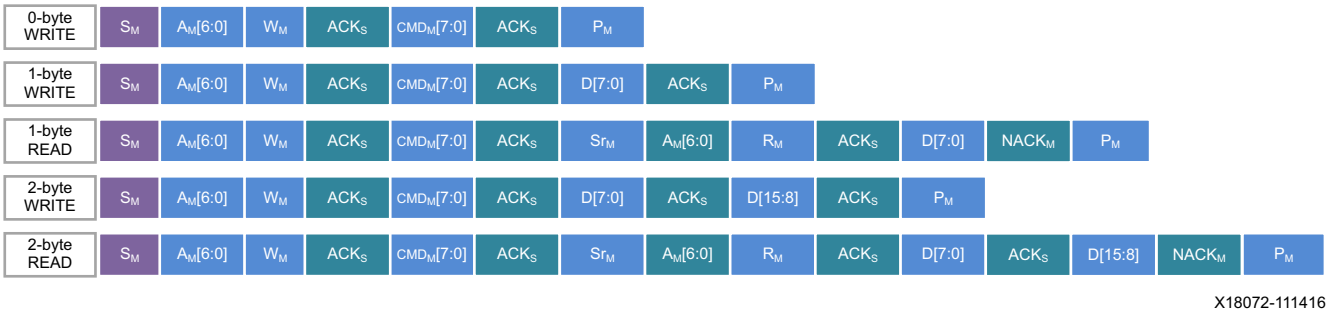


Figure 3-12: **Command Sequence**

Table 3-12: **Command Description**

Command	Description
S_M or Sr_M	Start or repeated start (there is no stop before repeated start) (master to slave)
$A_M[6:0]$	7-bit slave address (master to slave)
$CMD_M[7:0]$	8-bit PMBus command code (see Table 3-13)
ACK_S	0, acknowledgment (slave to master)
ACK_M	0, acknowledgment (master to slave)
$NACK_M$	1, not acknowledgment (master to slave)
D[7:0] or D[15:0]	Logical register/SYSMON DRP register address/SYSMON DRP register data (see Table 3-13)
P_M	Stop (master to slave)

PMBus Examples

Example 1: SYSMON Reg (40h) Write

1. Select SYSMON register address (D[7:0] = 40h, D[15:8] = 00h), and set command code as MFR_SPECIFIC_00 (CMD_M[7:0] = D0h).

Figure 3-13: **PMBus Example 1 Step 1 Sequence**

2. Write on 40h with proper SYSMON DRP setting DI[15:0] on D[15:0] with command code as MFR_SPECIFIC_01 (CMD_M[7:0] = D1h).

Figure 3-14: **PMBus Example 1 Step 2 Sequence**

Example 2: SYSMON Reg (41h) Read

1. Select SYSMON register address (D[7:0] = 41h, D[15:8] = 00h), and set command code as MFR_SPECIFIC_00 (CMD_M[7:0] = D0h).

Figure 3-15: **PMBus Example 2 Step 1 Sequence**

2. Read on SYSMON 41h data with DRP bus output DO[15:0] from D[15:0] with command code as MFR_SPECIFIC_01 (CMD_M[7:0] = D1h).

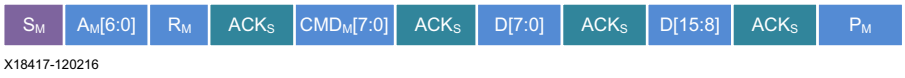


Figure 3-16: **PMBus Example 2 Step 2 Sequence**

SYSMONE4 supports the commands described in Table 3-13.

Note: Codes not listed in Table 3-13 are reserved.

Table 3-13: **PMBus Transfers (SYSMONE4) Commands**

Code	Command	Description	Transaction Type	Local Register Name/DRP Address	Data Bytes (Format)	Scope
00h	PAGE	Selects the supply for the single supply commands (Scope = PAGE). See Table 3-17.	Read Write	PMBUS_PAGE	1	Common
03h	CLEAR_FAULT	Clears all fault bits in all status registers simultaneously. At the same time, the device negates (clears, releases) its SMBALERT# signal output if the device is asserting the SMBALERT.	Write	ALL PMBUS STATUS REG	0	Common
19h	CAPABILITY	Allows host to identify key capabilities of PMBus device, i.e., PEC support, max bus speed, SMBALERT support.	Read	PMBUS_CAPABILITY	1	Common
20h	VOUT_MODE	To query the data format used by device for output voltage related data.	Read	PMBUS_MODE	1	Page
40h	VOUT_OV_FAULT_LIMIT	Sets the overvoltage value that causes an output overvoltage fault.	Read Write	Dynamic Upper threshold register for the supply addressed by PAGE setting	2 (LINEAR16)	Page
44h	VOUT_UV_FAULT_LIMIT	Sets the undervoltage value that causes an output undervoltage fault.	Read Write	Low threshold register for the supply addressed by PAGE setting	2 (LINEAR16)	Common
4Fh	OT_FAULT_LIMIT	Command sets the temperature of the unit at which it should indicate an over temperature fault OT.	Read Write	53h	2 (LINEAR11)	Common
51h	OT_WARNING_LIMIT	Command sets the temperature of the unit at which it should indicate an over temperature warning ALM_OV[0].	Read Write	50h	2 (LINEAR11)	Common
52h	UT_WARNING_LIMIT	Command sets the temperature of the unit at which it should indicate an under temperature warning ALM_UV[0].	Read Write	54h	2 (LINEAR11)	Common
53h	UT_FAULT_LIMIT	Command sets the temperature of the unit at which it should indicate an under temperature fault UT.	Read Write	57h	2 (LINEAR11)	Common
78h	STATUS_BYTE	Command returns one byte of information with a summary of the most critical faults.	Read	Low byte of PMBUS_STATUS_WORD [7:0]	1	Common

Table 3-13: **PMBus Transfers (SYSMONE4) Commands *(Cont'd)***

Code	Command	Description	Transaction Type	Local Register Name/DRP Address	Data Bytes (Format)	Scope
79h	STATUS_WORD	Command returns two bytes of information with a summary of the unit's fault condition.	Read	PMBUS_STATUS_WORD	2	Common
7Ah	STATUS_VOUT	Command returns one byte representing VOUT status.	Read Write	PMBUS_STATUS_VOUT	1	Page
7Dh	STATUS_TEMPERATURE	Command returns temperature status.	Read Write	PMBUS_STATUS_TEMP	1	Common
7Eh	STATUS_CML	Command returns communication, logic, and memory status.	Read Write	PMBUS_STATUS_CML	1	Common
8Bh	READ_VOUT	Command returns the actual, measured (not commanded) output voltage in the LINEAR16 format.	Read	Dynamic Voltage register for the supply addressed by PAGE setting	2 (LINEAR16)	Page
8Dh	READ_TEMPERATURE_1	Command returns temperature readings.	Read	00h	2 (LINEAR11)	Common
98h	PMBUS_REVISION	PMBUS_REVISION command stores or reads the revision of the PMBus to which the device is compliant.	Read	SYSMONE4_PMBUS_ REVISION	1	Common
99h	MFR_ID	The command is used to either set or read the Xilinx manufacturer's ID.	Block Read	mfr_id_ff	3	Common
9Ah	MFR_MODEL	The command is used to read the manufacturer's model number of the part.	Block Read	mfr_model_ff	2	Common
9Bh	MFR_REVISION	The MFR_REVISION command is used to either set or read the manufacturer's revision number.	Block Read	mfr_revision_ff	2	Common
D0h	MFR_SPECIFIC_00	(MFR_SELECT_REG) A manufacturer specific command to program config and sequence registers. The command is used to select a DRP register address.	Read Write	pmbus_page_r_ff	2	Common
D1h	MFR_SPECIFIC_01	(MFR_ACCESS_REG) Read or write data on the selected register.	Read Write	Dynamic Voltage register for the supply addressed by PAGE setting	2	Common
D2h	MFR_SPECIFIC_02	(MFR_READ_VOUT_MAX) Manufacturer specific command. Reads maximum recorded value for the selected supply.	Read	Dynamic Max register for the supply addressed by PAGE setting	2 (LINEAR16)	Page

Table 3-13: **PMBus Transfers (SYSMONE4) Commands *(Cont'd)***

Code	Command	Description	Transaction Type	Local Register Name/DRP Address	Data Bytes (Format)	Scope
D3h	MFR_SPECIFIC_03	(MFR_READ_VOUT_MIN) Manufacturer specific command. Reads minimum recorded value for the selected supply.	Read	Dynamic Min register for the supply addressed by PAGE setting	2 (LINEAR16)	Page
D5h	MFR_SPECIFIC_05	(MFR_ENABLE_VUSER_HRANGE) Reads the content of the vuser_en_hrange_pmbus[3:0] defined by the user via the memory cell setting.	Read	vuser_en_hrange_ff	1	Common
D6h	MFR_SPECIFIC_06	(MFR_READ_TEMP_MAX) Manufacturer specific command. Reads max recorded value of the local temperature channel.	Read	20h	2 (LINEAR11)	Common
D7h	MFR_SPECIFIC_07	(MFR_READ_TEMP_MIN) Manufacturer specific command. Reads min recorded value of the local temperature channel.	Read	24h	2 (LINEAR11)	Common

SYSMONE4 supports different data formats depending on the commands, LINEAR16 command (voltages using the PMBus format), LINEAR11 command (temperatures using the PMBus format), and one to three byte transfers. This section explains how the different data formats should be used for SYSMONE4. Additionally, example calculations for LINEAR11 and LINEAR16 are included in the TCL scripts in the example files. When reading temperature or supply voltages using the Vivado Hardware Manager, the LINEAR11 or LINEAR16 value is also shown.

LINEAR16 is based on a 16-bit unsigned value as described in Equation 3-2.

$$LINEAR16 = M \times 2^{-14} \qquad \textit{Equation 3-2}$$

For example, to set VOUT_OV_FAULT_LIMIT to 0.979V, `3EA8h` is written for code `41h`. From Table 3-14, high byte = `3E` and low byte = `A8h`. To set VOUT_UV_FAULT_LIMIT to 0.922V, `3B02h` is set to code `44h`.

Table 3-14: **LINEAR16 Data**

High Byte								Low Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M (16-bit, unsigned)															

IMPORTANT: *LINEAR16 values for 6V range sensors should not exceed 4V. Writing LINEAR16 values beyond these levels is not possible and if they were written earlier as higher values using I2C or DRP interfaces, they will be returned as 4V via PMBus reads.*

The LINEAR format settings can also be read using the VOUT_MODE command (code `20h`). The 8-bit data contains a 3-bit mode setting, `000b` for linear, and a 5-bit exponent setting as shown in Table 3-15.

Table 3-15: **VOUT_MODE Data Byte (Code 20h)**

Mode (linear)			Exponent (–14)				
7	6	5	4	3	2	1	0
0	0	0	1	0	0	1	0

For temperature values for PMBus commands, SYSMONE4 uses Equation 3-3.

$$LINEAR11 = M \times 2^{N} \qquad \textit{Equation 3-3}$$

For LINEAR11, M is an 11-bit, 2's complement value as shown in Table 3-16. N is a 5-bit, 2's complement exponential value. For example, N = `00h` and M = `50h` (`0050h`) is used to set the temperature to 80°C. N = `00h` and M = `7ECh` (`07ECh`) is used to set the temperature for -20°C. To set the temperature to 80.125°C, set N = `1Dh` and M = `281h` (`EA81h`).

Table 3-16: **LINEAR11 Data**

High Byte								Low Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N (5-bit, 2s complement)					M (11-bit, 2s complement)										

Because I2C and PMBus can also access the DRP register values, some of the data is stored in the original DRP register format. For example, MFR_SPECIFIC_04 (D4h) accesses DADDR = 70h. As such, the V_{CCINT} voltage follows the transfer curves in the power supply sensor. To highlight the difference, consider 0.979V (LINEAR16 = 3EA8h). The 16-bit value from the power supply sensor is:

ADC Code 16-bit = 0.979 x 65536 / 3

= 538Ah

Commands that are specific to a given supply are listed with the scope PAGE. See Table 3-17.

Table 3-17: **PAGE Address and Voltage Supply**

PAGE Address	Voltage Supply	ADC Channel
01h	V_{CCINT}	1
02h	V_{CCAUX}	2
06h	V_{CCBRAM}	6
0Dh	$V_{CC\_PSINTLP}$	13
0Eh	$V_{CC\_PSINTFP}$	14
0Fh	V_{CC_PSAUX}	15
20h	V_{USER0}	32
21h	V_{USER1}	33
22h	V_{USER2}	34
23h	V_{USER3}	35
FFh	All Supplies (CLEAR_FAULT)	All

Consequently, to read the V_{CCINT}, V_{CCAUX}, and V_{CCBRAM} voltages, the sequences shown in Table 3-18 are needed:

Table 3-18: **Read and Write Data for V_{CCINT}, V_{CCAUX}, and V_{CCBRAM}**

Code	Write Data	Read Data	Notes
00h	01		Set PAGE to V_{CCINT}
8Bh		LINEAR16	Read V_{CCINT} Voltage (LINEAR16)
00h	02		Set PAGE to V_{CCAUX}
8Bh		LINEAR16	Read V_{CCAUX} Voltage (LINEAR16)
00h	06		Set PAGE to V_{CCBRAM}
8Bh		LINEAR16	Read V_{CCBRAM} Voltage (LINEAR16)

Note: Additional details on the PMBus commands will be provided as information becomes available. Consult the PMBus specification [Ref 11] for general PMBus information.

I2C Read/Write Transfers

Access to the control and status registers is provided using I2C Write and Read transfers. I2C transfers data by the byte starting with the lowest byte first. Within the byte, the MSB is transferred first as shown in Figure 3-17.

I2C uses open-collector signaling, which allows bidirectional data on I2C_SDA. Figure 3-17 shows how I2C_SDA and I2C_SCLK are used to send a write to SYSMONE1 DRP. Because I2C_SDA is bidirectional, the master and slave devices control the I2C interface at different times during a transfer. Data is transmitted eight bits at a time with an acknowledge from the receiving device every eight bits. The transfer ends with the master device terminating the transfer with a stop command.

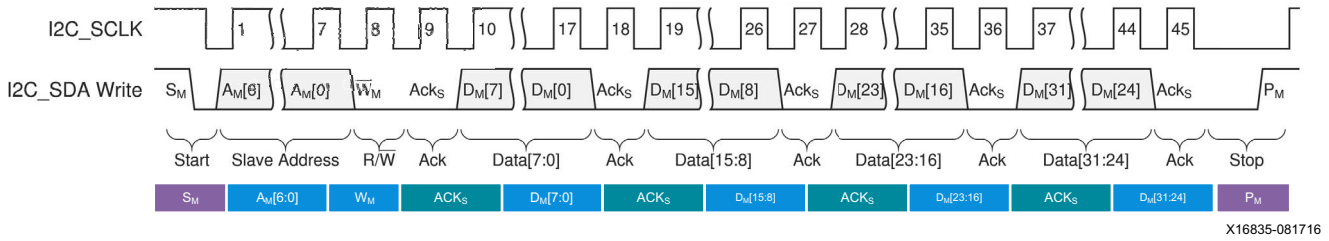


Figure 3-17: **SYSMONE1 I2C DRP Write**

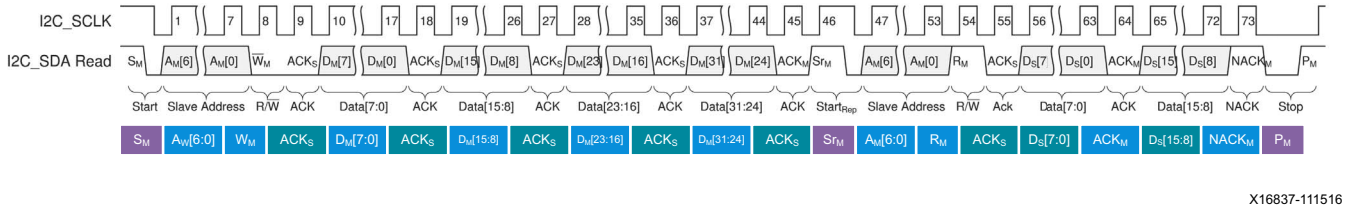


Figure 3-18: **SYSMONE1 I2C DRP Read**

Table 3-19: **SYSMONE1 and SYSMONE4 I2C DRP Label Descriptions**

Read Command	Description
A_M[6:0]	7-bit I2C slave address – master to slave
R/W_M	Read (1)/Write (0) command – master to slave
D_M[31:0]	32-bit DRP read command – master to slave
D_S[15:0]	16-bit DRP read data – slave to master (uses same commands as listed in Table 3-11)
ACK_M	Acknowledge – master to slave
ACK_S	Acknowledge – slave to master
$NACK_M$	Not acknowledge – master to slave
S_M	Start command – master to slave
Sr_M	Repeated start command – master to slave
P_M	Stop command – master to slave

Reading from status or control registers is performed with a combined format transfer, as shown in Figure 3-18. After writing the 32-bit DRP command, a repeated start command is sent followed by the read command. SYSMONE1 then takes over the transfer and sends the data back to the master. After the master acknowledges the transfer, the master terminates the transfer with a stop command.

I2C Slave Address Assignment

The I2C slave address can be set with these methods:

- During the power-up sequence using the voltages on the dedicated analog input channel V_P/V_N (preconfiguration and postconfiguration operation).
- By setting I2C_OR = 1 and writing the desired I2C slave address to I2C_A[6:0] in register `43h` (postconfiguration operation only).

To set the slave address at power-up, the initial voltage on the V_P/V_N input as set by a resistor divider off V_{CCAUX}, see Figure 3-19, are measured and recorded (in I2C Addr Meas (`38h`)) when INIT_B is released. Only this initial measurement is recorded. Consequently, V_{CCADC} should be stable by the time INIT_B is released, as well as V_{REF} if an external reference is being used. The four MSBs are decoded to create the I2C slave address as shown in Table 3-20. After the initial conversion, the V_P/V_N channel can be used for normal operation without affecting the I2C slave address.

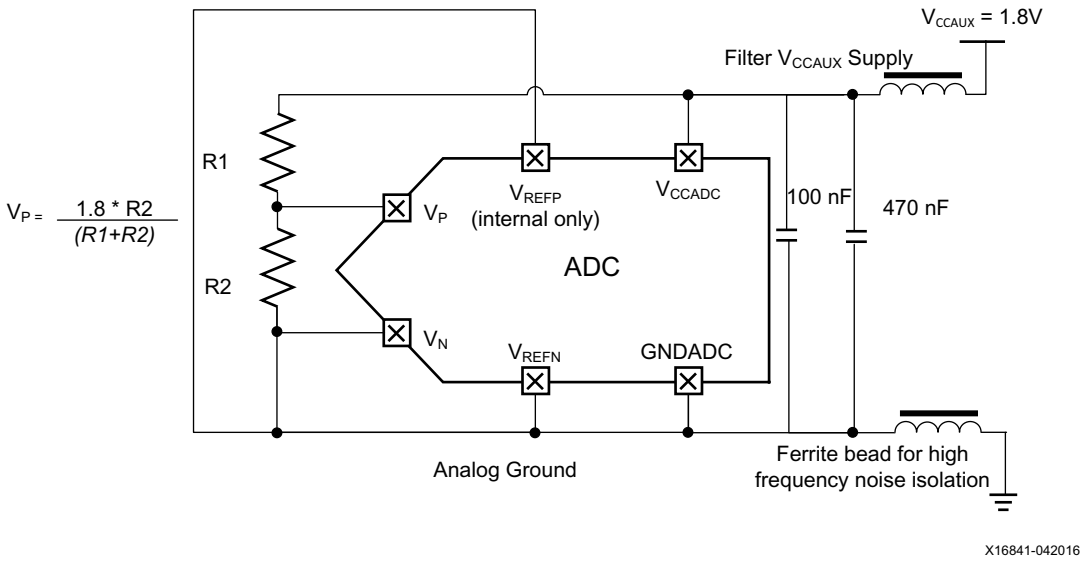


Figure 3-19: **SYSMON Pinout Recommendations for I2C Slave Address (Preconfiguration or Postconfiguration (I2C_OR = 0))**

In Table 3-20 resistor ratios for R1 and R2 are shown that divide down the 1.8V V_{CCAUX} to the required dedicated input voltages. Exact resistor values depend on system requirements. Ideally, the resistor values should be kept as low as possible to ensure that leakage current effects have minimal impact. The power consumption of the resistor divider network ultimately decides how low a resistance is tolerable. Because PMBus requires different functionality, the addresses for SYSMONE4 contain different addresses for I2C and PMBus. The third LSB within the 7-bit address must be High for PMBus addresses. This is also true when setting the addresses with the I2C_OR bit. Consequently, PMBus addresses are automatically xxx_x1xx, and I2C addresses are xxx_x0xx.

Table 3-20: **I2C Slave Address (I2C_OR = 0) Recommended Resistor Values**

Recommended R1, R2	I2C Addr Meas D[15:12] 38H	I2C Slave Address (I2C_OR = 0)	PMBus Address (I2C_OR = 0)
SYSMONE1 (X0Y0), SYSMONE4 (X0Y0)-SLR0			**SYSMONE4 (X0Y0)**
R2 = R1 * 5/283, or pulldown to ground	0	011_0010	011_0110
R2 = R1 * 15/273	1	000_1011	000_1111
R2 = R1 * 25/263	2	001_0011	001_0111
R2 = R1 * 35/253	3	001_1011	001_1111
R2 = R1 * 45/243	4	010_0011	010_0111
R2 = R1 * 55/233	5	010_1011	010_1111
R2 = R1 * 65/223	6	011_0011	011_0111
R2 = R1 * 75/213	7	011_1011	011_1111
R2 = R1 * 85/203	8	100_0011	100_0111

Table 3-20: **I2C Slave Address (I2C_OR = 0) Recommended Resistor Values *(Cont'd)***

Recommended R1, R2	I2C Addr Meas D[15:12] 38H	I2C Slave Address (I2C_OR = 0)	PMBus Address (I2C_OR = 0)
R2 = R1 * 95/193	9	100_1011	100_1111
R2 = R1 * 105/183	A	101_0011	101_0111
R2 = R1 * 115/173	B	101_1011	101_1111
R2 = R1 * 125/163	C	110_0011	110_0111
R2 = R1 * 135/153	D	110_1011	110_1111
R2 = R1 * 145/143	E	111_0011	111_0111
R2 = R1 * 155/133	F	011_1010	011_1110
SYSMONE1 (X1Y0), SYSMONE4 (X1Y0)-SLR2			**SYSMONE4 (X1Y0)**
R2 = R1 * 5/283, or pulldown to ground	0	100_0011	100_0111
R2 = R1 * 15/273	1	100_1011	100_1111
R2 = R1 * 25/263	2	101_0011	101_0111
R2 = R1 * 35/253	3	101_1011	101_1111
R2 = R1 * 45/243	4	110_0011	110_0111
R2 = R1 * 55/233	5	110_1011	110_1111
R2 = R1 * 65/223	6	111_0011	111_0111
R2 = R1 * 75/213	7	011_1010	011_1110
R2 = R1 * 85/203	8	011_0010	011_0110
R2 = R1 * 95/193	9	000_1011	000_1111
R2 = R1 * 105/183	A	001_0011	001_0111
R2 = R1 * 115/173	B	001_1011	001_1111
R2 = R1 * 125/163	C	010_0011	010_0111
R2 = R1 * 135/153	D	010_1011	010_1111
R2 = R1 * 145/143	E	011_0011	011_0111
R2 = R1 * 155/133	F	011_1011	011_1111
SYSMONE1 (X0Y1), SYSMONE4 (X0Y1)-SLR1			**SYSMONE4 (X0Y1)**
R2 = R1 * 5/283, or pulldown to ground	0	011_0000	011_0100
R2 = R1 * 15/273	1	000_1001	000_1101
R2 = R1 * 25/263	2	001_0001	001_0101
R2 = R1 * 35/253	3	001_1001	001_1101
R2 = R1 * 45/243	4	010_0001	010_0101
R2 = R1 * 55/233	5	010_1001	010_1101
R2 = R1 * 65/223	6	011_0001	011_0101
R2 = R1 * 75/213	7	011_1001	011_1101

Table 3-20: **I2C Slave Address (I2C_OR = 0) Recommended Resistor Values *(Cont'd)***

Recommended R1, R2	I2C Addr Meas D[15:12] 38H	I2C Slave Address (I2C_OR = 0)	PMBus Address (I2C_OR = 0)
R2 = R1 * 85/203	8	100_0001	100_0101
R2 = R1 * 95/193	9	100_1001	100_1101
R2 = R1 * 105/183	A	101_0001	101_0101
R2 = R1 * 115/173	B	101_1001	101_1101
R2 = R1 * 125/163	C	110_0001	110_0101
R2 = R1 * 135/153	D	110_1001	110_1101
R2 = R1 * 145/143	E	111_0001	111_0101
R2 = R1 * 155/133	F	011_1000	011_1100
SYSMONE1 (X1Y1), SYSMONE4 (X1Y1)-SLR3			**SYSMONE4 (X1Y1)**
R2 = R1 * 5/283, or pulldown to ground	0	100_0001	100_0101
R2 = R1 * 15/273	1	100_1001	100_1101
R2 = R1 * 25/263	2	101_0001	101_0101
R2 = R1 * 35/253	3	101_1001	101_1101
R2 = R1 * 45/243	4	110_0001	110_0101
R2 = R1 * 55/233	5	110_1001	110_1101
R2 = R1 * 65/223	6	111_0001	111_0101
R2 = R1 * 75/213	7	011_1000	011_1100
R2 = R1 * 85/203	8	011_0000	011_0100
R2 = R1 * 95/193	9	000_1001	000_1101
R2 = R1 * 105/183	A	001_0001	001_0101
R2 = R1 * 115/173	B	001_1001	001_1101
R2 = R1 * 125/163	C	010_0001	010_0101
R2 = R1 * 135/153	D	010_1001	010_1101
R2 = R1 * 145/143	E	011_0001	011_0101
R2 = R1 * 155/133	F	011_1001	011_1101

As an example, with some of the Xilinx evaluation kits, the resistor divider for V_P is effectively grounded by using only the R2 resistor connected to ground (R1 is left unpopulated). As a result, SYSMON measures 0V for the initial conversion causing D[15:12] = 0000 (I2C Addr Meas DADDR = 38h). From Table 3-20, the I2C slave address is 0110010 or 32h.

When using the I2C slave address at power-up (preconfiguration or postconfiguration with I2C_OR = 0), use any of the following options to ensure the correct I2C slave address decoding:

- Power V_{CCADC} off V_{CCAUX} and connect V_{REFP} to ADCGND to use the on-chip reference option.
- V_{CCADC}, V_{REF} (V_{REFP}/V_{REFN}) must be stable before V_{CCINT}, V_{CCAUX}, V_{CCO_0}, V_{CCBRAM}, V_{CCAUX_IO}, and V_{CCINT_IO} complete the power-on reset (see *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 7], *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893) [Ref 7], and *UltraScale Architecture Configuration User Guide* (UG570) [Ref 4] for additional details on power-on reset).
- Keep INIT_B asserted Low until V_{CCADC}, V_{REF}, and V_P/V_N have reached the expected DC levels.
- Initiate a new configuration sequence by asserting PROGRAM_B Low after V_{CCADC}, V_{REF}, and V_P/V_N have reached the expected DC levels.

To set the slave address with I2C_OR, set the I2C_OR = 1 and I2C_A to the desired I2C address in control register `43h`. The I2C_A values are used as the slave address. The override address and override enable can be set using DRP JTAG interface preconfiguration, in the configuration bitstream, or after configuration using the DRP port or JTAG.

SYSMON Operating Modes

SYSMON enables you to digitize a range of analog signals, such as an on-chip temperature sensor, on-chip supply sensors, the dedicated analog input (V_P/V_N), the auxiliary analog inputs, and the user supplies. SYSMON provides a number of operating modes to select the analog signals used for a design (see Table 3-8).

In default mode, SYSMON converts on-chip sensors. The default mode is available even when SYSMON is not instantiated in a design. The default mode uses calibration and on-chip oscillators to automatically measure temperature, V_{CCINT}, V_{CCAUX}, and V_{CCBRAM}.

The single channel mode uses a control register to select the analog channel. By writing to a control register, a design can select different analog channels. When used with the external multiplexer operation, the single channel mode can use a single analog input to read multiple analog signals.

In automatic channel sequencer mode, control registers configure the analog signals to be used. The sequencer then cycles through the selected analog signals updating the status registers as the conversions are completed.

Single Channel Mode

The single-channel mode is enabled when bits SEQ3 to SEQ0 in control register 41h are set to 0011 (see Table 3-8). In this mode, select the channel for analog-to-digital conversion by writing to bit locations CH5 to CH0 in control register 40h. Various configurations for single channel mode, such as analog input mode (BU) and settling time (ACQ), must also be set by writing to control register 40h. In applications where many channels need to be monitored, there can be a significant overhead for the microprocessor or other controller. To automate this task, a function called the automatic channel sequencer is provided.

Automatic Channel Sequencer

The automatic channel sequencer sets up a range of predefined operating modes, where a number of channels (on-chip sensors and external inputs) are used. The sequencer automatically selects the next channel for conversion, sets the averaging, configures the analog input channels, sets the required settling time for acquisition, and stores the results in the status registers based on a once off setting. The sequencer modes are set by writing to the SEQ3, SEQ2, SEQ1, and SEQ0 bits in configuration register 1 (see Table 3-8).

The channel sequencer functionality is implemented using thirteen control registers (46h - 4Fh and 7Ah - 7Ch (for SYSMONE4)). See Control Registers:

- ADC Channel Selection Registers (46h, 48h, and 49h)
- Slow Channel Selection Registers (7Ah, 7Bh, and 7Ch)
- ADC Channel Averaging (47h, 4Ah, and 4Bh)
- ADC Channel Analog-Input Mode (4Ch, 4Dh)
- ADC Channel Settling Time (4Eh, 4Fh)

ADC Channel Selection Registers (46h, 48h, and 49h)

The ADC channel selection registers enable and disable a channel in the automatic channel sequencer. The bits for these registers are defined in Table 4-1 and Table 4-2. The 16-bit registers are used to enable or disable the associated channels. A logic 1 enables a particular channel in the sequence. The sequence order is also listed.

Table 4-1: **Sequencer Registers (Channel Selection)**

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
X	X	X	X	X	X	X	X	X	X	X	X	CHSEL_USER3	CHSEL_USER2	CHSEL_USER1	CHSEL_USER0	SEQCHSEL0 (46h)
X	CHSEL_BRAM_AVG	CHSEL_VREFN	CHSEL_VREFP	CHSEL_V_pV_n	CHSEL_AUX_AVG	CHSEL_INT_AVG	CHSEL_TEMP	CHSEL_V_{CC}_PSAUX	CHSEL_V_{CC}_PSINTFP	CHSEL_V_{CC}_PSINTLP	X	X	X	X	CHSEL_SYSMON_CAL	SEQCHSEL1 (48h)
CHSEL_AUX15	CHSEL_AUX14	CHSEL_AUX13	CHSEL_AUX12	CHSEL_AUX11	CHSEL_AUX10	CHSEL_AUX9	CHSEL_AUX8	CHSEL_AUX7	CHSEL_AUX6	CHSEL_AUX5	CHSEL_AUX4	CHSEL_AUX3	CHSEL_AUX2	CHSEL_AUX1	CHSEL_AUX0	SEQCHSEL2 (49h)

Table 4-2: **Sequencer Register (Channel Selection) Bit Definitions**

Name	Sequence Number			ADC Channel CH[5:0]	Description
	SYSMONE1	SYSMONE4 (Kintex UltraScale+ Virtex UltraScale+)	SYSMONE4 (Zynq UltraScale+ MPSoC)		
CHSEL_SYSMON_CAL	1	1(1)	1(1)	001000 (8)	Enables system monitor calibration in the sequencer (High). Calibration enable only applies to SYSMONE1. For SYSMONE4, calibration is automatically in the slow sequence, which can be set to run at a lower rate.
CHSEL_$V_{CC_PSINTLP}$	N/A	N/A	1	001101 (13)	Enables on-chip $V_{CC\_PSINTLP}$ for sequencer (High).
CHSEL_$V_{CC_PSINTFP}$	N/A	N/A	2	001110 (14)	Enables on-chip $V_{CC\_PSINTFP}$ for sequencer (High).
CHSEL_V_{CC_PSAUX}	N/A	N/A	3	001111 (15)	Enables on-chip V_{CC_PSAUX} for sequencer (High).
CHSEL_TEMP	2	1	4	000000 (0)	Enables on-chip temperature for sequencer (High).
CHSEL_INT_AVG	3	2	5	000001 (1)	Enables on-chip V_{CCINT} for sequencer (High).
CHSEL_AUX_AVG	4	3	6	000010 (2)	Enables on-chip V_{CCAUX} for sequencer (High).
CHSEL_V_pV_n	5	4	7	000011 (3)	Enabled for sequencer (high) for V_P, V_N; dedicated analog inputs.

Table 4-2: **Sequencer Register (Channel Selection) Bit Definitions *(Cont'd)***

Name	Sequence Number			ADC Channel CH[5:0]	Description
	SYSMONE1	SYSMONE4 (Kintex UltraScale+ Virtex UltraScale+)	SYSMONE4 (Zynq UltraScale+ MPSoC)		
CHSEL_VREFP	6	5	8	000100 (4)	Enables V_{REFP} for sequencer (High).
CHSEL_VREFN	7	6	9	000101 (5)	Enables V_{REFN} for sequencer (High).
CHSEL_BRAM_AVG	8	7	10	000110 (6)	Enables on-chip V_{CCBRAM} for sequencer (High).
CHSEL_AUX15 to CHSEL_AUX0	24 to 9	23 to 8	26 to 11	011111 to 010000 (31 to 16)	Enables auxiliary channels for sequencer (High).
CHSEL_USER3 to CHSEL_USER0	28 to 25	27 to 24	30 to 27	100011 to 100000 (35 to 32)	Enables V_{USER} supplies for sequencer (High).

Notes:
1. Calibration is on by default in the slow channel sequencer. If enabled in the fast sequence, it will override slow.

Slow Channel Selection Registers (7Ah, 7Bh, and 7Ch)

The slow channel selection registers enable and disable a channel for the slow sequence (SYSMONE4 only). The bits for these registers are defined in Table 4-3 and Table 4-4. The 16-bit registers are used to enable or disable the associated channels. A logic 1 enables a particular channel in the sequence. See Continuous Sequence Mode (Slow Sequence - SYSMONE4) for additional details on how the slow sequence operates.

Table 4-3: **Sequencer Registers (Slow Channel Selection)**

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
X	SLOW_BRAM	SLOW_VREFN	SLOW_VREFP	SLOW_VpVn	SLOW_AUX_AVG	SLOW_INT_AVG	SLOW_TEMP	SLOW_VCC_PSAUX	SLOW_VCC_PSINTFP	SLOW_VCC_PSINTLP	X	X	X	X	SLOW_SYSMON	SLOWCHSEL (7Ah)
SLOW_AUX15	SLOW_AUX14	SLOW_AUX13	SLOW_AUX12	SLOW_AUX11	SLOW_AUX10	SLOW_AUX9	SLOW_AUX8	SLOW_AUX7	SLOW_AUX6	SLOW_AUX5	SLOW_AUX4	SLOW_AUX3	SLOW_AUX2	SLOW_AUX1	SLOW_AUX0	SLOWCHSEL (7Bh)
X	X	X	X	X	X	X	X	X	X	X	X	SLOW_USER3	SLOW_USER2	SLOW_USER1	SLOW_USER0	SLOWCHSEL2 (7Ch)

Table 4-4: **Sequencer Register (Slow Channel Selection) Bit Definitions**

Name	Description
SLOW_SYSMON	Enable for slow sequence for system monitor calibration (SYSMONE4 only)
$V_{CC\_PLINTLP}$	Enables on-chip $V_{CC\_PSINTLP}$ for sequencer (High)
$V_{CC\_PSINTFP}$	Enables on-chip $V_{CC\_PSINTFP}$ for sequencer (High)
V_{CC_PSAUX}	Enables on-chip V_{CC_PSAUX} for sequencer (High)
SLOW_TEMP	Enable for slow sequence for on-chip temperature
SLOW_INT_AVG	Enable for slow sequence for on-chip V_{CCINT}
SLOW_AUX_AVG	Enable for slow sequence for on-chip V_{CCAUX}
SLOW_V_pV_n	Enable for slow sequence for V_P, V_N – dedicated analog inputs
SLOW_VREFP	Enable for slow sequence for V_{REFP} (1.25V)
SLOW_VREFN	Enable for slow sequence for V_{REFN} (0V)
SLOW_BRAM	Enable for slow sequence for on-chip V_{CCBRAM}
SLOW_AUX15 to SLOW_AUX0	Enable for slow sequence for auxiliary channels
SLOW_USER3 to SLOW_USER0	Enable for slow sequence for V_{USER} supplies

ADC Channel Averaging (47h, 4Ah, and 4Bh)

The ADC channel averaging registers enable and disable the averaging of the channel data in a sequence. The result of a measurement on an averaged channel is generated by using 16, 64, or 256 samples. The amount of averaging is selected by using the AVG1 and AVG0 bits in configuration register 0 (see Control Registers). These registers also have the same bit assignments as the channel sequence registers listed in Table 4-5 and Table 4-6.

Table 4-5: **Sequencer Registers (Averaging)**

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
X	X	X	X	X	X	X	X	X	X	X	X	AVG_USER3	AVG_USER2	AVG_USER1	AVG_USER0	SEQAVG0 (47h)
X	AVG_BRAM_AVG	X	X	AVG_V_pV_n	AVG_AUX_AVG	AVG_INT_AVG	AVG_TEMP	AVG_V_{CC_PSAUX}	AVG_$V_{CC_PSINTFP}$	AVG_$V_{CC_PSINTLP}$	X	X	X	X	X	SEQAVG1 (4Ah)
AVG_AUX15	AVG_AUX14	AVG_AUX13	AVG_AUX12	AVG_AUX11	AVG_AUX10	AVG_AUX9	AVG_AUX8	AVG_AUX7	AVG_AUX6	AVG_AUX5	AVG_AUX4	AVG_AUX3	AVG_AUX2	AVG_AUX1	AVG_AUX0	SEQAVG2 (4Bh)

Table 4-6: **Sequencer Register (Averaging) Bit Definitions**

Name	Description
AVG_$V_{CC_PSINTLP}$	Enables average (High) for on-chip $V_{CC\_PSINTLP}$
AVG_$V_{CC_PSINTFP}$	Enables average (High) for on-chip $V_{CC\_PSINTFP}$
AVG_V_{CC_PSAUX}	Enables average (High) for on-chip V_{CC_PSAUX}
AVG_TEMP	Enables averaging (High) for on-chip temperature
AVG_INT_AVG	Enables averaging (High) for average on-chip V_{CCINT}
AVG_AUX_AVG	Enables averaging (High) for average on-chip V_{CCAUX}
AVG_V_pV_n	Enables averaging (High) for V_p, V_n; dedicated analog inputs
AVG_BRAM_AVG	Enables averaging (High) for average on-chip V_{CCBRAM}
AVG_AUX15 to AVG_AUX0	Enable averaging (High) for auxiliary channels
AVG_USER3 to AVG_USER0	Enable averaging (High) for V_{USER} supplies

Averaging can be selected independently for each channel in the sequence. When averaging is enabled for some of the channels of the sequence, the EOS is only pulsed after the sequence has completed the amount of averaging selected by using AVG1 and AVG0 bits (see Table 3-7). If a channel in the sequence does not have averaging enabled, its status register is updated for every pass through the sequencer. When a channel has averaging enabled, its status register is only updated after the averaging is complete. An example sequence is temperature and V_PV_N, where an averaging of 16 is enabled on V_PV_N. The sequence is calibration, temperature, calibration, temperature, V_PV_N temperature, V_PV_N, ..., temperature, and V_PV_N for each of the conversions where the temperature status register is updated. The V_PV_N status register is updated after the averaging of the 16 conversions.

For SYSMONE1, if averaging is enabled for the calibration channel by setting CAVG to a logic 0 (see Control Registers), the coefficients are updated after the first pass through the sequence. Subsequent updates to coefficient registers require 16 conversions before the coefficients are updated. Averaging is fixed at 16 samples for calibration. In the SYSMONE4, calibration is always enabled in the sequence and always averaged at 16 conversions.

ADC Channel Analog-Input Mode (4Ch, 4Dh)

These registers are used to configure an ADC channel as either unipolar or bipolar in the automatic sequence (see Analog Inputs). These registers also have the same bit assignments as the channel sequence registers listed in Table 4-7 and Table 4-8. However, only external analog input channels, such as the dedicated input channels (V_P and V_N) and the auxiliary analog inputs (VAUXP[15:0] and VAUXN[15:0]) can be configured in this manner. Setting a bit to logic 1 enables a bipolar input mode for the associated channel. Setting a bit to logic 0 (default) enables a unipolar input mode. All internal sensors use a unipolar transfer function.

Table 4-7: **Sequencer Registers (Analog Input Mode)**

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
X	X	X	X	INSEL_ VpVn	X	X	X	X	X	X	X	X	X	X	X	SEQINMODE0 (4Ch)
INSEL_ AUX15	INSEL_ AUX14	INSEL_ AUX13	INSEL_ AUX12	INSEL_ AUX11	INSEL_ AUX10	INSEL_ AUX9	INSEL_ AUX8	INSEL_ AUX7	INSEL_ AUX6	INSEL_ AUX5	INSEL_ AUX4	INSEL_ AUX3	INSEL_ AUX2	INSEL_ AUX1	INSEL_ AUX0	SEQINMODE1 (4Dh)

Table 4-8: **Sequencer Registers (Analog Input Mode) Bit Definitions**

Name	Description
INSEL_V_pV_n	Selects analog input-mode as unipolar (Low) or bipolar (High) input for V_p , V_n; dedicated analog inputs
INSEL_AUX15 to INSEL_AUX0	Selects analog input-mode as unipolar (Low) or bipolar (High) for auxiliary channels

ADC Channel Settling Time (4Eh, 4Fh)

The default settling time for an external channel in continuous sampling mode is four ADCCLK cycles. The settling time is additional acquisition time after the end of a conversion. However, by setting the corresponding bits (for external channels) to logic 1 in registers 4Eh and 4Fh, the associated channel can extend its settling time to 10 ADCCLK cycles. The bit definitions (the bits that correspond to specific external channels) for these registers are the same as the sequencer channel selection shown in Table 4-9 and Table 4-10.

Table 4-9: **Sequencer Registers (Acquisition)**

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	
X	X	X	X	ACQ_ V_pV_n	X	X	X	X	X	X	X	X	X	X	X	SEQACQ0 (4Eh)
ACQ_ AUX15	ACQ_ AUX14	ACQ_ AUX13	ACQ_ AUX12	ACQ_ AUX11	ACQ_ AUX10	ACQ_ AUX9	ACQ_ AUX8	ACQ_ AUX7	ACQ_ AUX6	ACQ_ AUX5	ACQ_ AUX4	ACQ_ AUX3	ACQ_ AUX2	ACQ_ AUX1	ACQ_ AUX0	SEQACQ1 (4Fh)

Table 4-10: **Sequencer Registers (Acquisition) Bit Definitions**

Name	Description
ACQ_V_pV_n	Selects acquisition time as 4 (Low) or 10 (High) ADCCLK cycles for V_P, V_N; dedicated analog inputs
ACQ_AUX15 to ACQ_AUX0	Selects acquisition time as 4 (Low) or 10 (High) ADCCLK cycles for auxiliary channels

Sequencer Modes

There are several sequencer modes, as defined by Table 3-8. These modes are described in this section.

Default Mode

The default mode is enabled by setting SEQ[3:0] = 0h. In this mode of operation, the SYSMON automatically monitors the on-chip sensors and stores the results in the status registers. The ADC is calibrated in this mode and an averaging of 16 samples is applied to all sensors. The SYSMON operates independently of any other control register settings in this mode. The SYSMON also operates in default mode after initial power up and during device configuration. Table 4-11 shows the default sequence for the SYSMON.

TIP: *All alarm outputs (ALM[15:0]) except OT are disabled in default mode. ADC calibration is automatically enabled in default mode.*

Table 4-11: **Default Mode Sequence**

Order			Channel	Address	Description
SYSMONE1	SYSMONE4 (Kintex UltraScale+, Virtex UltraScale+ FPGAs)	SYSMONE4 (Zynq UltraScale+ MPSoC)			
	Seq[0]	Seq[0]			
1	1	1	Calibration	08h	Calibration of the ADC
N/A	N/A	2	$V_{CC\_PSINTLP}$	0Dh	$V_{CC\_PSINTLP}$ supply sensor
N/A	N/A	3	$V_{CC\_PSINTFP}$	0Eh	$V_{CC\_PSINTFP}$ supply sensor
N/A	N/A	4	V_{CC_PSAUX}	0Fh	V_{CC_PSAUX} supply sensor
2	2	5	Temperature	00h	Temperature sensor
3	3	6	V_{CCINT}	01h	V_{CCINT} supply sensor
4	4	7	V_{CCAUX}	02h	V_{CCAUX} supply sensor
5	5	8	V_{CCBRAM}	06h	V_{CCBRAM} supply sensor

When a sample is enabled in both regular and slow sequence, only the regular sequence of that channel is enabled.

Single Pass Mode

The single pass mode is enabled by setting SEQ[3:0] = 1h. In single pass mode, the sequencer operates for one pass through the sequencer channel select registers (46h, 48h, and 49h) and then halts. A sequence of channels as selected in these registers is converted. When the sequence bits as shown in Table 3-8 are set to enable the automatic channel sequencer in single pass mode, the sequence starts. The settings in sequencer registers 46h–4fh are used to operate the sequence in a user-defined mode of operation. All channels listed in Table 4-1 and Table 4-2 are available to be used in a sequence. For an explanation of the sequencer registers, see Automatic Channel Sequencer. Another single pass can be started by writing to the sequence bits again. When the single pass is complete, the SYSMON defaults to Single Channel Mode described at the start of this chapter. Thus, the SYSMON converts the channel selected by bits CH5 to CH0 in configuration register 0.

Continuous Sequence Mode (Slow Sequence - SYSMONE4)

The continuous sequence mode is enabled by setting SEQ[3:0] = 2h. The continuous sequence mode is similar to single pass mode; however, the sequence automatically restarts as long as the mode is enabled. SYSMONE4 allows two sets of sequences to be defined with both sequences running concurrently. The ADC switches between the channels selected for the sequencer (ADCCHSEL 46h, 48h and 49h) and the channels selected for the slow sequence in the SLOWCHSEL control registers (7Ah, 7Bh, and 7Ch). Because the channel order for each of the sequences is dependent on ADCCHSEL and SLOWCHSEL, labels are used instead. SEQ(0) represents an entire fast sequence. SLOW(0) represents the first channel of the SLOW sequence. SLOW_SEQ allows lower priority channels in the slow sequence to be converted at a much lower rate. For example, because temperature changes slowly, the temperature channel can be enabled in the slow sequence, such that one channel of the enabled slow sequence channels is sampled every 64 iterations through the entire fast sequence (see Figure 4-1).

IMPORTANT: *Any of the analog channels can be enabled for the continuous sequence mode (ADCCHSEL 46h, 48h, or 49h) or for the slow sequence mode (SLOWCHSEL 7Ah, 7Bh, or 7Ch), but cannot be enabled for both. Channels that are enabled for both modes are ignored for the slow sequence. By default, both temperature and calibration are enabled in the slow sequence.*

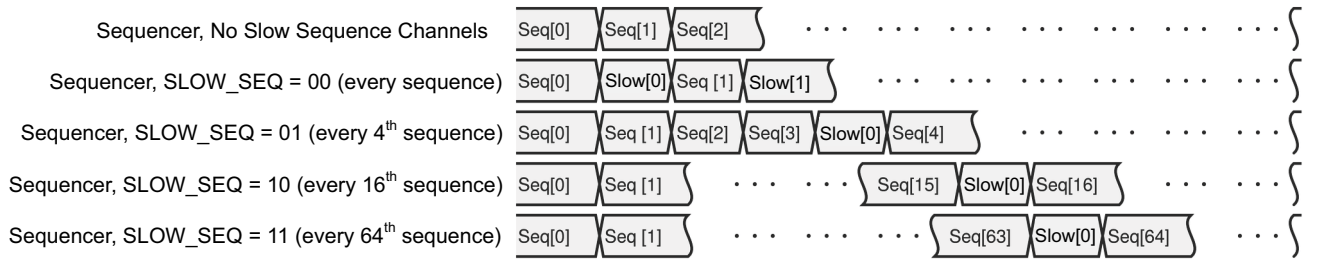


Figure 4-1: **Slow Sequence Comparison**

For SYSMONE4 designs using the slow sequence, two sequences now run concurrently. EOS is typically used to indicate when a sequencer has been completed. Because the sequence can have a different number of channels to be converted, each sequencer completes at different times. SLOW_EOS[1:0] determines if EOS should indicate the end of the sequence or the end of the slow sequence as shown in Figure 4-2.

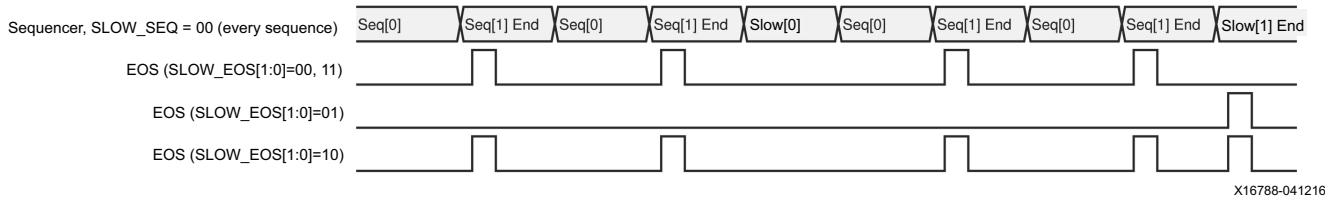


Figure 4-2: **EOS Operation for Slow Sequences (SYSMONE4 Only)**

The channel sequencer registers can also be reconfigured via the DRP at run time. The sequencer must first be disabled by writing to sequence bits SEQ3 to SEQ0 before writing to any of the sequencer channel registers.

IMPORTANT: *The SYSMON must be placed in default mode by writing zeros to SEQ0, SEQ1, SEQ2, and SEQ3 while updating these registers.*

The SYSMON is automatically reset whenever SEQ3 to SEQ0 are written to. The current status register contents are not reset at this time. Restarting the sequencer by writing to bits SEQ3 to SEQ0 resets all channel averaging.

Single Channel Mode (Sequencer Off)

The single channel mode is enabled by setting SEQ[3:0] = `3h`. See Single Channel Mode for more information.

External Multiplexer Mode

The SYSMON supports the use of an external analog multiplexer to implement several external analog inputs in situations where I/O resources are limited and auxiliary analog inputs are not available.

The SYSMON track/hold amplifiers return to track mode as soon as a conversion starts. Therefore, the acquisition on the next channel can start during the current conversion cycle. An output bus called MUXADDR[4:0] allows the SYSMON to control an external multiplexer. The address on this bus reflects the channel currently being acquired, and it changes state as soon as the SYSMON enters acquisition mode. The external multiplexer can be connected to the dedicated analog input or one of the auxiliary analog inputs.

External Multiplexer Operation

Figure 4-3 illustrates the external multiplexer concept. In this example, an external 16:1 analog multiplexer is used instead of consuming the 32 I/Os required to implement the 16 auxiliary analog input channels using the internal multiplexer. Any four general purpose I/Os can be used for the external multiplexer decode operation. As shown in Figure 4-3, the dedicated analog inputs (V_P/V_N) are used to connect the external multiplexer to the SYSMON block, thereby making 16 analog inputs available. The external multiplexer mode of operation is enabled by setting the MUX bit in configuration register 0 (see Control Registers).

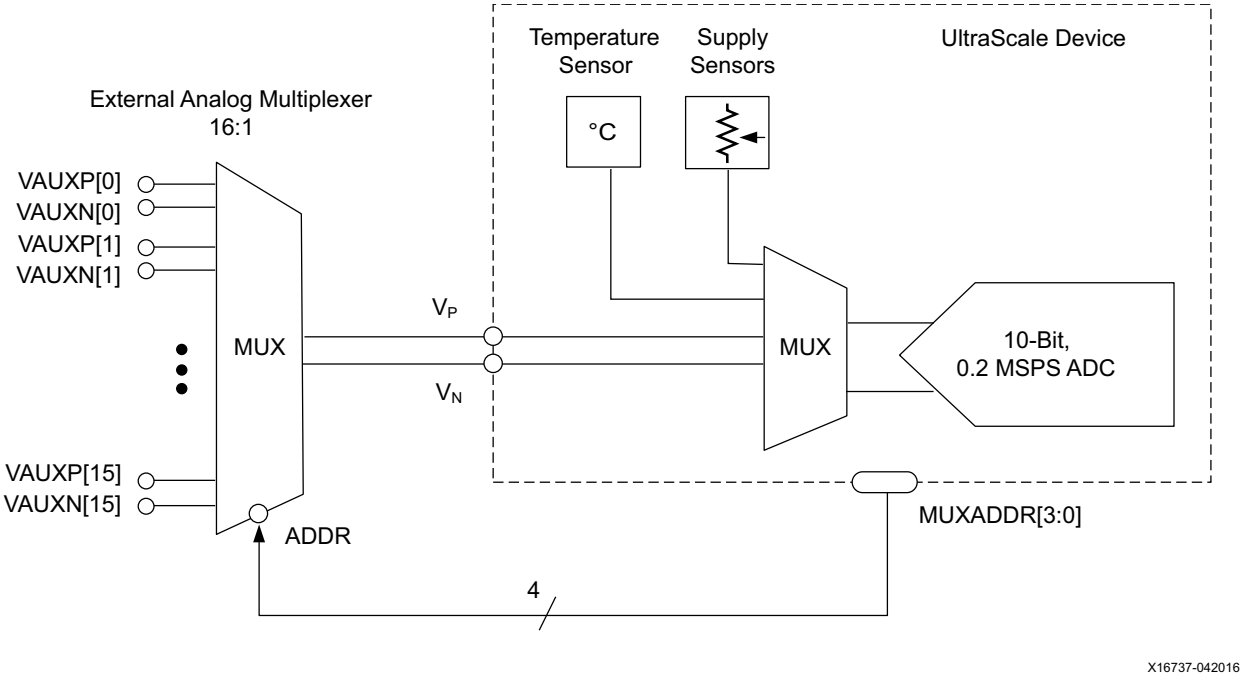


Figure 4-3: **External Multiplexer Mode**

When the MUX bit is set to `1`, the channel selection bits (CH5 to CH0) in configuration register 0 are used to nominate the channel for connection to the external multiplexer. For example, as shown in Figure 4-3, the dedicated analog input channel V_P/V_N is used. In this case, channel 3 (`00011b`) should be written to CH5 to CH0 in control register `40h`. Any one of the auxiliary channels can also be used for connection to the external multiplexer.

When using the external multiplexer mode, the status registers for the auxiliary analog inputs are used. For example, status registers `10-1Fh` store the measurement results. Similarly, the automatic channel sequencer should be used to select the desired external analog multiplexer addresses.

Automatic Alarms

The SYSMON also generates an alarm signal on the logic outputs ALM[15:0] when an internal sensor measurement exceeds some user-defined thresholds. Only the values written to the status registers are used to generate alarms. If averaging has been enabled for a sensor channel, the averaged value is compared to the alarm threshold register contents. The alarm outputs are disabled by writing a `1` to bits ALM15 to ALM0 in configuration register 1. The alarm thresholds are stored in control registers `50h` to `6Dh`. Table 4-12 defines the alarm thresholds that are associated with specific control registers. The limits written to the threshold registers are MSB justified. Limits are derived from the temperature and power-supply sensor transfer functions (see Figure 2-11 and Figure 2-12).

Table 4-12: **Alarm Threshold Registers**

Control Register	Description	Alarm
50h	Temperature upper	ALM[0]
51h	V_{CCINT} upper	ALM[1]
52h	V_{CCAUX} upper	ALM[2]
53h	OT upper(1)	OT
54h	Temperature lower	ALM[0]
55h	V_{CCINT} lower	ALM[1]
56h	V_{CCAUX} lower	ALM[2]
57h	OT lower(1)	OT
58h	V_{CCBRAM} upper	ALM[3]
59h	$V_{CC\_PSINTLP}$ upper(2)	ALM[4]
5Ah	$V_{CC\_PSINTFP}$ upper(2)	ALM[5]
5Bh	V_{CC_PSAUX} upper(2)	ALM[6]
5Ch	V_{CCBRAM} lower	ALM[3]
5Dh	$V_{CC\_PSINTLP}$ lower(2)	ALM[4]
5Eh	$V_{CC\_PSINTFP}$ lower(2)	ALM[5]
5Fh	V_{CC_PSAUX} lower(2)	ALM[6]
60h	V_{USER0} upper	ALM[8]
61h	V_{USER1} upper	ALM[9]
62h	V_{USER2} upper	ALM[10]
63h	V_{USER3} upper	ALM[11]
68h	V_{USER0} lower	ALM[8]
69h	V_{USER1} lower	ALM[9]
6Ah	V_{USER2} lower	ALM[10]

Table 4-12: **Alarm Threshold Registers *(Cont'd)***

Control Register	Description	Alarm
6Bh	V_{USER3} lower	ALM[11]

1. OT upper and OT lower are described in Thermal Management.
2. Zynq UltraScale+ MPSoC.

Supply Sensor Alarms

Supply sensor alarms are enabled when the given supply's value falls outside of a window which has a range determined by the upper and lower thresholds. For example, when the measured value on the supply sensor (for example, 01h for V_{CCINT}) is greater than the upper thresholds (51h for V_{CCINT} upper) or less than the lower thresholds (55h for V_{CCINT} lower), the corresponding alarm will be High. The alarms are reset when a subsequently measured value falls inside the threshold. Supply sensor alarms for V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, $V_{CC\_PSINTLP}$, $V_{CC\_PSINTFP}$, V_{CC_PSAUX}, and V_{USER}[3:0] behave in this manner. For Temperature alarms, see Thermal Management.

Four additional user selectable alarms are available in SYSMONE1 USER3 to USER0 and can be connected to different voltage supplies using the System Management Wizard.

Thermal Management

Over Temperature Automatic Shutdown

The on-chip temperature measurement is used for critical temperature warnings and also supports automatic shutdown to help prevent the device from being permanently damaged.

IMPORTANT: *Automatic shutdown is off by default. Automatic shutdown must be enabled by setting control register 53 as described in this section. As an additional safeguard, the OT bit in control register 41 can be used to further disable automatic shutdown regardless of the setting of control register 53. Additionally, an XDC command must be used to enable the automatic shutdown. Read this entire section when using automatic shutdown.*

The on-chip temperature measurements record the junction temperatures continuously during preconfiguration and automatic shutdown. For configured devices, the on-chip temperature measurements are on by default (see Default Mode). To use automatic shutdown, this constraint must be added to the project XDC file:

```
set_property BITSTREAM.CONFIG.OVERTEMPSHUTDOWN ENABLE [current_design]
```

The default over temperature (OT) threshold is 125°C. The 125°C threshold is used when the contents of the OT upper alarm register (53h) is 000h, including preconfiguration. To override this default condition, the four LSBs must be set to 3h and the 12 MSBs of the OT upper register (control register 53h) must be set using the temperature sensor transfer functions (see Equation 2-5, Equation 2-7, Equation 2-9, or Equation 2-11). Equation 4-1 and Equation 4-2 are examples for the SYSMONE1 block with an external reference (see Equation 2-7 for SYSMONE1 with internal reference and Equation 2-9 and Equation 2-11 for the SYSMONE4 block).

$$\textit{12-bit OT alarm limit with SYSMONE1 using external reference} = (\textit{Temp} + 273.8195) \times \left(\frac{4096}{502.9098}\right) \qquad \textit{Equation 4-1}$$

$$\textit{OT temp with SYSMONE1 using external reference} = \frac{(\textit{12 bit ADC Code}) \times 502.9098}{4096} - 273.8195 \qquad \textit{Equation 4-2}$$

Consequently, for 125°C, control register 53h must be set to CB0h for the 12 MSBs when using the external reference. Because the four LSBs must be set to 3h, this gives the 16-bit register value of CB03h for control 53h.

$$\textit{Temp upper/lower (C)} = \frac{(\textit{16 bit ADC Code}) \times 502.9098}{65536} - 273.8195 \qquad \textit{Equation 4-3}$$

For the remaining temperature thresholds, use Equation 2-5, Equation 2-7, and Equation 2-9 to define the 16-bit ADC code values.

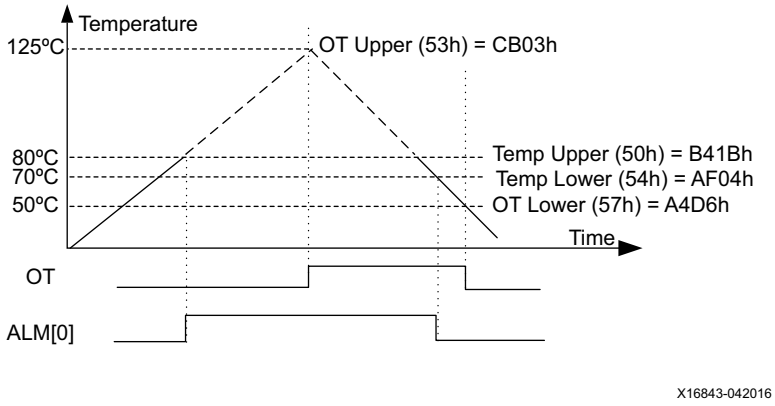


Figure 4-4: **Thermal Management Operation Example with External Reference**

As shown in Figure 4-4, when the die temperature exceeds the OT upper threshold (or the default of 125°C), the OT alarm logic output becomes active and 10 ms later the device initiates a shutdown sequence. When the automatic shutdown starts, the device is disabled and GHIGH is asserted to prevent any contention (see the *UltraScale Architecture Configuration User Guide* (UG570) [Ref 4]. When OT is deasserted (50°C as shown in Figure 4-4), GHIGH is also deasserted and the start-up sequence is initiated releasing all global resources.

While the device is shutdown, SYSMON automatically uses the internal clock oscillator, but otherwise remains unchanged. SYSMON temperature data can be accessed using the JTAG interface. JTAG is only guaranteed to 125°C. During shutdown, I2C is not available.

When the OT alarm has been triggered in UltraScale devices, asserting PROGRAM_B low can potentially start a reconfiguration even though the die temperature has not reset the OT alarm (OT lower 57h). In 7 series devices, PROGRAM_B was ignored until the OT alarm was reset.

The automatic shutdown feature is intended to prevent permanent damage to the device. After the temperature has gone below the OT lower (57h) setting and OT is deasserted, the device should be reconfigured to ensure the device is reset to a known safe state. Additionally, because a catastrophic failure occurred, all power to the device should be removed and it should be determined why the device temperature increased so dramatically. Designs should use a thermal management procedure with the temperature alarm (ALM[0]) to actively control the devices temperature during operation.

In SYSMONE4 for UltraScale+ devices only, there are two types of alarm criteria available for temperature readings. The hysteresis mode as described above and the window mode as seen in the power supply alarms. When bit 0 of 0x54 or 0x57 is set High, the temperature alarm behaves in window mode. The respective alarm is enabled when the temperature range falls outside the target temperature window as specified in bits 15:1 in these registers. This results in temperature alarms asserting when outside of the upper and lower range limits and deasserting when temperature readings fall inside a range set by the upper and lower thresholds. When bit 0 of 0x54 or x57 is set Low, the respective temperature alarm operates in hysteresis mode with the alarm asserting when the upper limit is exceeded and deasserting when the temperature is reduced below the lower threshold limit. Temperature and OT are independent and can operate in the same or different mode depending on bit 0 of the respective lower temperature register.

IMPORTANT: *In SYSMONE4 only, bits 15:1 of registers 0x54 or 0x57 should be used to define alarm temperatures. The LSB dictates the threshold behavior of the respective alarm.*

The default OT threshold is only overridden to a custom OT upper alarm threshold value when the OT upper alarm threshold register (53h) ends with 0011b (DI[3:0]). The automatic shutdown feature can be disabled by setting the OT signal within Config Reg 1 (41h) High or by adding this constraint to the project's XDC file:

```
set_property BITSTREAM.CONFIG.OVERTEMPSHUTDOWN DISABLE [current_design]
```

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 3] for additional details on device configuration bitstream settings. ENABLE and DISABLE are allowable values.

User-Programmable Temperature Alarms

A second user-programmable temperature threshold level (temperature upper, 50h) can be used to carry out a user-defined thermal management procedure, such as powering on or controlling the speed of a fan. Alarm signal ALM[0] is High when the device temperature exceeds the limit in the temperature upper control register 50h. ALM[0] remains High until the temperature falls below the lower threshold, temperature lower (54h). As shown in Figure 4-4, this means that ALM[0] is High when the temperature reaches 80°C and remains High until the temperature falls to 70°C. This operation differs for the supply sensor alarm because the supply alarm resets when the measurement is between the upper and lower thresholds.

Application Guidelines

The SYSMON is a precision analog measurement system based on a 10-bit analog-to-digital converter (ADC) with an LSB size approximately equal to 1mV. To achieve the best possible performance and accuracy with all measurements (both on-chip and external), several dedicated pins for the ADC reference and power supply are provided. When connecting these pins, follow the guidelines in this chapter to ensure the best possible performance from the ADC. This chapter outlines the basic design guidelines to consider as part of the requirements for board design.

Reference Inputs (VREFP and VREFN)

These high-impedance inputs are used to deliver a differential reference voltage for the analog-to-digital conversion process. The ADC is only as accurate as the reference provided. Any reference-voltage error results in a gain error versus the ideal ADC transfer function (see Chapter 2, Basic Functionality). Errors in the reference voltage affect the accuracy of absolute measurements for both on-chip sensors and external channels. Noise on the reference voltage also adds noise to the ADC conversion and results in more code transition noise or poorer than expected SNR.

For typical usage, the reference voltage between V_{REFP} and V_{REFN} should be maintained at 1.25V ± 0.2% using an external reference IC. Reference voltage ICs that deliver 1.25V are widely available from several vendors. Many vendors offer reference voltage ICs in small packages (SOT-23 and SC70).

RECOMMENDED: *The 1.25V reference should be placed as close as possible to the reference pins and connected directly to the V_{REFP} input, using the decoupling capacitors recommended in the reference IC data sheet. The recommended reference connections are illustrated in Figure 5-1.*

The SYSMON also has an on-chip reference option that is selected by connecting V_{REFP} and V_{REFN} to ADCGND as shown in Figure 5-1. Due to reduced accuracy, the on-chip reference does impact the measurement performance of the SYSMON. The performance with on-chip reference is specified in the UltraScale device data sheets. The accuracy of the reference has a direct impact on the accuracy of the samples taken by the ADC. For example, with an external reference maintaining a ±0.2% tolerance, there can be a ±2 LSB (at 10 bit) accuracy impact on the ADC samples. With an external reference maintaining a ±1% tolerance, there can be a ±10 LSB (at 10 bit) accuracy impact. See the UltraScale device data sheets for the applicable external reference specification.

Analog Power Supply and Ground

The analog power supply (V_{CCADC} and V_{CC_PSADC}) and ground (GNDADC and GND_PSADC) inputs provide the power supply and ground reference for the analog circuitry in the SYSMON. A common mechanism for the coupling of noise into an analog circuit is from the power supply and ground connections. Excessive noise on the analog supply or ground reference affects the ADC measurement accuracy. For example, I/O switching activity can cause significant disturbance of the digital ground reference plane. Thus, it is not advisable to use the digital ground as an analog ground reference for SYSMON.

Similarly, for the digital supplies for the interconnect logic, high switching rates easily result in high-frequency voltage variations on the supply, even with decoupling. In an effort to mitigate these effects on the ADC performance, a dedicated supply and ground reference is provided. Figure 5-1 illustrates how to use the 1.8V V_{CCAUX} supply to power the analog circuitry. V_{CCAUX} is filtered using a low-pass network. The filter design depends on the ripple and ripple frequency (if any) on the V_{CCAUX} supply if, for example, a switching regulator is used. There is also a power-supply rejection specification for the external reference circuit to consider. The filtering should ensure no more than 1 LSB (1mV) of noise on the reference output to minimize any impact on ADC accuracy at 10 bits. Depending on the ripple frequency of the supply, a 10-20 uH inductor might be better than a ferrite bead. If the low-pass network filtering of V_{CCAUX} contains more than 1 LSB of noise, an additional regulator might be required (for example, ADP123). See *XADC Layout Guidelines* (XAPP554) [Ref 2] for additional details.

IMPORTANT: *V_{CC_PSADC} and GND_PSADC might require additional filtering if there is excessive noise on V_{CCAUX}. Make sure all supplies meet the data sheet requirements. V_{CC_PSADC} and V_{CCADC} must meet the 1.8V ±3% requirements at the package pins.*

The other source of noise coupling into the ADC is from the ground reference GNDADC. In mixed-signal designs, it is common practice to use a separate analog ground plane for analog circuits to isolate the analog and digital ground return paths to the supply. Common ground impedance is a mechanism for noise coupling and needs to be carefully considered when designing the PCB. This is shown in Figure 2-3, where the common ground impedance RG converts digital switching currents into a noise voltage for the analog circuitry. While a

separate analog ground plan is recommended for 10-bit operation, it is often not possible or practical to implement a separate analog ground plane in a design. For example, if only the on-chip sensors are used, one low-cost solution is to isolate V_{REFN} and GNDADC ground references (such as a trace) from the digital ground (plane) using a ferrite bead as shown in Figure 5-1.

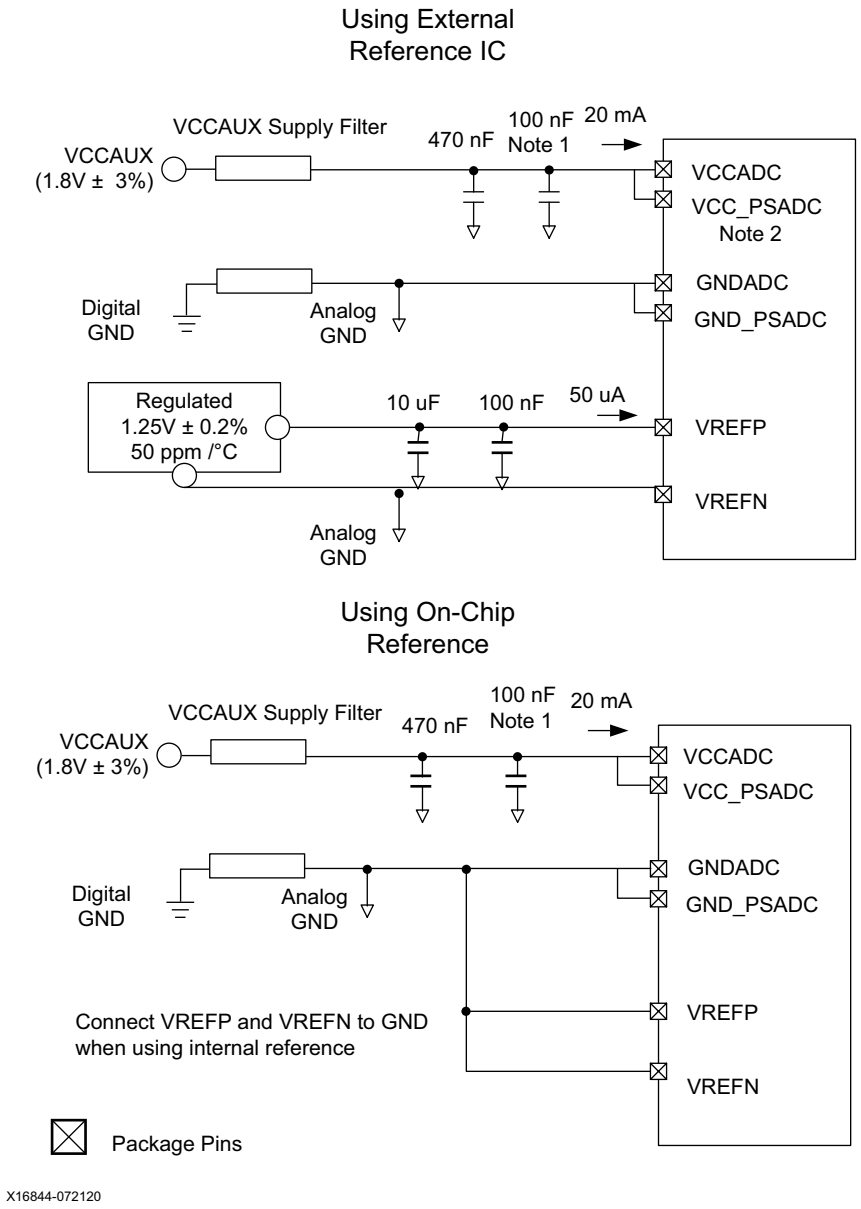


Figure 5-1: **ADC Power and Ground Connections**

Notes relevant to Figure 5-1:

1. Place the 100 nF capacitor as close as possible to the package balls.

   The ferrite bead behaves like a resistor at high frequencies and functions as a lossy inductor. A typical ferrite impedance versus frequency plot is shown in Figure 5-2. The ferrite helps provide high frequency isolation between digital and analog grounds. The reference IC maintains a 1.25V difference of between V_{REFP} and V_{REFN}. The ferrite offers little resistance to the analog DC return current.

   The reference inputs should be routed as a tightly coupled differential pair from the reference IC to the package pins. If routed on the same signal layer, the supply and analog ground traces (V_{CCADC} and GNDADC) should be used to shield the reference inputs because they have a higher tolerance to any coupled noise.

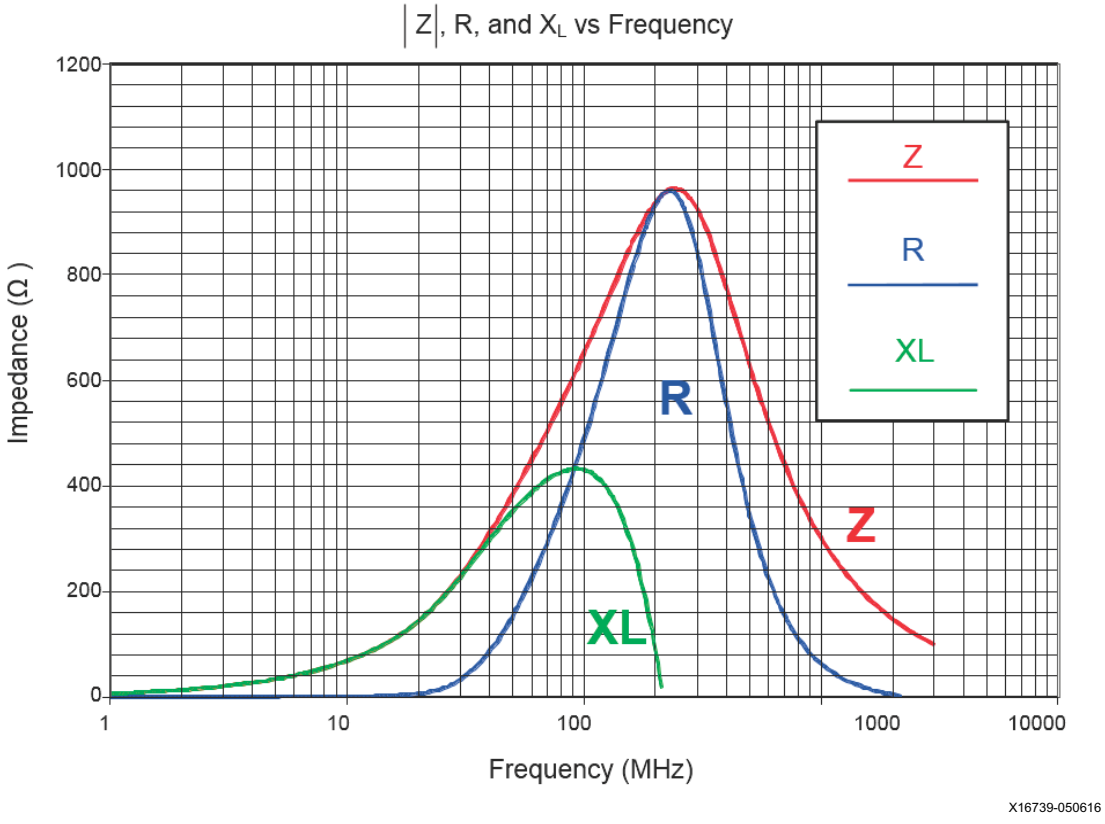


Figure 5-2: **Ferrite Bead Characteristics**

2. When the PS and PL supplies are both powered for the Zynq UltraScale+ MPSoC.

External Analog Inputs

The analog inputs are high-impedance differential inputs. The differential input scheme enables the rejection on common mode noise on any externally applied analog-input signal. Because of the high impedance of each input (such as V_P and V_N), the input AC impedance is typically determined by the sensor, the output impedance of the driving circuitry, or other external components. Figure 5-3 illustrates a simple resistor divider network is used to monitor an external 2.5V supply rail in unipolar input mode. To ensure that noise coupled onto the analog inputs is common to both inputs (reduce differential noise), the impedance on each input should be matched. Analog-input traces on the PCB should also be routed as tightly coupled differential pairs.

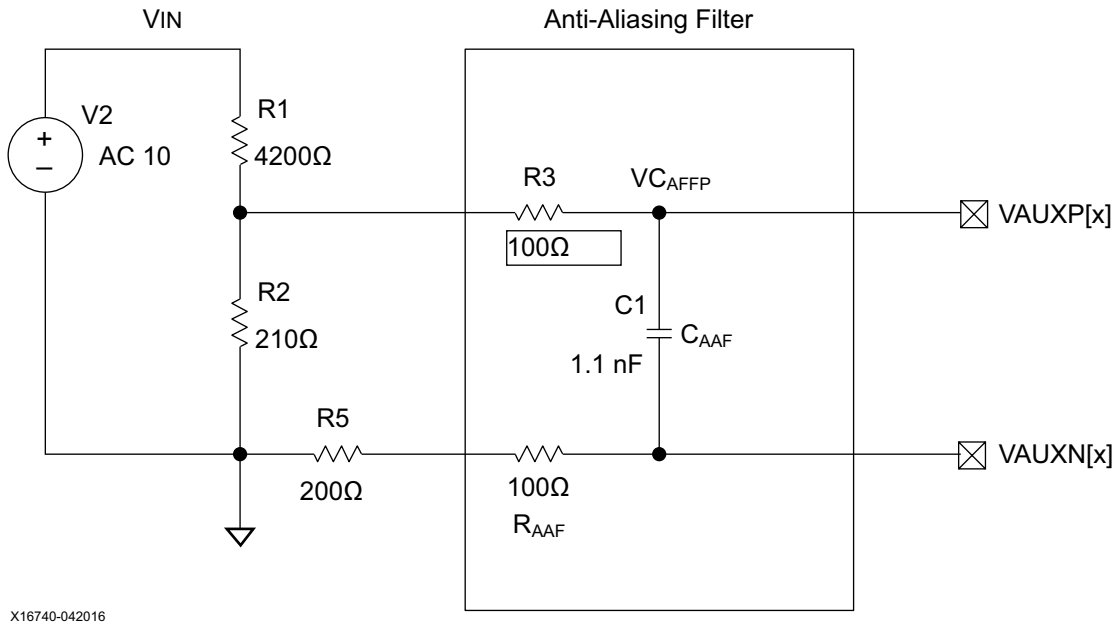


Figure 5-3: **Voltage Attenuation**

Anti-Alias Filters

Also shown in Figure 5-3, is a low-pass filter network at the analog differential inputs. This filter network is commonly referred to as the anti-alias filter and should be placed as close as possible to the package pins. The sensor can be placed remotely from the package as long as the differential input traces are closely coupled. The anti-alias filter attenuates high-frequency signal components entering the ADC where they could be sampled and aliased, resulting in ADC measurement corruption.

As shown in Figure 5-3, resistors R1 and R2 divide the 10V supply down to 0.5V to work with the SYSMON. R5 has been impedance matched to the parallel resistance of R1 and R2.

The anti-aliasing filter's settling time for this example is determined by Equation 5-1. With a resolution of 10 bits, the example's components result in a settling time of 5.0 x 10^{-6} s or 200 Ks/s.

$$Tsettling = ln^{(2^{Resolution+1})\times\left(\frac{R1\times R2}{R1+R2}+R3+R4+R5\right)\times C1} = 5.0\times10^{-6}\times[s] \quad \textit{Equation 5-1}$$

See *Driving the Xilinx Analog-to-Digital Converter* (XAPP795) [Ref 5] for more details. A discussion of aliasing in sampled systems is beyond the scope of this document. A reference book on data converters can provide more information on this topic.

Over and Under Voltages

The input voltage can exceed V_{CCADC} (1.8V) or go below GNDADC by as much as 100 mV without damage to the SYSMON. A current-limiting resistor of at least 100Ω should be placed in series with the analog inputs to limit the current to 1 mA. The resistors in the anti-alias filters fulfill this requirement. If the analog input range (1V) is exceeded, the ADC output code clips at the maximum output code shown in Figure 2-1 or Figure 2-2, depending on the analog input mode. Negative input voltages clip at zero code.

SYSMON Software Support

Example Design Instantiation

The rdf0304-ultrascale-sysmon.zip design file can be downloaded here:

https://www.xilinx.com/support/documentation/user_guides/rdf0304-ultrascale-sysmon.zip

This HDL example sets up the SYSMON to monitor all the on-chip sensors, i.e., temperature, V_{CCINT}, V_{CCAUX}, and V_{CCBRAM}. See Temperature Sensor and Power and User Supply Sensors. In addition, four auxiliary analog input channels are also monitored. The SYSMON is also set to automatically generate alarm outputs when the defined operating ranges for the device supply voltages and temperature are exceeded (see Automatic Alarms). The SYSMON is operated in continuous sequence mode for this example (see Sequencer Modes). For clarity (and shorter simulations), the averaging function is disabled by the design. The disabling of the averaging function can be seen during the DRP write. Averaging does not have any impact on the simulation results because an ideal model of the SYSMON is used.

The `ug580_setup.tcl` download file provides TCL examples that can be used with the Vivado Hardware Manager to access SYSMON DRP registers.

RECOMMENDED: *Enable averaging when monitoring the on-chip sensors in a typical application to minimize any noise impacts. This is especially true if the automatic alarm functions are used.*

Averaging is enabled for the on-chip sensors in this instantiation example.

Apart from initializing the alarm threshold registers and the automatic channel sequencer register, the configuration registers need to be initialized to enable alarm outputs, sequencer modes, and ADC clock divider (see Configuration Registers (40h to 44h) for more information). Here is an instantiation in Verilog of the SYSMON example design:

```
`timescale 1ns / 1ps
module ug580 (
    input DCLK, // Clock input for DRP
    input RESET,
    input [15:0] VAUXP, VAUXN,  // Auxiliary analog channel inputs
    input VP, VN,// Dedicated and Hardwired Analog Input Pair
  inout I2C_SCLK, // uncomment when using I2C DRP interface
  inout I2C_SDA,   // uncomment when using I2C DRP interface

    output reg [15:0] MEASURED_TEMP, MEASURED_VCCINT,
    output reg [15:0] MEASURED_VCCAUX, MEASURED_VCCBRAM,
    output reg [15:0] MEASURED_AUX0, MEASURED_AUX1,
    output reg [15:0] MEASURED_AUX2, MEASURED_AUX3,

    output wire [15:0] ALARM,
    output wire [5:0]  CHANNEL,
    output wire        OT,
    output wire        SYSMON_EOC,
    output wire        SYSMON_EOS
    );

    wire busy;
    wire [5:0] channel;
    wire drdy;
    wire eoc;
    wire eos;

  wire i2c_sclk_in;
  wire i2c_sclk_ts;
  wire i2c_sda_in;
  wire i2c_sda_ts;

    reg [7:0] daddr;
    reg [15:0] di_drp;
    wire [15:0] do_drp;

    reg [1:0]  den_reg;
    reg [1:0]  dwe_reg;

    reg [7:0]   state = init_read;
    parameter       init_read       = 8'h00,
                    read_waitdrdy   = 8'h01,
                    write_waitdrdy  = 8'h03,
                    read_reg00      = 8'h04,
                    reg00_waitdrdy  = 8'h05,
                    read_reg01      = 8'h06,
                    reg01_waitdrdy  = 8'h07,
                    read_reg02      = 8'h08,
                    reg02_waitdrdy  = 8'h09,
                    read_reg06      = 8'h0a,
                    reg06_waitdrdy  = 8'h0b,
                    read_reg10      = 8'h0c,
                    reg10_waitdrdy  = 8'h0d,
```

```
                read_reg11      = 8'h0e,
                reg11_waitdrdy  = 8'h0f,
                read_reg12      = 8'h10,
                reg12_waitdrdy  = 8'h11,
                read_reg13      = 8'h12,
                reg13_waitdrdy  = 8'h13;

always @(posedge DCLK)
   if (RESET) begin
      state   <= init_read;
      den_reg <= 2'h0;
      dwe_reg <= 2'h0;
      di_drp  <= 16'h0000;
   end
   else
      case (state)
      init_read : begin
         daddr = 8'h40;
         den_reg = 2'h2; // performing read
         if (EOC == 0 ) state <= read_waitdrdy;
         end
      read_waitdrdy :
         if (EOC ==1)  begin
            di_drp = do_drp  & 16'h03_FF; //Clearing AVG bits for Configreg0
            daddr = 8'h40;
            den_reg = 2'h2;
            dwe_reg = 2'h2; // performing write
            state = write_waitdrdy;
         end
         else begin
            den_reg = { 1'b0, den_reg[1] } ;
            dwe_reg = { 1'b0, dwe_reg[1] } ;
            state = state;
         end
      write_waitdrdy :
         if (drdy ==1) begin
            state = read_reg00;
            end
         else begin
            den_reg = { 1'b0, den_reg[1] } ;
            dwe_reg = { 1'b0, dwe_reg[1] } ;
            state = state;
         end
      read_reg00 : begin
         daddr   = 8'h00;
         den_reg = 2'h2; // performing read
         if (eos == 1) state   <=reg00_waitdrdy;
         end
      reg00_waitdrdy :
         if (drdy ==1)  begin
            MEASURED_TEMP = do_drp;
            state <=read_reg01;
            end
         else begin
            den_reg = { 1'b0, den_reg[1] } ;
            dwe_reg = { 1'b0, dwe_reg[1] } ;
            state = state;
         end
      read_reg01 : begin
```

```
   daddr   = 8'h01;
   den_reg = 2'h2; // performing read
   state   <=reg01_waitdrdy;
   end
   reg01_waitdrdy :
   if (drdy ==1)  begin
      MEASURED_VCCINT = do_drp;
      state <=read_reg02;
      end
   else begin
      den_reg = { 1'b0, den_reg[1] } ;
      dwe_reg = { 1'b0, dwe_reg[1] } ;
      state = state;
   end
read_reg02 : begin
   daddr   = 8'h02;
   den_reg = 2'h2; // performing read
   state   <=reg02_waitdrdy;
   end
reg02_waitdrdy :
   if (drdy ==1)  begin
      MEASURED_VCCAUX = do_drp;
      state <=read_reg06;
      end
   else begin
      den_reg = { 1'b0, den_reg[1] } ;
      dwe_reg = { 1'b0, dwe_reg[1] } ;
      state = state;
   end
read_reg06 : begin
   daddr   = 8'h06;
   den_reg = 2'h2; // performing read
   state   <=reg06_waitdrdy;
   end
reg06_waitdrdy :
   if (drdy ==1)  begin
      MEASURED_VCCBRAM = do_drp;
      state <= read_reg10;
   end
   else begin
      den_reg = { 1'b0, den_reg[1] } ;
      dwe_reg = { 1'b0, dwe_reg[1] } ;
      state = state;
   end
read_reg10 : begin
      daddr   = 8'h10;
      den_reg = 2'h2; // performing read
      state   <= reg10_waitdrdy;
   end
reg10_waitdrdy :
   if (drdy ==1)  begin
      MEASURED_AUX0 = do_drp;
      state <= read_reg11;
   end
   else begin
      den_reg = { 1'b0, den_reg[1] } ;
      dwe_reg = { 1'b0, dwe_reg[1] } ;
      state = state;
   end
```

```
            read_reg11 : begin
               daddr   = 8'h11;
               den_reg = 2'h2; // performing read
               state   <= reg11_waitdrdy;
               end
            reg11_waitdrdy :
               if (drdy ==1)  begin
                  MEASURED_AUX1 = do_drp;
                  state <= read_reg12;
                  end
               else begin
                  den_reg = { 1'b0, den_reg[1] } ;
                  dwe_reg = { 1'b0, dwe_reg[1] } ;
                  state = state;
               end
            read_reg12 : begin
               daddr   = 8'h12;
               den_reg = 2'h2; // performing read
               state   <= reg12_waitdrdy;
               end
            reg12_waitdrdy :
               if (drdy ==1)  begin
                  MEASURED_AUX2= do_drp;
                  state <= read_reg13;
                  end
               else begin
                  den_reg = { 1'b0, den_reg[1] } ;
                  dwe_reg = { 1'b0, dwe_reg[1] } ;
                  state = state;
               end
            read_reg13 : begin
               daddr   = 8'h13;
               den_reg = 2'h2; // performing read
               state   <= reg13_waitdrdy;
               end
            reg13_waitdrdy :
               if (drdy ==1)  begin
                  MEASURED_AUX3= do_drp;
                  state <=read_reg00;
                  daddr   = 8'h00;
               end
               else begin
                  den_reg = { 1'b0, den_reg[1] } ;
                  dwe_reg = { 1'b0, dwe_reg[1] } ;
                  state = state;
               end
            endcase

   SYSMONE1 #(// Initializing the SYSMON Control Registers
        .INIT_40(16'h9000),// averaging of 16 selected for external channels
        .INIT_41(16'h2ef0),// Continuous Seq Mode, Disable unused ALMs, Enable calibration
        .INIT_42(16'h0400),// Set DCLK divides
        .INIT_43(16'h2ef0),// CONFIG3
        .INIT_46(16'h0001),// CHSEL0 - enable USER0
        .INIT_47(16'h0000),// SEQAVG0 disabled
        .INIT_48(16'h4701),// CHSEL1 - enable Temp VCCINT, VCCAUX, VCCBRAM, and calibration
        .INIT_49(16'h000f),// CHSEL2 - enable aux analog channels 0 - 3
        .INIT_4A(16'h0000),// SEQAVG1 disabled
        .INIT_4B(16'h0000),// SEQAVG2 disabled
```

```
      .INIT_4C(16'h0000),// SEQINMODE0
      .INIT_4D(16'h0000),// SEQINMODE1
      .INIT_4E(16'h0000),// SEQACQ0
      .INIT_4F(16'h0000),// SEQACQ1
      .INIT_50(16'hB723),// Temp upper alarm trigger 85°C for on-chip ref
      .INIT_51(16'h5999),// Vccint upper alarm limit 1.05V
      .INIT_52(16'hA147),// Vccaux upper alarm limit 1.89V
      .INIT_53(16'hCB93),// OT upper alarm limit 125°C - see Thermal Management
      .INIT_54(16'hAA5F),// Temp lower alarm reset 60°C for on-chip ref
      .INIT_55(16'h5111),// Vccint lower alarm limit 0.95V
      .INIT_56(16'h91Eb),// Vccaux lower alarm limit 1.71V
      .INIT_57(16'hAF7B),// OT lower alarm reset 70°C - see Thermal Management
      .INIT_58(16'h5999),// VCCBRAM upper alarm limit 1.05V
      .INIT_5C(16'h5111), // VUSER0 upper alarm limit 1.05V
      .INIT_60(16'h5999), // VUSER1 upper alarm limit 1.05V
      .INIT_61(16'h5999), // VUSER2 upper alarm limit 1.05V
      .INIT_62(16'h5999), // VUSER3 upper alarm limit 1.05V
      .INIT_63(16'h5999), // VCCBRAM lower alarm limit 1.05V
      .INIT_64(16'h5999), // VCCADC upper alarm limit 1.05V
      .INIT_68(16'h5111), // VUSER0 lower alarm limit 0.95V
      .INIT_69(16'h5111), // VUSER1 lower alarm limit 0.95V
      .INIT_6A(16'h5111), // VUSER2 lower alarm limit 0.95V
      .INIT_6B(16'h5111), // VUSER3 lower alarm limit 0.95V
      .INIT_6C(16'h5111), // VCCBRAM lower alarm limit 0.95V
      .INIT_78(16'h0000), // reserved
      .INIT_79(16'h0000), // reserved
      .SYSMON_VUSER0_BANK(66),
      .SYSMON_VUSER0_MONITOR("VCCO"),
      .SIM_MONITOR_FILE("design.txt")// Analog Stimulus file for simulation
  )
  SYSMON_INST (// Connect up instance IO. See UG580 for port descriptions
      .CONVST (1'b0),// not used
      .CONVSTCLK  (1'b0), // not used
      .DADDR  (daddr),
      .DCLK   (DCLK),
      .DEN    (den_reg[0]),
      .DI     (di_drp),
      .DWE    (dwe_reg[0]),
      .RESET  (RESET),
      .VAUXN  (VAUXN),
      .VAUXP  (VAUXP),
      .ALM    (ALARM),
      .BUSY   (busy),
      .CHANNEL(CHANNEL),
      .DO     (do_drp),
      .DRDY   (drdy),
      .EOC    (eoc),
      .EOS    (eos),
      .JTAGBUSY   (),// not used
      .JTAGLOCKED (),// not used
      .JTAGMODIFIED   (),// not used
    .I2C_SCLK   (i2c_sclk_in), // uncomment when using I2C DRP interface
    .I2C_SCLK_TS (i2c_sclk_ts), // uncomment when using I2C DRP interface
    .I2C_SDA    (i2c_sda_in), // uncomment when using I2C DRP interface
    .I2C_SDA_TS (i2c_sda_ts), // uncomment when using I2C DRP interface
      .OT     (OT),
      .MUXADDR    (),// not used
      .VP     (VP),
      .VN     (VN)
```

```
);

IOBUF I2C_SCLK_inst (
.O(i2c_sclk_in),      // Buffer output
.IO(I2C_SCLK),    // Buffer inout port (connect directly to top-level port)
.I(1'b0),      // Buffer input
.T(i2c_sclk_ts)       // 3-state enable input, high=input, low=output
);

IOBUF I2C_SDA_inst (
.O(i2c_sda_in),      // Buffer output
.IO(I2C_SDA),    // Buffer inout port (connect directly to top-level port)
.I(1'b0),      // Buffer input
.T(i2c_sda_ts)       // 3-state enable input, high=input, low=output
);

   assign SYSMON_EOC = eoc;
  assign SYSMON_EOS = eos;

endmodule
```

Example Design Test Bench

The next example is a test bench that sets up a DCLK of 50 MHz. Analog signals are read from the analog stimulus file by the simulation model. The SIM_MONITOR_FILE attribute used in the SYSMONE1 instantiation points the model to the location of this file. The analog stimulus file used for this example is shown here:

```
TIME    VAUXP[0] VAUXN[0] VAUXP[1] VAUXN[1] VAUXP[2] VAUXN[2] VAUXP[3] VAUXN[3] Temp VCCINT VCCAUX VCCBRAM VUSER0
00000   0.005    0.0      0.2       0.0      0.5      0.0      0.1      0.0      25   1.0    1.8     1.0     1.0
34000   0.020    0.0      0.400     0.0      0.49     0.0      0.2      0.0      85   1.05   1.9     1.05    1.0
67000   0.049    0.0      0.600     0.0      0.51     0.0      0.5      0.0      105  0.95   1.71    0.95    1.0
100000  0.034    0.0      0.900     0.0      0.53     0.0      0.0      0.0      0    1.00   1.8     1.0     1.0
```

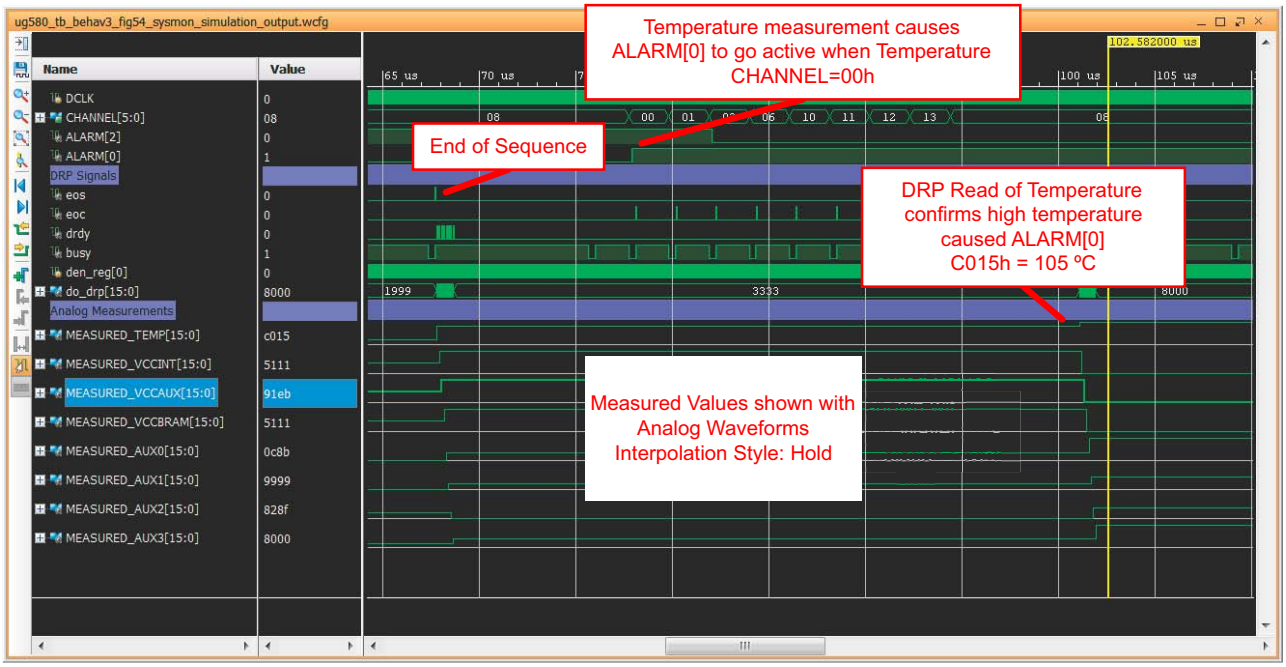
The format of the analog stimulus file is based on space- or tab-delimited data and can be created in a spreadsheet. Many tools such as SPICE simulators or equipment such as oscilloscopes export comma-separated value (CSV) formats, that can be manipulated in a spreadsheet to generate an analog stimulus file for simulation. All time stamp information must be listed in the first column. Other columns list the analog values for the on-chip sensors and external analog inputs. The order of the columns is not important. The only requirement is that time stamp information is listed in the first column. For each time stamp added to the first column, a corresponding value is added to the other columns. Only the required analog input channel columns must be listed. In this example, only the on-chip sensors and auxiliary channels zero to three are listed in the analog stimulus file. In this stimulus file, the temperature moves 85°C to 105°C at 67 µs after the start of the simulation. The temperature alarm (ALARM[0]) becomes active High shortly after this event when the temperature is measured by the ADC (see Figure 5-4). The upper alarm threshold for temperature has been set to B723h or 85°C.

IMPORTANT: *When simulating the SYSMON, the simulation models always assume the on-chip voltage reference is being used. For the internal temperature transfer function, see Equation 2-7 for SYSMONE1 and Equation 2-11 for SYSMONE4.*

```
`timescale 1ns / 1ps
module ug580_tb;
    reg [15:0]      VAUXP, VAUXN;
    reg             RESET;
    reg             DCLK;

    wire [15:0]     MEASURED_TEMP, MEASURED_VCCINT, MEASURED_VCCAUX;
    wire [15:0]     MEASURED_VCCBRAM, MEASURED_AUX0, MEASURED_AUX1;
    wire [15:0]     MEASURED_AUX2, MEASURED_AUX3;
    wire [15:0]     ALARM;

initial
    begin
                DCLK = 0;
                RESET = 1;
        #100    RESET = 0;
    end

always #(10) DCLK= ~DCLK;


// Instantiate the Unit Under Test (UUT)
ug580 uut (
    .VAUXP  (VAUXP),
    .VAUXN  (VAUXN),
    .RESET  (RESET),
    .ALARM  (ALARM),
    .DCLK   (DCLK),

    .MEASURED_TEMP    (MEASURED_TEMP),
    .MEASURED_VCCINT  (MEASURED_VCCINT),
    .MEASURED_VCCAUX  (MEASURED_VCCAUX),
    .MEASURED_VCCBRAM (MEASURED_VCCBRAM),
    .MEASURED_AUX0    (MEASURED_AUX0),
    .MEASURED_AUX1    (MEASURED_AUX1),
    .MEASURED_AUX2    (MEASURED_AUX2),
    .MEASURED_AUX3    (MEASURED_AUX3)
);

endmodule
```

Simulation Output

The simulation output in Figure 5-4 shows the user-defined sequence in continuous sampling mode. The channels monitored in the sequence can be seen by looking at the CHANNEL[5:0] bus. The sequence is 8, 0, 1, 2, 3, 6, 10, 11, 12, and 13, which corresponds to calibration, temperature, V_{CCINT}, V_{CCAUX}, V_{CCBRAM}, AUX0, AUX1, AUX2, and AUX3. Then the sequence repeats. The calibration channel takes longer to complete than the other channels. This is because the calibration routine involves three conversions (measurements) using the ADC. The measured results are shown using the analog waveform settings for the Vivado simulator.



X16845-042016

Figure 5-4: **SYSMON Simulation Output**

The design initially performs a DRP write to register 40h (configuration register 0) to set the AVG1 and AVG0 bits to 00 (see Figure 5-5). This disables the averaging functionality for simulation. The DRP Write has been simulated with Config Reg0 (40H) set to 903FH to show how the DRP writes 003FH. Do_drp is updated to 003FH after the DRP write is completed.

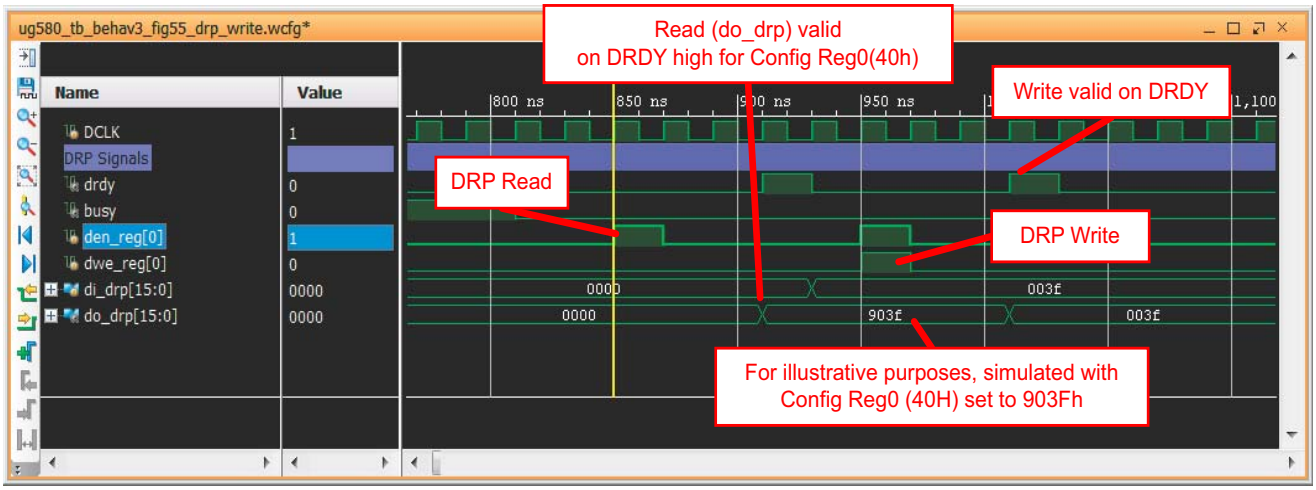
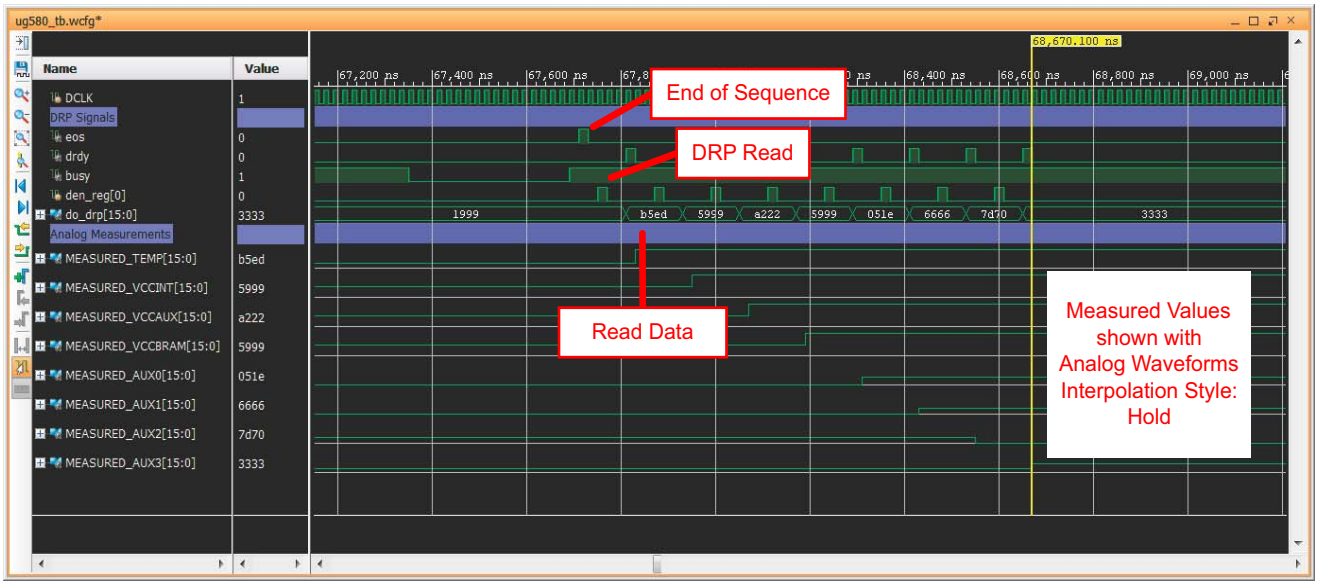


Figure 5-5: **DRP Write at 4 µs**

As shown in Figure 5-6, when the EOS signal pulses High for one DCLK period at the end of a sequence, the test bench reads the status registers. The simulation model uses the full 16-bit ADC conversion result because it is an ideal model of the ADC. For example, V_{CCINT} is shown as 5999h for 1.05V. From Equation 2-16, 1.05V is 166h for 10-bit data. For 16-bit data, 5999h = 1.05 x (65536/3). This is a 10-bit MSB justified result. However, the six LSBs of the status register also contain data that would be 011001b if the ADC was an ideal 16-bit ADC.



X16847-120216

Figure 5-6: **DRP Read of Status Register at EOS**

The temperature output transitions High (see Figure 5-4). The ALARM[0] becomes active at the end of the conversion on the temperature channel (`00h`) when the result is loaded in the status register. The result is read from the status register by the design when EOS next transitions High. The temperature is 105°C (as set in the stimulus file) and is greater than 85°C limit set when the SYSMONE1 was instantiated in the design (.INIT_50 (`16'hb5ed`)).

The V_{CCAUX_ALARM} output transitions High during the second pass through the sequence (see Figure 5-4). The alarm becomes active at the end of the conversion on the V_{CCAUX} channel when the result is loaded in the status register. The result is read from the status register by the test bench when EOS next transitions High. The V_{CCAUX} is ~1.9V (as set in the stimulus).

Appendix A

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

These documents and website provide supplemental material useful with this guide:

1. UltraScale and UltraScale+ device packaging and pinout user guides:
   - *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification User Guide* (UG575)
   - *Zynq UltraScale+ MPSoC Packaging and Pinouts User Guide* (UG1075)
2. *XADC Layout Guidelines* (XAPP554)
3. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
4. *UltraScale Architecture Configuration User Guide* (UG570)
5. *Driving the Xilinx Analog-to-Digital Converter* (XAPP795)
6. *7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide* (UG480)
7. UltraScale and UltraScale+ device data sheets:
   - *UltraScale Architecture and Products Overview* (DS890)
   - *Zynq UltraScale+ MPSoC Overview* (DS891)
   - *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925)
   - *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892)
   - *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922)
   - *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893)
   - *Virtex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS923)
8. *UltraScale Architecture Libraries Guide* (UG974)
9. *System Management Wizard LogiCORE IP Product Guide (AXI)* (PG185)
10. *Zynq UltraScale+ MPSoC Technical Reference Manual* (UG1085)
11. Power Management Bus specification

Please Read: Important Legal Notices