

# **7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter *User Guide***

UG480 (v1.2) October 25, 2012



The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials, or to advise you of any corrections or update. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

© Copyright 2011–2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/11	1.0	Initial Xilinx release.
03/28/11	1.1	Added "Dual 12-Bit MSPS Analog-to-Digital Converter" to document title. Modified first paragraph and added second paragraph in <a href="#">Chapter 1, Introduction and Quick Start</a> . Added <a href="#">Table 1-1</a> . Removed the <i>Thermal Diode (DXP and DXN)</i> section from Chapter 4.
10/25/12	1.2	This version was updated to include information for the Zynq-7000 All Programmable SoC devices: Added "Zynq-7000 All Programmable SoC" to document title. In <a href="#">Figure 1-1</a> Zynq-7000 All Programmable SoC information was added and the control and status registers changed from 32 x 16 bits to 64 x 16 bits. The <i>System Monitor Support</i> section changed to <a href="#">Differences versus Virtex-5 and Virtex-6 System Monitor</a> . Functionality for the XADC block in 7 series FPGAs is now defined for previously undefined status registers in subsequent chapters. Capacitor values in <a href="#">Figure 1-2</a> changed, and a note was added about the placement of 100 nF decoupling capacitors. In <a href="#">Table 1-1</a> , for $V_{REFP,0}$ , GND changed to GNDADC, accurate reference <i>source</i> changed to accurate reference <i>IC</i> , and $V_{P,0}$ and $V_{N,0}$ types were changed to <i>dedicated</i> analog inputs. Footnotes were added to support 7 series and Zynq device pin packages. A note about application guidelines was added after <a href="#">Table 1-1</a> . Information about auxiliary analog channels was added to the <a href="#">External Analog Inputs</a> section. In <a href="#">Table 1-2</a> , ports ALM[0] through [ALM[3] have <i>XADC</i> removed from the description. Alarms specific to Zynq-7000 All Programmable SoC devices (ALM[4], ALM[5], and ALM[6]) were added. The example in <a href="#">Example Instantiation</a> was updated. In <a href="#">ADC and Sensors</a> , information about an analog input signal of 200 mV was modified, and <a href="#">Equation 1-1</a> was added. A new <a href="#">Zynq-7000 All Programmable SoC (AP SoC)</a> subsection was added to the end of the chapter.

Date	Version	Revision
10/25/12	1.2 (Cont'd)	<p>In <a href="#">ADC Transfer Functions</a> MSBs are defined as left-most bits. In <a href="#">Auxiliary Analog Inputs</a> <i>in the instantiation</i> was changed to <i>on the primitive</i>, and information that configuration is automatic when analog inputs are connected was added. The Note on <a href="#">page 27</a> was expanded to clarify device support of auxiliary analog channels. <a href="#">Equation 2-2</a> was added (unipolar mode), <a href="#">Equation 2-1</a>, <a href="#">Equation 2-3</a>, and <a href="#">Equation 2-4</a> were modified (“10” was changed to “9”), and the paragraph following <a href="#">Equation 2-4</a> was added. Information about additional external resistance was added to <a href="#">Unipolar Input Signals</a>. In <a href="#">Temperature Sensor</a> this sentence was deleted: “The on-chip temperature sensor has a maximum measurement error of <math>\pm 4^{\circ}\text{C}</math> over a range of <math>-40^{\circ}\text{C}</math> to <math>+125^{\circ}\text{C}</math>.” Information was added to the <a href="#">Power Supply Sensor</a> section about which supplies are monitored and where measurements are stored for Zynq-7000 All Programmable SoC devices.</p> <p>Min/max register lists were updated for Zynq-7000 All Programmable SoC devices in <a href="#">Figure 3-1</a>. In <a href="#">Table 3-1</a> the <math>V_{\text{REFN}}</math> description was updated, new Zynq-7000 device channels <math>V_{\text{CCPINT}}</math>, <math>V_{\text{CCPAUX}}</math>, and <math>V_{\text{CCO\_DDR}}</math> were added. Zynq-7000 All Programmable SoC status registers were added to <a href="#">Figure 3-1</a>. Flag registers for DI5 through DI8 in <a href="#">Figure 3-2</a> changed and DIS was removed from <a href="#">Table 3-2</a>. <a href="#">XADC Calibration Coefficients</a> were added. Configuration registers ALM[4], ALM[5], and ALM[6] were added to <a href="#">Figure 3-3</a>. In <a href="#">Table 3-3</a>, a new row was added for bits DI19 to DI11. In <a href="#">Table 3-7</a>, rows for ADC channels 13, 14, and 15 were added. Zynq-7000 All Programmable SoC alarm threshold register information was added to <a href="#">Alarm Registers (50h to 5Fh)</a>. A note in the section <a href="#">DRP JTAG Interface</a> mentions conditions in which the external JTAG access is disabled for Zynq devices. Section <a href="#">Zynq-7000 AP SoC Processing System (PS) to XADC Dedicated Interface</a> was added to the end of the chapter.</p> <p>In <a href="#">XADC Operating Modes</a>, <i>operate both ADCs in parallel</i> in the first paragraph changed to <i>operate both ADCs in lock step</i>. In <a href="#">Single Channel Mode</a>, the first sentence was deleted (“Single channel mode is the most basic way users can modify the operation of the XADC.”) Sequence numbers and descriptions in <a href="#">Table 4-1</a> and <a href="#">Table 4-2</a> changed and were added for Zynq-7000 All Programmable SoC devices. In <a href="#">Sequencer Modes</a>, default mode sequences were changed in <a href="#">Table 4-3</a> and added for Zynq-7000 All Programmable SoC devices. The sentence before <a href="#">Table 4-4</a> was deleted. The footnote in <a href="#">Table 4-4</a> and <a href="#">Table 4-6</a> was enhanced to mention auxiliary analog channel support. <a href="#">Table 4-7</a> was updated to include Zynq-7000 All Programmable SoC channels. Alarm threshold registers for the Zynq-7000 All Programmable SoC were added to <a href="#">Table 4-8</a>. A new section <a href="#">XADC Enhanced Linearity Mode</a> and <a href="#">Figure 4-4</a> were added to the end of the chapter.</p> <p>The <a href="#">Dynamic Reconfiguration Port (DRP) Timing</a> section was added. Removed <a href="#">Table 5-1: XADC Timing Information</a>.</p> <p>In <a href="#">Reference Inputs (<math>V_{\text{REFP}}</math> and <math>V_{\text{REFN}}</math>)</a>, added “Noise on the reference voltage also adds noise to the ADC conversion and results in more code transition noise or poorer than expected SNR” to the end of the first paragraph. Capacitor values were changed, a new 10 <math>\mu\text{F}</math> capacitor was added, and notes were added to <a href="#">Figure 6-1</a>. Notes were added to <a href="#">Figure 6-5</a>. <a href="#">XADC Software Support</a> was completely replaced. In that section, the Verilog instantiation was replaced. <a href="#">Figure 6-6: Analog Stimulus File</a> and <a href="#">Figure 6-7: XADC Simulation Output</a> were deleted. In the new section, <a href="#">Figure 6-5</a> through <a href="#">Figure 6-9</a> were added.</p>



# Table of Contents

---

Revision History .....	2
<b>Preface: About This Guide</b>	
Guide Contents .....	9
Additional Resources .....	9
References .....	10
<b>Chapter 1: Introduction and Quick Start</b>	
XADC Overview .....	11
Differences versus Virtex-5 and Virtex-6 System Monitor .....	12
XADC Pinout Requirements .....	13
Dedicated Package Pins .....	13
External Analog Inputs .....	15
Instantiating the XADC .....	16
XADC Ports .....	16
XADC Attributes .....	18
Example Instantiation .....	19
ADC and Sensors .....	20
Analog-to-Digital Converter .....	20
Temperature Sensor .....	21
Power Supply Sensors .....	21
<b>Chapter 2: Analog-to-Digital Converter</b>	
ADC Transfer Functions .....	23
Unipolar Mode .....	24
Bipolar Mode .....	25
Analog Inputs .....	26
Auxiliary Analog Inputs .....	26
Adjusting the Acquisition Settling Time .....	27
Analog Input Description .....	27
Unipolar Input Signals .....	28
Bipolar Input Signals .....	29
Temperature Sensor .....	30
Power Supply Sensor .....	31
<b>Chapter 3: XADC Register Interface</b>	
Status Registers .....	34
Flag Register .....	37
XADC Calibration Coefficients .....	38
Calibration Coefficients Definition .....	38
Control Registers .....	39
Configuration Registers (40h to 42h) .....	39
Test Registers (43h to 47h) .....	43

Channel Sequencer Registers (48h to 4Fh) . . . . .	43
Alarm Registers (50h to 5Fh) . . . . .	43
<b>DRP JTAG Interface</b> . . . . .	44
XADC DRP JTAG Write Operation . . . . .	44
XADC DRP JTAG Read Operation . . . . .	45
JTAG DRP Commands . . . . .	46
DRP Arbitration . . . . .	46
JTAGBUSY . . . . .	47
JTAGMODIFIED . . . . .	47
JTAGLOCKED . . . . .	47
XADC JTAG Reset . . . . .	47
<b>Zynq-7000 AP SoC Processing System (PS) to XADC Dedicated Interface</b> . . . .	48

## Chapter 4: XADC Operating Modes

<b>Single Channel Mode</b> . . . . .	49
<b>Automatic Channel Sequencer</b> . . . . .	49
ADC Channel Selection Registers (48h and 49h) . . . . .	49
ADC Channel Averaging (4Ah and 4Bh) . . . . .	51
ADC Channel Analog-Input Mode (4Ch and 4Dh) . . . . .	52
ADC Channel Settling Time (4Eh and 4Fh) . . . . .	52
<b>Sequencer Modes</b> . . . . .	52
Default Mode . . . . .	52
Single Pass Mode . . . . .	53
Continuous Sequence Mode . . . . .	53
Simultaneous Sampling Mode . . . . .	53
Timing . . . . .	54
Independent ADC Mode . . . . .	54
Sequencer Operation . . . . .	55
Timing . . . . .	56
<b>External Multiplexer Mode</b> . . . . .	56
External Multiplexer Operation . . . . .	57
<b>Maximum and Minimum Status Registers</b> . . . . .	58
<b>Automatic Alarms</b> . . . . .	59
Supply Sensor Alarms . . . . .	60
Thermal Management . . . . .	60
XADC Enhanced Linearity Mode . . . . .	61
Changes to Offset and Gain Calibration Operation . . . . .	61

## Chapter 5: XADC Timing

<b>Continuous Sampling</b> . . . . .	63
Acquisition Phase . . . . .	64
Conversion Phase . . . . .	64
<b>Event-Driven Sampling</b> . . . . .	65
<b>Dynamic Reconfiguration Port (DRP) Timing</b> . . . . .	66

## Chapter 6: Application Guidelines

<b>Reference Inputs (<math>V_{REFP}</math> and <math>V_{REFN}</math>)</b> . . . . .	69
<b>Analog Power Supply and Ground (<math>V_{CCADC}</math> and <math>GNDADC</math>)</b> . . . . .	69

---

<b>External Analog Inputs</b> .....	71
Anti-Alias Filters .....	72
Over and Under Voltages .....	72
<b>PC Board Design Guidelines</b> .....	72
XADC Software Support .....	74
Using XADC Instantiation Wizard .....	74
Example Design Instantiation .....	75
Example Design Test Bench .....	79
Simulation Output .....	85





# About This Guide

---

Xilinx® 7 series FPGAs include three scalable, optimized FPGA families that are all designed for lowest power to enable a common design to scale across families for optimal power, performance, and cost. The Artix™-7 family is optimized for lowest cost and absolute power for the highest volume applications. The Virtex®-7 family is optimized for highest system performance and capacity. The Kintex™-7 family is an innovative class of FPGAs optimized for the best price-performance. This guide serves as a technical reference describing the 7 series FPGAs and Zynq™-7000 All Programmable SoC's XADC, a dual 12-bit, 1 MSPS analog-to-digital converter with on-chip sensors.

This user guide is part of an overall set of documentation on the 7 series FPGAs. This documentation is available on the Xilinx website at [www.xilinx.com/7](http://www.xilinx.com/7).

## Guide Contents

This manual contains these chapters:

- [Chapter 1, Introduction and Quick Start](#)
- [Chapter 2, Analog-to-Digital Converter](#)
- [Chapter 3, XADC Register Interface](#)
- [Chapter 4, XADC Operating Modes](#)
- [Chapter 5, XADC Timing](#)
- [Chapter 6, Application Guidelines](#)

## Additional Resources

To find references about boards, documentation, tutorials, and more for the Analog Microcontroller Software Development Kit, refer to their landing page [\[Ref 2\]](#).

To find additional Xilinx documentation, see the Xilinx website at:

<http://www.xilinx.com/support/documentation/index.htm>.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

<http://www.xilinx.com/support>.

To find Xilinx Agile Mixed Signal (AMS) information, see the Xilinx website at:

<http://www.xilinx.com/ams>

## References

Additional material useful to this document can be found here:

1. [PG019](#), *LogiCORE IP AXI XADC (v1.00.a) Product Guide*
2. [UG475](#), *7 Series FPGAs Packaging and Pinout Advance Specification*
3. [UG470](#), *7 Series FPGAs Configuration User Guide*
4. [UG585](#), *Zynq-7000 All Programmable SoC Technical Reference Manual*
5. [UG772](#), *LogiCORE IP XADC Wizard User Guide*
6. [XAPP554](#), *XADC Layout Guidelines Application Note*

# Introduction and Quick Start

---

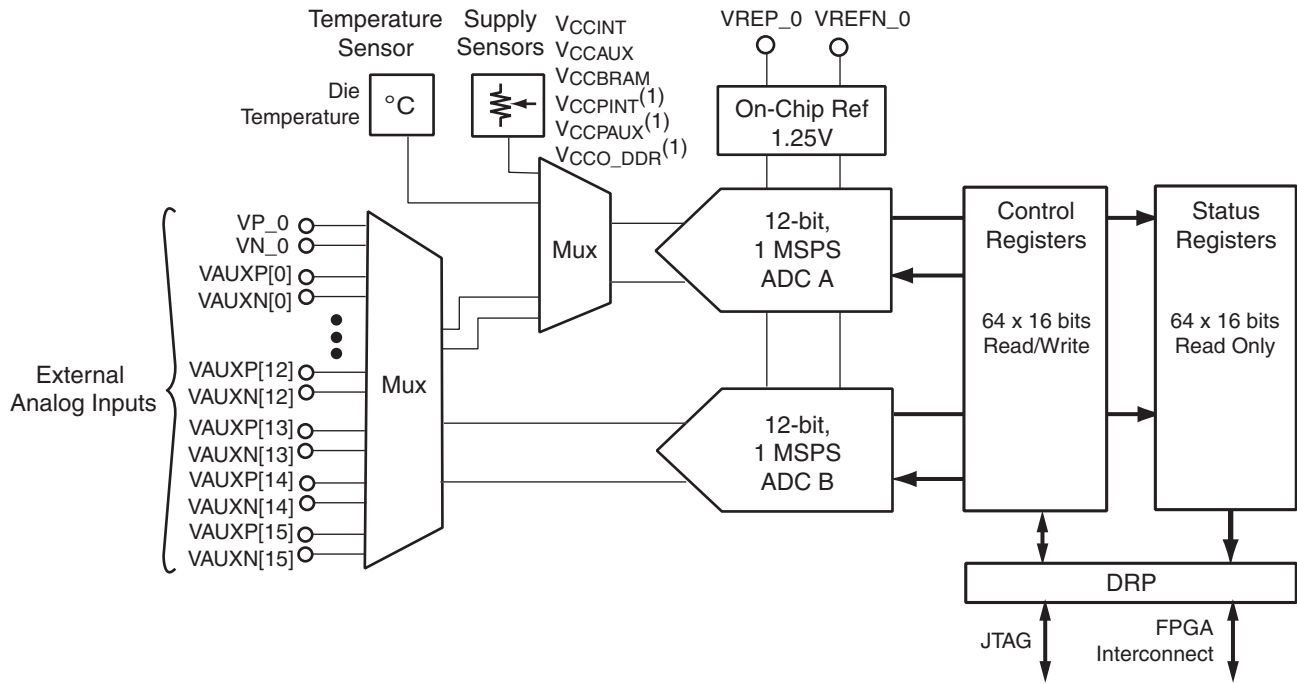
This chapter provides a brief overview of the Xilinx® 7 series FPGAs XADC functionality. The XADC is available in all Artix™, Kintex™, Virtex®, and Zynq™-7000 All Programmable SoC (AP SoC) devices.

The XADC is the basic building block that enables Agile Mixed Signal (AMS) functionality which is new to 7 series FPGAs. By combining high quality analog blocks with the flexibility of programmable logic, it is possible to craft customized analog interfaces for a wide range of applications. Please see <http://www.xilinx.com/ams> for more information.

This chapter contains only key information to allow a basic understanding of the XADC block. With this introduction, users can learn the pinout requirements and determine how to instantiate basic functionality in their designs. Subsequent chapters provide more detailed descriptions of the XADC functionality.

## XADC Overview

The XADC includes a dual 12-bit, 1 Mega sample per second (MSPS) ADC and on-chip sensors. These ADCs are fully tested and specified (see the respective 7 series FPGAs data sheet). The ADCs provide a general-purpose, high-precision analog interface for a range of applications. [Figure 1-1](#) shows a block diagram of the XADC. The dual ADCs support a range of operating modes, for example, externally triggered and simultaneous sampling on both ADCs (see [Chapter 4, XADC Operating Modes](#)) and various analog input signal types, for example, unipolar and differential (see [Chapter 2, Analog-to-Digital Converter](#)). The ADCs can access up to 17 external analog input channels.



UG480\_c1\_01\_04-612

Figure 1-1: XADC Block Diagram

Notes relevant to Figure 1-1:

1. Zynq-7000 All Programmable SoC devices only.

The XADC also includes a number of on-chip sensors that support measurement of the on-chip power supply voltages and die temperature. The ADC conversion data is stored in dedicated registers called status registers. These registers are accessible via the FPGA interconnect using a 16-bit synchronous read and write port called the Dynamic Reconfiguration Port (DRP). ADC conversion data is also accessible via the JTAG TAP. In the latter case, users are not required to instantiate the XADC because it is a dedicated interface that uses the existing FPGA JTAG infrastructure. As discussed later, if the XADC is not instantiated in a design, the device operates in a predefined mode (called *default mode*) that monitors on-chip temperature and supply voltages.

XADC operation is user defined by writing to the control registers using either the DRP or JTAG interface. It is also possible to initialize these register contents when the XADC is instantiated in a design using the block attributes.

## Differences versus Virtex-5 and Virtex-6 System Monitor

For Virtex®-5 and Virtex®-6 FPGA System Monitor users, the XADC functionality is fully backward compatible with legacy System Monitor designs. The XADC functionality and interface are familiar to those who have previously designed with the System Monitor. System Monitor designs are automatically retargeted to the XADC site by the software tools.

However, the XADC block in 7 series FPGAs contains a large number of new features and enhancements detailed in subsequent chapters. The new functionality is enabled by initializing previously undefined status registers and bit locations. Old System Monitor designs that did not initialize these new registers or bit locations behave exactly the same way as before.

# XADC Pinout Requirements

## Dedicated Package Pins

All XADC dedicated pins are located in bank 0 and thus have the `_0` suffix in the package file names. [Figure 1-2](#) shows the basic pinout requirements for the XADC. There are two recommended configurations. On the left, the XADC 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 (see [Chapter 6, Application Guidelines](#) for more information). Shared or common ground impedance is the most common way to introduce unwanted noise into analog circuits.

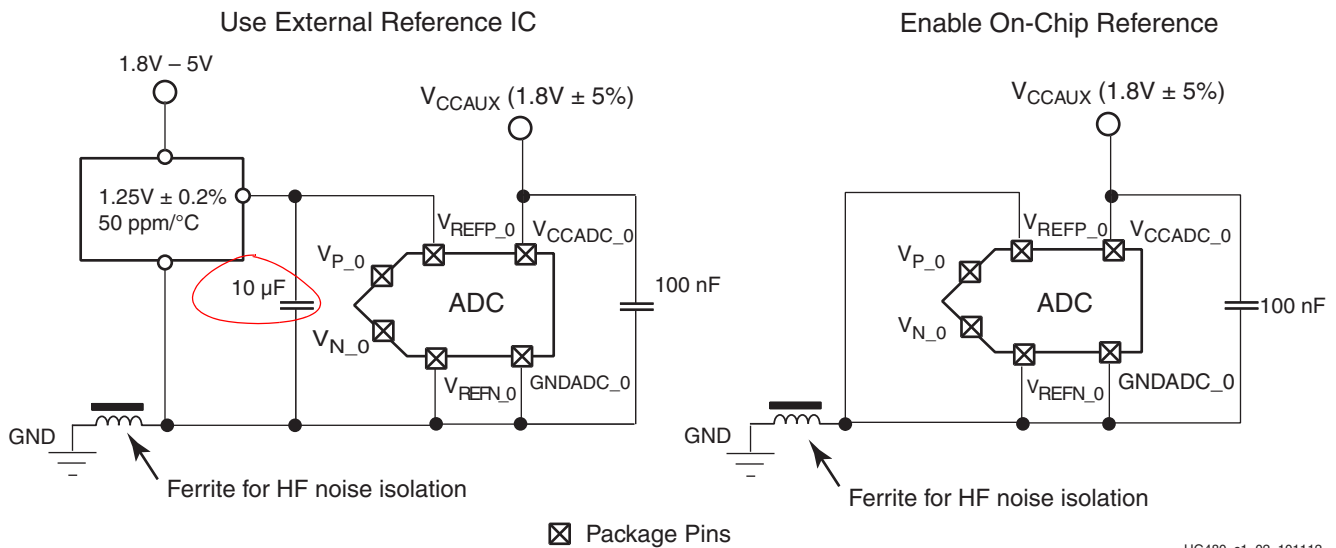


Figure 1-2: XADC Pinout Requirements

It is also possible to use an on-chip reference for the ADCs. To enable the on-chip reference source, the  $V_{REFP}$  pin must be connected to ground as shown on the right of [Figure 1-2](#). Where only basic on-chip thermal and supply monitoring is required, using the on-chip reference provides good performance. Users should consult the respective 7 series FPGAs data sheet to see the accuracy specifications when using the external and on-chip reference sources. [Table 1-1](#) lists the pins associated with the XADC and the recommended connectivity.

**Note:** It is also important to place the 100 nF decoupling capacitors as close as possible to the package balls to minimize inductance between the decoupling and package balls.

Table 1-1: XADC Package Pins

Package Pin	Type	Description
V <sub>CCADC_0</sub>	Power supply	This is the analog supply pin for the ADCs and other analog circuits in the XADC. It can be tied to the 1.8V V <sub>CCAUX</sub> supply however in a mixed-signal system the supply should be connected to a separate 1.8V analog if available. See <a href="#">Analog Power Supply and Ground (V<sub>CCADC</sub> and GNDADC)</a> , page 69 for more information. This pin should never be tied to GND. The pin should be tied to V <sub>CCAUX</sub> even if the XADC is not being used.
GNDADC_0	Power supply	This is the ground reference pin for the ADCs and other analog circuits in the XADC. It can be tied to the system ground via an isolating ferrite bead as shown in <a href="#">Figure 1-2</a> . In a mixed-signal system this pin should be tied to an analog ground plane if available, in which case the ferrite bean is not required. See <a href="#">Analog Power Supply and Ground (V<sub>CCADC</sub> and GNDADC)</a> , page 69 for more information. This pin should always be tied to GND even if the XADC is not being used.
V <sub>REFP_0</sub>	Reference voltage input	This pin should be tied to an external 1.25V accurate reference IC ( $\pm 0.2\%$ ) for best performance of the ADCs. It should be treated as an analog signal that together with the V <sub>REFN</sub> signal provides a differential 1.25V voltage. By connecting this pin to GNDADC (see <a href="#">Figure 1-2</a> ) an on-chip reference source ( $\pm 1\%$ ) is activated. This pin should always be connected to GNDADC if an external reference is not supplied. See <a href="#">Reference Inputs (V<sub>REFP</sub> and V<sub>REFN</sub>)</a> , page 69 for more information.
V <sub>REFN_0</sub>	Reference voltage input	This pin should be tied to GND pin of an external 1.25V accurate reference IC ( $\pm 0.2\%$ ) for best performance of the ADCs. It should be treated as an analog signal that together with the V <sub>REFP</sub> signal provides a differential 1.25V voltage. This pin should always be connected to GND even if an external reference is not supplied. See <a href="#">Reference Inputs (V<sub>REFP</sub> and V<sub>REFN</sub>)</a> , page 69 for more information.
V <sub>P_0</sub>	Dedicated analog input	This is the positive input terminal of the dedicated differential analog input channel (V <sub>P</sub> /V <sub>N</sub> ). The analog input channels are very flexible and support multiple analog input signal types. For more information see <a href="#">Analog Inputs</a> , page 26. This pin should be connected to GND if not used.
V <sub>N_0</sub>	Dedicated analog input	This is the negative input terminal of the dedicated differential analog input channel (V <sub>P</sub> /V <sub>N</sub> ). The analog input channels are very flexible and support multiple analog input signal types. For more information see <a href="#">Analog Inputs</a> , page 26. This pin should be connected to GND if not used.
_AD0P_ to _AD15P_ <sup>(1)(2)</sup>	Auxiliary analog inputs/digital I/O	These are multi-function pins that can support analog inputs or can be used as regular digital I/O (see <a href="#">Figure 1-1</a> ). These pins support up to 16 positive input terminals of the differential auxiliary analog input channels (V <sub>AUXP</sub> /V <sub>AUXN</sub> ). The analog input channels are very flexible and support multiple analog input signal types. For more information see <a href="#">Analog Inputs</a> , page 26. When not being used as analog input these pins can be treated like any other digital I/O.

Table 1-1: XADC Package Pins

Package Pin	Type	Description
<code>_AD0N_ to _AD15N_</code> <sup>(1)(2)</sup>	Auxiliary analog inputs/digital I/O	These are multi-function pins that can support analog inputs or can be used as regular digital I/O (see <a href="#">Figure 1-1</a> ). These pins support up to 16 negative input terminals of the differential auxiliary analog input channels ( $V_{AUXP}/V_{AUXN}$ ). The analog input channels are very flexible and support multiple analog input signal types. For more information see <a href="#">Analog Inputs, page 26</a> . When not being used as analog input these pins can be treated like any other digital I/O.

**Notes:**

1. FPGA I/Os that are analog input-enabled contain the `_ADxP_` and `_ADxN_` designation in the package file name, e.g., `IO_L1P_T0_AD0P_35` is the input pin for analog auxiliary channel  $V_{AUXP}[0]$ . `IO_L1N_T0_AD0N_35` is the input pin for analog auxiliary channel  $V_{AUXN}[0]$ . For more information, see [UG475, 7 Series FPGAs Packaging and Pinout Advance Specification \[Ref 2\]](#).
2. Auxiliary channels 6, 7, 13, 14, and 15 are not supported in Kintex-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC device package options. Users should consult the package file for the device.

**Note:** [Chapter 6, Application Guidelines](#), should be consulted before commencing any PC board layout. Board layout and external component choices can greatly impact the performance of the ADCs.

## External Analog Inputs

Apart from a single dedicated analog input pair ( $V_P/V_N$ ), the external analog inputs use dual-purpose I/O. These FPGA digital I/Os are individually nominated as analog inputs when the XADC is instantiated in a design. This document refers to these analog inputs as auxiliary analog inputs. A maximum of 16 auxiliary analog inputs are available. The auxiliary analog inputs are enabled by connecting the analog inputs on the XADC primitive to the top level of the design. When enabled as analog inputs, these package balls are unavailable as digital I/Os. It is also possible to enable the auxiliary analog inputs preconfiguration (for example, for PCB diagnostics) via the JTAG TAP (see [JTAG DRP Commands](#) for more information.) The auxiliary analog inputs do not require any user-specified constraints or pin locations. They do not require an I/O standard setting to be added to the user constraints file (UCF) or in the PlanAhead™ design tool. All configuration is automatic when the analog inputs are connected to the top level of the design.

All analog input channels are differential and require two package balls. Typically, the auxiliary analog inputs are allocated evenly over banks 15 and 35. However, users should consult the pinout information in [UG475, 7 Series FPGAs Packaging and Pinout Advance Specification](#) for a particular device and package combination [\[Ref 2\]](#). Analog-capable I/O have the `ADxP` or `ADxN` suffix on the I/O name in the package files. For example, auxiliary analog input channel 8 has associated package ball names ending with `AD8P` and `AD8N`. See [UG475, 7 Series FPGAs Packaging and Pinout Advance Specification](#) for more information [\[Ref 2\]](#). The auxiliary analog inputs have a fixed package ball assignment and cannot be moved.

**Note:** Auxiliary channels 6, 7, 13, 14, and 15 are not supported on Kintex-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 All Programmable SoC device package options. Users should consult the package file for the device.

## Instantiating the XADC

As mentioned previously, it is not necessary to instantiate the XADC in a design to access the on-chip monitoring capability. However, if the XADC is not instantiated in a design, the only way to access this information is via the JTAG TAP. To allow access to the status registers (measurement results) from the FPGA logic, the XADC must be instantiated. These subsections give a brief overview of the XADC primitive (ports and attributes).

### XADC Ports

Figure 1-3 shows the ports on the XADC primitive, and Table 1-2 describes the functionality of the ports.

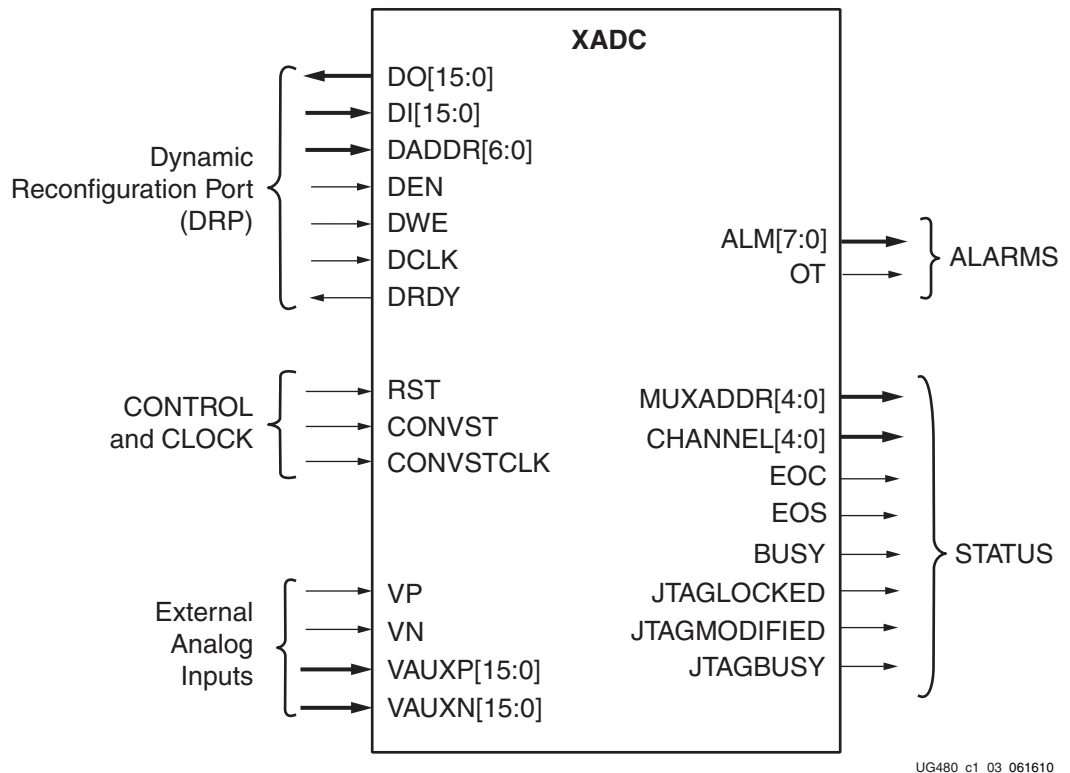


Figure 1-3: XADC Primitive Ports

Table 1-2: XADC Port Descriptions

Port	I/O	Description
DI[15:0]	Inputs	Input data bus for the DRP. <sup>(1)</sup>
DO[15:0]	Outputs	Output data bus for the DRP. <sup>(1)</sup>
DADDR[6:0]	Input	Address bus for the DRP. <sup>(1)</sup>
DEN <sup>(2)</sup>	Input	Enable signal for the DRP. <sup>(1)</sup>
DWE <sup>(2)</sup>	Input	Write enable for the DRP. <sup>(1)</sup>
DCLK	Input	Clock input for the DRP. <sup>(1)</sup>
DRDY <sup>(2)</sup>	Output	Data ready signal for the DRP. <sup>(1)</sup>



Table 1-2: XADC Port Descriptions (Cont'd)

Port	I/O	Description
RESET <sup>(2)</sup>	Input	Reset signal for the XADC control logic.
CONVST <sup>(3)</sup>	Input	Convert start input. This input controls the sampling instant on the ADC(s) input and is only used in event mode timing (see <a href="#">Event-Driven Sampling, page 65</a> ). This input comes from the general-purpose interconnect in the FPGA logic.
CONVSTCLK <sup>(3)</sup>	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 mode timing. This input comes from the local clock distribution network in the FPGA logic. Thus, for the best control over the sampling instant (delay and jitter), a global clock input can be used as the CONVST source.
V <sub>P</sub> , V <sub>N</sub>	Input	One dedicated analog input pair. The XADC has one pair of dedicated analog input pins that provide a differential analog input. When designing with the XADC feature but not using the dedicated external channel of V <sub>P</sub> and V <sub>N</sub> , the user should connect both V <sub>P</sub> and V <sub>N</sub> to analog ground.
VAUXP[15:0], VAUXN[15:0]	Inputs	Sixteen auxiliary analog input pairs. In addition to the dedicated differential analog input, the XADC can access 16 differential analog inputs by configuring digital I/O as analog inputs. These inputs can also be enabled pre-configuration via the JTAG port (see <a href="#">DRP JTAG Interface, page 44</a> ).
ALM[0] <sup>(2)</sup>	Output	Temperature sensor alarm output.
ALM[1] <sup>(2)</sup>	Output	V <sub>CCINT</sub> sensor alarm output.
ALM[2] <sup>(2)</sup>	Output	V <sub>CCAUX</sub> sensor alarm output.
ALM[3] <sup>(2)</sup>	Output	V <sub>CCBRAM</sub> sensor alarm output.
ALM[4] <sup>(4)</sup>	Output	V <sub>CCPINT</sub> sensor alarm output.
ALM[5] <sup>(4)</sup>	Output	V <sub>CCPAUX</sub> sensor alarm output.
ALM[6] <sup>(4)</sup>	Output	V <sub>CCO_DDR</sub> sensor alarm output.
ALM[7] <sup>(2)</sup>	Output	Logic OR of bus ALM[6:0]. Can be used to flag the occurrence of any alarm.
OT <sup>(2)</sup>	Output	Over-Temperature alarm output.
MUXADDR[4:0]	Outputs	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 <a href="#">External Multiplexer Mode, page 56</a> ).
CHANNEL[4:0]	Outputs	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.

Table 1-2: XADC Port Descriptions (Cont'd)

Port	I/O	Description
EOC <sup>(2)</sup>	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 (see <a href="#">Chapter 5, XADC Timing</a> ).
EOS <sup>(2)</sup>	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 (see <a href="#">Chapter 5, XADC Timing</a> ).
BUSY <sup>(2)</sup>	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 <sup>(2)</sup>	Output	Indicates that a DRP port lock request has been made by the JTAG interface (see <a href="#">DRP JTAG Interface, page 44</a> ). This signal is also used to indicate that the DRP is ready for access (when Low).
JTAGMODIFIED <sup>(2)</sup>	Output	Used to indicate that a JTAG write to the DRP has occurred.
JTAGBUSY <sup>(2)</sup>	Output	Used to indicate that a JTAG DRP transaction is in progress.

**Notes:**

1. The DRP is the interface between the XADC and FPGA. All XADC registers can be accessed from the FPGA logic using this interface. For more details on the timing for these DRP signals, see [Figure 5-3, page 67](#) and consult UG470, *7 Series FPGAs Configuration User Guide* [Ref 3].
2. Active High signal.
3. Rising edge triggered signal.
4. Only available on Zynq-7000 All Programmable SoC devices.

## XADC Attributes

The block diagram in [Figure 1-1](#) shows the control registers that define the operation of the XADC. The control registers are a set of 32 16-bit registers. As mentioned, these registers can be read and written via the DRP or JTAG ports. It is also possible to initialize the contents of these registers during the configuration of the FPGA. This enables the XADC to start operating in a user-defined mode after configuration is complete. There are 32 attributes associated with the XADC primitive that allow users to initialize these registers. [Table 1-3](#) lists these attributes. The attributes 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.

Table 1-3: XADC Primitive Attributes

Attribute	Name	Control Register Address	Description
INIT_40	Configuration Register 0	40h	XADC configuration registers (see <a href="#">Control Registers, page 39</a> ).
INIT_41	Configuration Register 1	41h	
INIT_42	Configuration Register 2	42h	

Table 1-3: XADC Primitive Attributes (Cont'd)

Attribute	Name	Control Register Address	Description
INIT_43 to INIT_47	Test Registers	43h to 47h	XADC Test registers for factory use only. The default initialization is 0000h.
INIT_48 to INIT_4F	Sequence Registers	48h to 4Fh	Sequence registers used to program the XADC Channel Sequencer function (see <a href="#">Automatic Channel Sequencer, page 49</a> ).
INIT_50 to INIT_5F	Alarm Limit Registers	50h to 5Fh	Alarm threshold registers for the XADC alarm function (see <a href="#">Automatic Alarms, page 59</a> ).

The XADC primitive also has an attribute called SIM\_MONITOR\_FILE 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 (e.g., 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 XADC. For more information see [XADC Software Support, page 74](#).

## Example Instantiation

Instantiating the XADC involves connecting the required I/O (including analog inputs) to the design and optionally initializing the control registers to define the XADC operation after configuration. Alternatively, users can write to the control registers via the DRP after device configuration. The timing diagram for DRP read and write operations is shown in [Figure 5-3](#). Note the read/write operation is not valid or complete until the DRDY signal goes active.

This subsection provides a brief example of an XADC instantiation using Verilog. First, the control registers are initialized, and then the required XADC I/Os are connected to the design. The software correctly ties off unconnected I/Os on the primitive.

This design assumes an external 50 MHz clock is used for DCLK, and the XADC is configured to monitor temperature, supply voltages, and activate alarms if safe limits are exceeded. This example is explained in detail in [XADC Software Support, page 74](#).

```

module My_XADC

XADC #(
    // Initializing the XADC Control Registers
    .INIT_40(16'h9000), // Calibration coefficient averaging disabled
                        // averaging of 16 selected for external channels
    .INIT_41(16'h2ef0), // Continuous Sequencer Mode, Disable unused ALMs,
                        // Enable calibration
    .INIT_42(16'h0400), // Set DCLK divider to 4, ADC = 500Ksps, DCLK = 50MHz
    .INIT_48(16'h4701), // Sequencer channel - enable Temp sensor, VCCINT, VCCAUX,
                        // VCCBRAM, and calibration
    .INIT_49(16'h000f), // Sequencer channel - enable aux analog channels 0 - 3
    .INIT_4A(16'h4700), // Averaging enabled for Temp sensor, VCCINT, VCCAUX,
                        // VCCBRAM
    .INIT_4B(16'h0000), // No averaging on external channels
    .INIT_4C(16'h0000), // Sequencer Bipolar selection
    .INIT_4D(16'h0000), // Sequencer Bipolar selection
    .INIT_4E(16'h0000), // Sequencer Acq time selection

```

```

.INIT_4F(16'h0000), // Sequencer Acq time selection
.INIT_50(16'hb5ed), // Temp upper alarm trigger 85°C
.INIT_51(16'h5999), // Vccint upper alarm limit 1.05V
.INIT_52(16'hA147), // Vccaux upper alarm limit 1.89V
.INIT_53(16'hdddd), // OT upper alarm limit 125°C
.INIT_54(16'ha93a), // Temp lower alarm reset 60°C
.INIT_55(16'h5111), // Vccint lower alarm limit 0.95V
.INIT_56(16'h91Eb), // Vccaux lower alarm limit 1.71V
.INIT_57(16'hae4e), // OT lower alarm reset 70°C
.INIT_58(16'h5999), // VCCBRAM upper alarm limit 1.05V
.INIT_5C(16'h5111), // VCCBRAM lower alarm limit 0.95V
.SIM_MONITOR_FILE("sensor_input.txt")
                        // Analog Stimulus file. Analog input values for simulation
)

XADC_INST (           // Connect up instance IO. See UG480 for port descriptions
.CONVST(GND_BIT), // not used
.CONVSTCLK(GND_BIT), // not used
.DADDR(DADDR_IN[6:0]),
.DCLK(DCLK_IN),
.DEN(DEN_IN),
.DI(DI_IN[15:0]),
.DWE(DWE_IN),
.RESET(RESET_IN),
.VAUXN(aux_channel_n[15:0]),
.VAUXP(aux_channel_p[15:0]),
.ALM(alm_int),
.BUSY(BUSY_OUT),
.CHANNEL(CHANNEL_OUT[4:0]),
.DO(DO_OUT[15:0]),
.DRDY(DRDY_OUT),
.EOC(EOC_OUT),
.EOS(EOS_OUT),
.JTAGBUSY(), // not used
.JTAGLOCKED(), // not used
.JTAGMODIFIED(), // not used
.OT(OT_OUT),
.MUXADDR(), // not used
.VP(VP_IN),
.VN(VN_IN)
);

endmodule

```

## ADC and Sensors

More comprehensive information regarding the operation of the ADCs and on-chip sensors can be found in [Chapter 2, Analog-to-Digital Converter](#). This section provides a brief overview to help users to quickly interpret data read from the status registers and verify the operation of the XADC.

### Analog-to-Digital Converter

The ADCs have a nominal analog input range from 0V to 1V. In unipolar mode (default), the analog inputs of the ADCs produce a full scale code of FFFh (12 bits) when the input is 1V. Thus, an analog input signal of 200 mV in unipolar mode produces and outputs code of [Equation 1-1](#).

$$((0.2/1.0) \times \text{FFFh}) = 819 \text{ or } 333\text{h} \quad \text{Equation 1-1}$$

In bipolar mode, the ADCs use 2's complement coding and produces a full scale code of 7FFh with +0.5V input and 800h with -0.5V input.

## Temperature Sensor

The temperature sensor has a transfer function given by [Equation 1-2](#).

$$Temp (\text{°C}) = \frac{ADC \text{ Code} \times 503.975}{4096} - 273.15 \quad \text{Equation 1-2}$$

For example, ADC Code 2423 (977h) = 25°C.

The temperature sensor result can be found in status register 00h.

## Power Supply Sensors

The XADC power supply sensors have a transfer function that generates a full scale ADC output code of FFFh with a 3V input voltage. This voltage is outside the allowed supply range, but the FPGA supply measurements map into this measurement range. Thus,  $V_{CCINT} = 1V$  generates an output code of  $1/3 \times 4096 = 1365 = 555h$ . The XADC monitors  $V_{CCINT}$ ,  $V_{CCAUX}$ , and  $V_{CCBRAM}$ . The measurement results are stored in status registers 01h, 02h, and 06h, respectively.

### Zynq-7000 All Programmable SoC (AP SoC)

The XADC monitors three additional power supplies on the Zynq-7000 AP\_SoC devices. The supplies are  $V_{CCPINT}$ ,  $V_{CCPAUX}$ , and  $V_{CCO\_DDR}$ . These measurements are stored in status registers 0Dh, 0Eh, and 0Fh, respectively.



# Analog-to-Digital Converter

---

The XADC block contains two 12-bit, 1 MSPS ADCs. These ADCs are available for use with both external analog inputs and on-chip sensors. A number of predefined operating modes are available that cover the most typical use cases for these ADCs. The various operating modes are covered in [Chapter 4, XADC Operating Modes](#). 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 XADC control registers. A detailed description of the control registers is covered in [Chapter 3, XADC Register Interface](#).

## ADC Transfer Functions

The ADCs have 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. Users can optionally configure the external analog input channels to operate in unipolar or bipolar modes (see [Analog Inputs, page 26](#)).

For the ADCs to function as specified, the power supplies and reference options must be configured correctly. The required package ball connections are shown in [Figure 1-2](#). Other important aspects to ensure optimal ADC performance are the PCB layout and external component selection. These issues are covered in [Chapter 6, Application Guidelines](#). It is recommended that this chapter is read before the board design is started.

**Note:** The ADCs always produce a 16-bit conversion result, and the full 16-bit result is stored in the 16-bit status registers. The 12-bit transfer functions shown in this section correspond to the 12 MSBs (most significant or left-most bits) in the 16-bit status registers. The four LSBs can be used to minimize quantization effects or improve resolution through averaging or filtering.

## Unipolar Mode

Figure 2-1 shows the 12-bit unipolar transfer function for the ADCs. The nominal analog input range to the ADCs 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 (FFFh) 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  $1V/2^{12}$  or  $1V/4096 = 244 \mu V$ . The analog input channels are differential in nature and require both the positive ( $V_P$ ) and negative ( $V_N$ ) inputs of the differential input to be driven. More details on the analog inputs and the kinds of inputs signals that can be accommodated are covered in the [Analog Inputs](#) section.

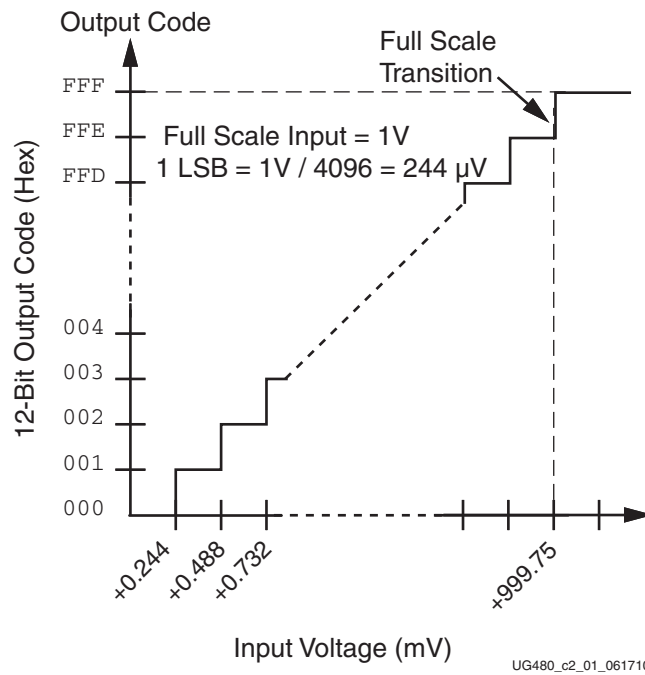


Figure 2-1: Unipolar Transfer Function



## Bipolar Mode

When the external analog input channels of the ADCs 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 is intended to indicate 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  $1V/2^{12}$  or  $1V/4096 = 244 \mu V$ .

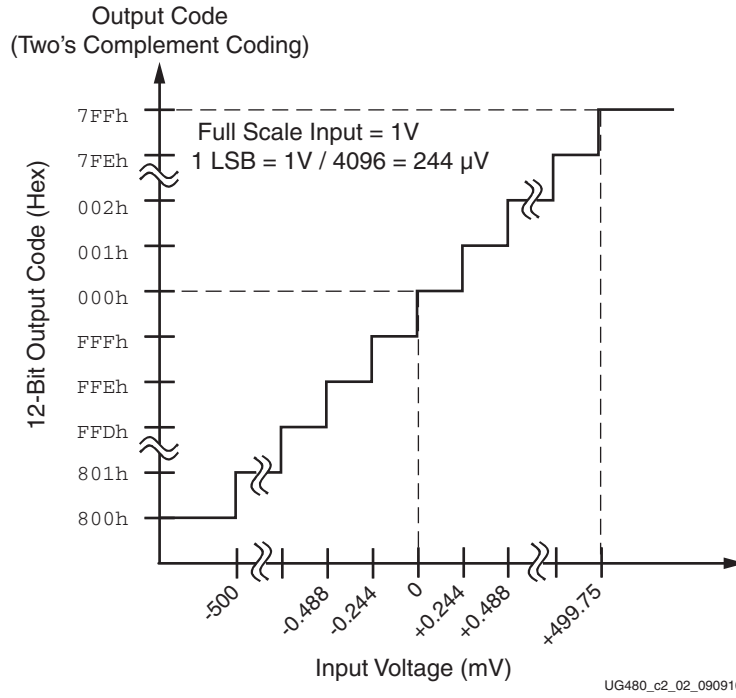
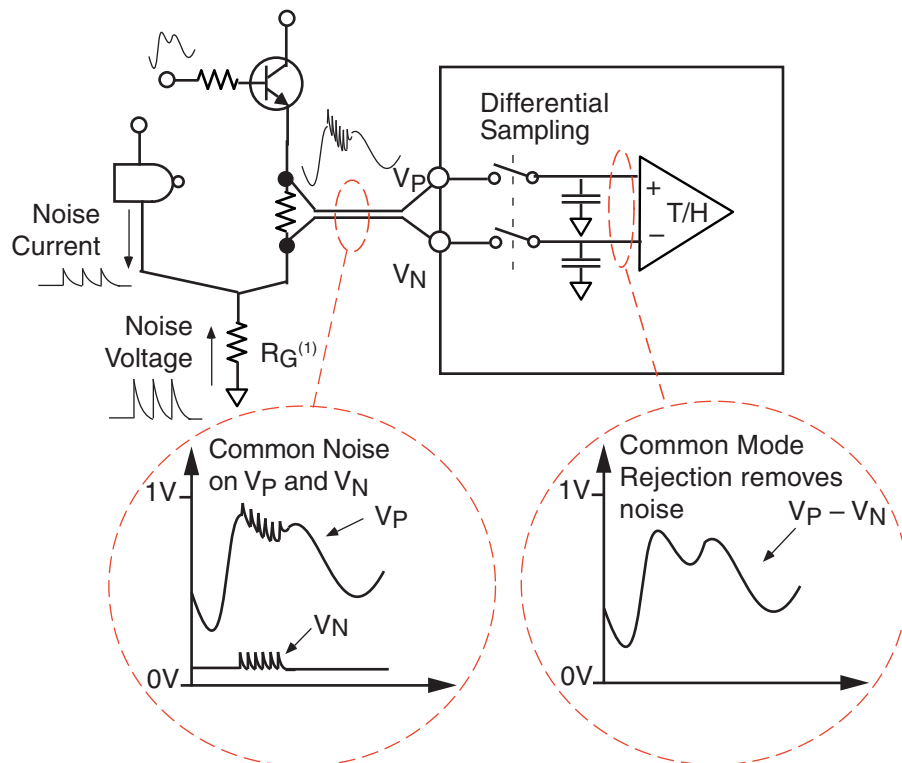


Figure 2-2: **Bipolar Transfer Function**

## 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 ADCs, 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, users need only connect  $V_P$  and  $V_N$  in a differential configuration.



Note 1:  $R_G$  is Common Ground Impedance.

UG480\_c2\_03\_062310

Figure 2-3: Common Mode Noise Rejection

## 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. The auxiliary analog inputs are automatically enabled when the XADC is instantiated in a design, and these inputs are connected on the top level of the design. The auxiliary analog inputs do not require any user-specified constraints or pin locations. They do not require an I/O standard setting to be added to the UCF or in the PlanAhead pinout tool. All configuration is automatic when the analog inputs are connected to the top level of the design. Only those auxiliary inputs connected in a design are enabled as analog inputs. The XADC auxiliary inputs pins are labeled in the *7 Series FPGAs Packaging and Pinout Advance Specification* by appending

`_ADxP_` and `_ADxN_` to the I/O name, where  $x$  is the auxiliary pair number [Ref 2]. 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. 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” section of the respective 7 series FPGAs data sheet.

It is possible to enable any number of 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.

**Note:** Auxiliary channels 6, 7, 13, 14, and 15 are not supported on Kintex-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC device package options. Users should consult the package file for the device.

## Adjusting the Acquisition Settling Time

The maximum conversion rate specified for the ADC is 1 MSPS or a conversion time of 1  $\mu$ s. In continuous sampling mode (see [Chapter 5, XADC Timing](#)), 26 ADCCLK cycles are required to acquire an analog signal and perform a conversion. This implies a maximum ADCCLK frequency of 26 MHz. If the ACQ (see [Control Registers, page 39](#)) bit has not been set, four ADCCLKs or 150 ns is allowed for the final stages of the acquisition. This “settling” time ensures that the analog input voltage is acquired to a 12-bit accuracy. The settling time can be increased by reducing the ADCCLK frequency or setting the ACQ bit. In the latter case, the settling time is increased to 380 ns (10 ADCCLK cycles), and the conversion rate would be reduced to 812 kSPS for the same ADCCLK frequency. In event timing mode (see [Chapter 5, XADC Timing](#)), the user initiates the conversion cycle by using CONVST or CONVSTCLK, allowing more control over the acquisition time, if required.

## Analog Input Description

[Figure 2-4](#) 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 ( $\pm 0.5$  LSBs at 12 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.

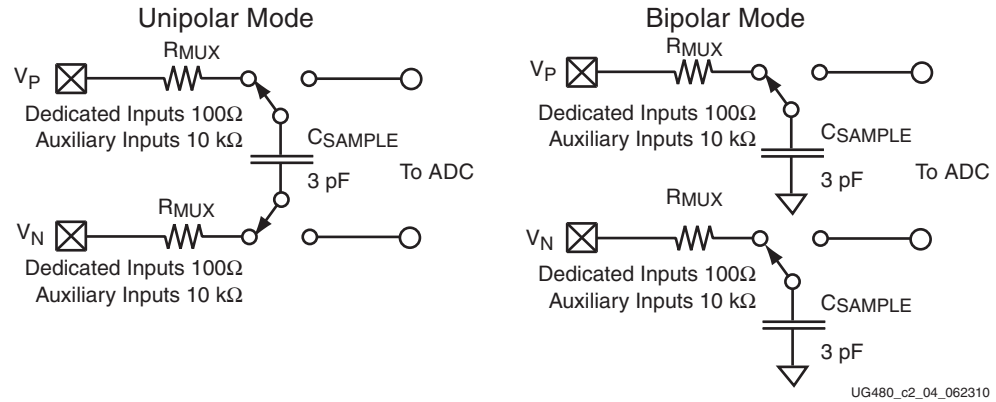


Figure 2-4: Equivalent Analog Input Circuits

The required 12-bit acquisition time (assuming no additional external or source resistance) in bipolar mode for example is approximated by Equation 2-1.

$$t_{ACQ} = 9 \times R_{MUX} \times C_{SAMPLE} \quad \text{Equation 2-1}$$

The required 12-bit acquisition time in unipolar mode is approximated by Equation 2-2.

$$t_{ACQ} = 9 \times (R_{MUX} + R_{MUX}) \times C_{SAMPLE} \quad \text{Equation 2-2}$$

For the dedicated channel ( $V_P/V_N$ ), the minimum acquisition time (bipolar mode) required is given by Equation 2-3.

$$t_{ACQ} = 9 \times 100 \times 3 \times 10^{-12} = 3 \text{ ns} \quad \text{Equation 2-3}$$

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 k $\Omega$ . Equation 2-4 shows the minimum acquisition time.

$$t_{ACQ} = 9 \times 10 \times 10^3 \times 3 \times 10^{-12} = 300 \text{ ns} \quad \text{Equation 2-4}$$

Note any additional external resistance such as the anti-alias filter or resistor divider for example increases the acquisition time requirement due to the increased  $R_{MUX}$  value in Equation 2-1. To calculate the new acquisition time, simply 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 XAPP795, *Guidelines for Driving Analog Inputs of the XADC*.

## Unipolar Input Signals

When measuring unipolar analog input signals, the ADCs must operate in a unipolar input mode. This mode is selected by writing to Configuration Register 0 (see [Control Registers, page 39](#)). 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 2-5 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 2-5 shows

the maximum signal levels on  $V_N$  and  $V_P$  in unipolar mode, measured with respect to analog ground (GNDADC package ball).

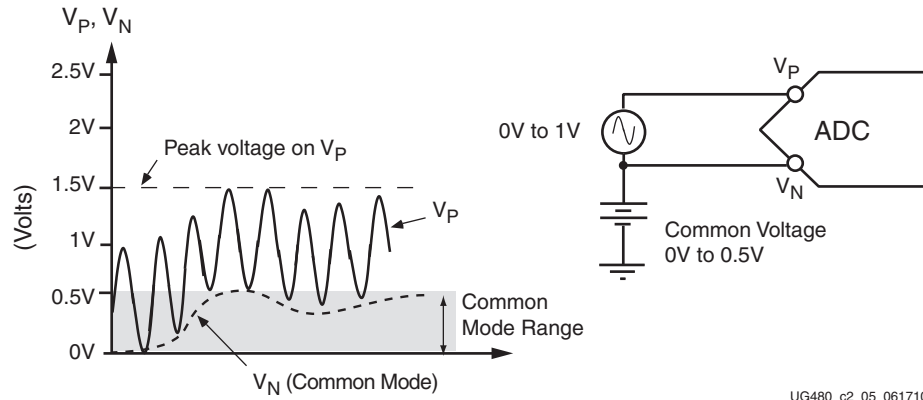


Figure 2-5: Unipolar Input Signal Range

### 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  $\pm 0.5V$ . The common mode or reference voltage should not exceed 0.5V in this case (see [Figure 2-6](#)).

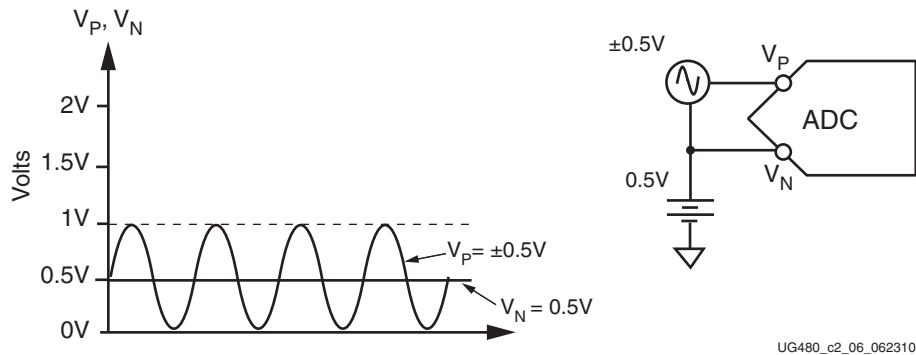
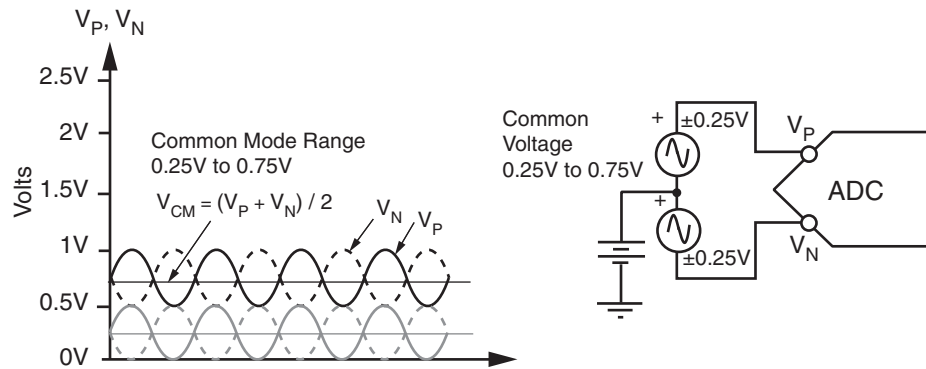


Figure 2-6: Bipolar Input Signal Range

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 2-7](#)). The maximum differential input ( $V_P - V_N$ ) is  $\pm 0.5V$ . With maximum differential input voltages of  $\pm 0.5V$  and assuming balanced inputs on  $V_N$  and  $V_P$ , the common mode voltage must lie in the range 0.25V to 0.75V.



UG480\_c2\_07\_061710

Figure 2-7: Differential Input Signal Range

## Temperature Sensor

The XADC contains a temperature sensor that produces a voltage output proportional to the die temperature. Equation 2-5 shows the output voltage of the temperature sensor.

$$\text{Voltage} = 10 \times \frac{\kappa T}{q} \times \ln(10) \quad \text{Equation 2-5}$$

Where:

$$\kappa = \text{Boltzmann's constant} = 1.38 \times 10^{-23} \text{ J/K}$$

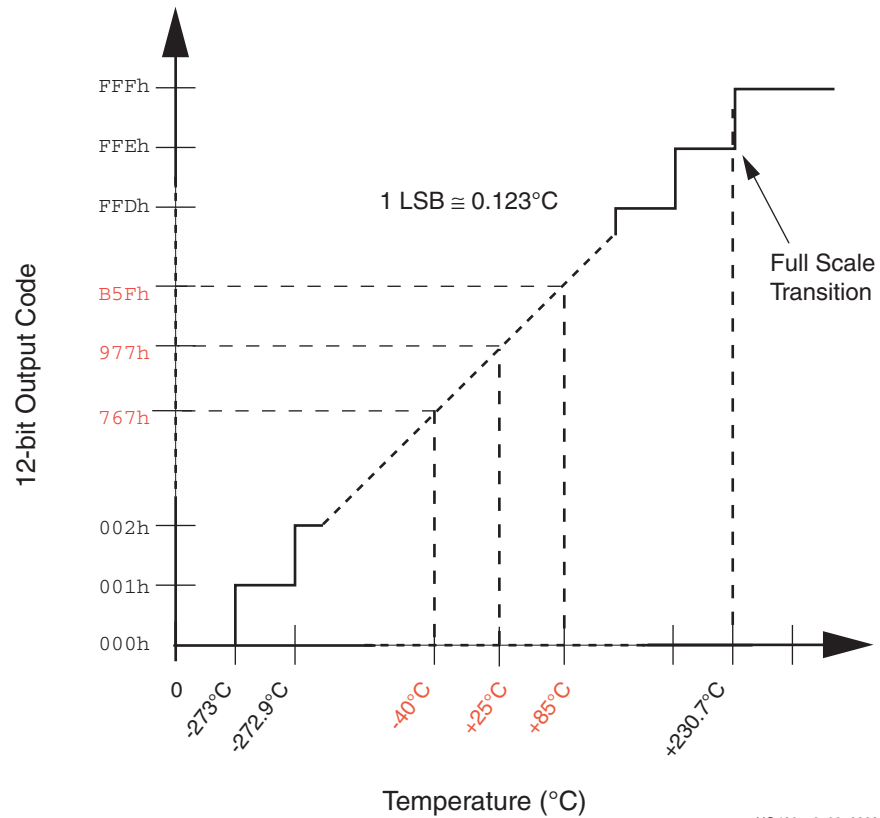
$$T = \text{Temperature K (Kelvin)} = ^\circ\text{C} + 273.15$$

$$q = \text{Charge on an electron} = 1.6 \times 10^{-19} \text{ C}$$

The output voltage of this sensor is digitized by the ADC to produce a 12-bit digital output code (ADC code). Figure 2-8 illustrates the digital output transfer function for this temperature sensor.

For simplification, the temperature sensor plus the ADC transfer function is rewritten as shown in Equation 2-6.

$$\text{Temperature } (^\circ\text{C}) = \frac{\text{ADC Code} \times 503.975}{4096} - 273.15 \quad \text{Equation 2-6}$$



UG480\_c2\_08\_090910

Figure 2-8: Temperature Sensor Transfer Function

The temperature measurement result is stored in the status registers at DRP address 00h. Monitoring FPGA on-chip temperature avoids functional and irreversible failures by ensuring critical operating temperatures are not exceeded.

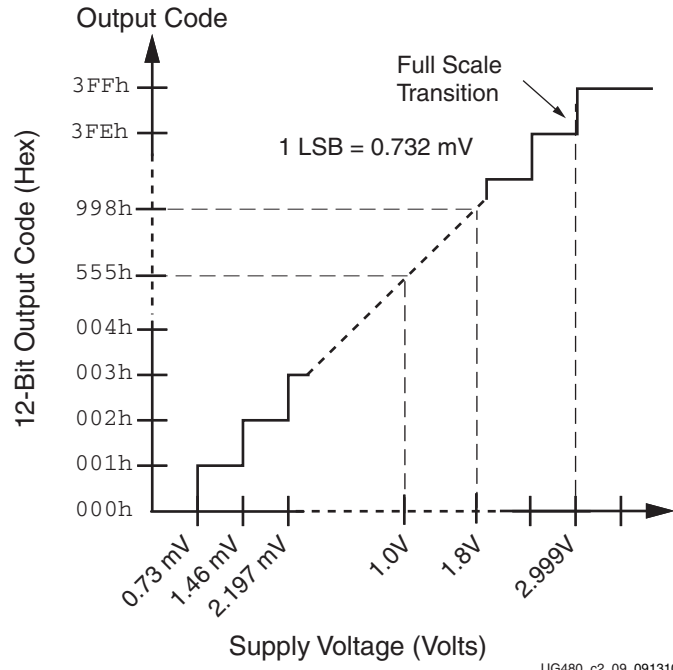
## Power Supply Sensor

The XADC also includes on-chip sensors that allow a user to monitor the FPGA power-supply voltages using the ADC. The sensors sample and attenuate (by a factor of three) the power supply voltages  $V_{CCINT}$ ,  $V_{CCAUX}$ , and  $V_{CCBRAM}$  on the package power supply balls. On Zynq-7000 AP SoC devices, the  $V_{CCPINT}$ ,  $V_{CCPAUX}$ , and  $V_{CCO\_DDR}$  supplies are also monitored.

Figure 2-9 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  $V_{CCAUX} + 5\%$  with a resolution of approximately 0.73 mV. The transfer function for the supply sensor is shown in Equation 2-7.

$$Voltage = \frac{ADC\ Code}{4096} \times 3V \quad \text{Equation 2-7}$$

The power-supply measurement results for  $V_{CCINT}$ ,  $V_{CCAUX}$ , and  $V_{CCBRAM}$  are stored in the status registers at DRP addresses 01h, 02h, and 06h, respectively. On Zynq-7000 AP SoC devices, the measurements for  $V_{CCPINT}$ ,  $V_{CCPAUX}$ , and  $V_{CCO\_DDR}$  are stored in status registers 0Dh, 0Eh, and 0Fh, respectively.



UG480\_c2\_09\_091310

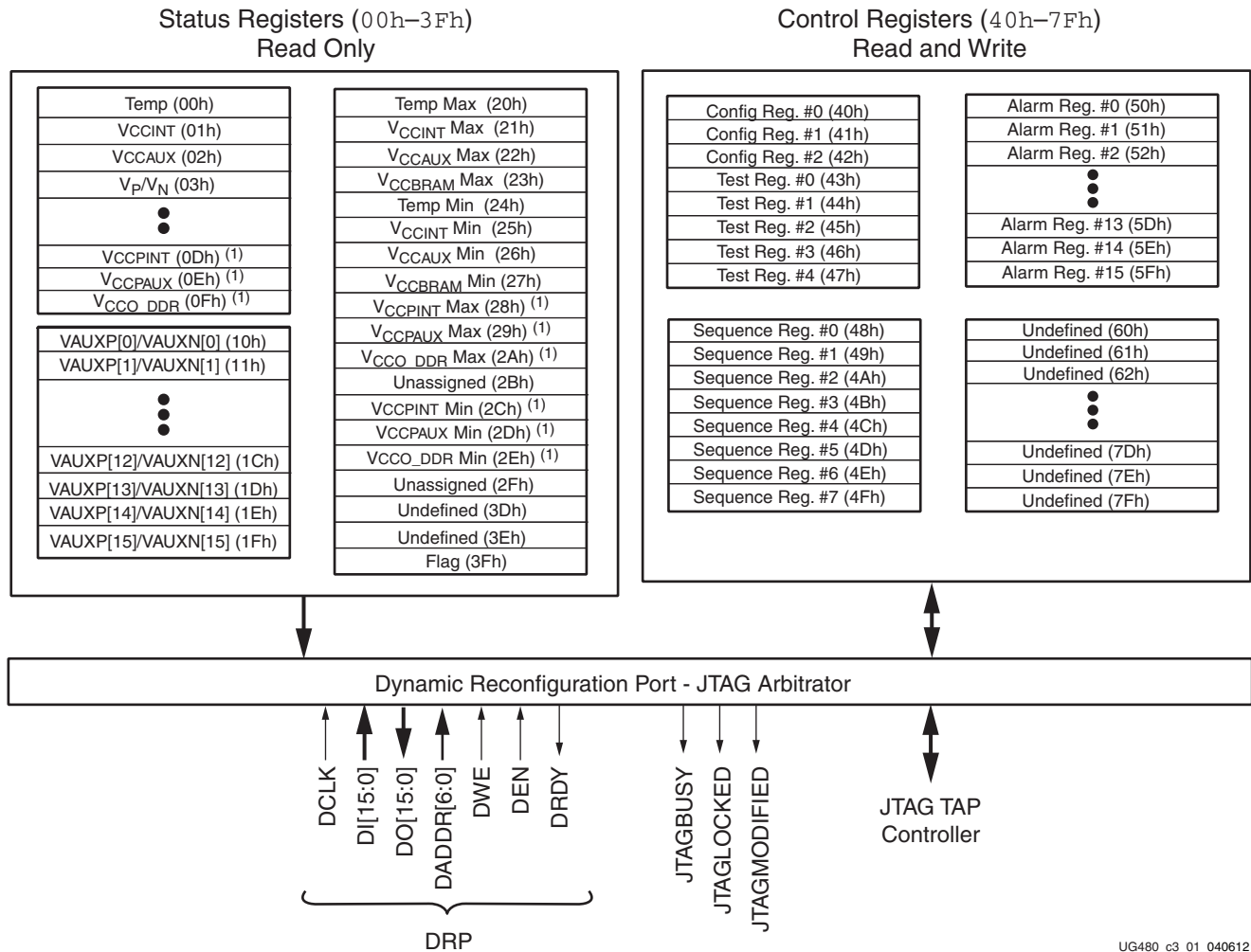
Figure 2-9: Ideal Power Supply Transfer Function



## *XADC Register Interface*

---

Figure 3-1 illustrates the XADC register interface. All registers in the register interface are accessible through the Dynamic Reconfiguration Port (DRP). The DRP can be accessed via a fabric port or the JTAG TAP. Access is governed by an arbitrator (see [DRP Arbitration](#), page 46). The DRP allows the user to access up to 128 16-bit registers (DADDR[6:0] = 00h to 7Fh). The first 64 access locations (DADDR[6:0] = 00h to 3Fh) are read-only and contain the ADC measurement data. These registers are known as status registers. The control registers are located at addresses 40h to 7Fh and are readable or writable via the DRP. The DRP timing is shown in [Figure 5-3](#).



UG480\_c3\_01\_040612

Figure 3-1: XADC Register Interface

Notes relevant to Figure 3-1:

1. Zynq-7000 family of AP SoC devices only.

For a detailed description of the DRP timing, see [Dynamic Reconfiguration Port \(DRP\) Timing, page 66](#). For more information on the JTAG DRP interface, see [DRP JTAG Interface, page 44](#).

## Status Registers

The first 64 address locations (DADDR[6:0] = 00h to 3Fh) contain the read-only status registers. The status registers contain the results of an analog-to-digital conversion of the on-chip sensors and external analog channels. All sensors and external analog-input channels have a unique channel address (see [Table 3-7, page 41](#)). The measurement result from each channel is stored in a status register with the same address on the DRP.

For example, the result from an analog-to-digital conversion on ADC multiplexer channel 0 (temperature sensor) is stored in the status register at address 00h. The result from ADC multiplexer channel 1 (V<sub>CCINT</sub>) is stored at address 01h.

The status registers also store the maximum and minimum measurements recorded for the on-chip sensors from the device power-up or the last user reset of the XADC. [Table 3-1](#) defines the status registers.

**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 12 MSBs correspond to the temperature sensor transfer function shown in <a href="#">Figure 2-8, page 31</a> .
V <sub>CCINT</sub>	01h	The result of the on-chip V <sub>CCINT</sub> supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register. The 12 MSBs correspond to the supply sensor transfer function shown in <a href="#">Figure 2-9, page 32</a> .
V <sub>CCAUX</sub>	02h	The result of the on-chip V <sub>CCAUX</sub> data supply monitor measurement is stored at this location. The data is MSB justified in the 16 bit register. The 12 MSBs correspond to the supply sensor transfer function shown in <a href="#">Figure 2-9</a> .
V <sub>P</sub> /V <sub>N</sub>	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 12 MSBs correspond to the transfer function shown in <a href="#">Figure 2-5, page 29</a> or <a href="#">Figure 2-6, page 29</a> depending on analog input mode settings.
V <sub>REFP</sub>	04h	The result of a conversion on the reference input V <sub>REFP</sub> is stored in this register. The 12 MSBs correspond to the ADC transfer function shown in <a href="#">Figure 2-9</a> . The data is MSB justified in the 16-bit register. The supply sensor is used when measuring V <sub>REFP</sub> .
V <sub>REFN</sub>	05h	The result of a conversion on the reference input V <sub>REFN</sub> is stored in this register. This channel is measured in bipolar mode with a 2's complement output coding as shown in <a href="#">Figure 2-2, page 25</a> . By measuring in bipolar mode, small positive and negative offset around 0V (V <sub>REFN</sub> ) can be measured. The supply sensor is also used to measure V <sub>REFN</sub> , thus 1 LSB = 3V/4096. The data is MSB justified in the 16-bit register.
V <sub>CCBRAM</sub>	06h	The result of the on-chip V <sub>CCBRAM</sub> supply monitor measurement is stored at this location. The data is MSB justified in the 16-bit register. The 12 MSBs correspond to the supply sensor transfer function shown in <a href="#">Figure 2-9</a> .
Undefined	07h	This location is unused and contains invalid data.
Supply A Offset	08h	The calibration coefficient for the supply sensor offset using ADC A is stored at this location.
ADC A Offset	09h	The calibration coefficient for the ADC A offset is stored at this location.
ADC A Gain	0Ah	The calibration coefficient for the ADC A gain error is stored at this location.

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

Name	Address	Description
Undefined	0Bh to 0Ch	These locations are unused and contain invalid data.
$V_{CCPINT}^{(1)}$	0Dh	The result of a conversion on the PS supply, $V_{CCPINT}$ is stored in this register. The 12 MSBs correspond to the ADC transfer function shown in <a href="#">Figure 2-9, page 32</a> . The data is MSB justified in the 16-bit register. The supply sensor is used when measuring $V_{CCPINT}$ .
$V_{CCPAUX}^{(1)}$	0Eh	The result of a conversion on the PS supply, $V_{CCPAUX}$ is stored in this register. The 12 MSBs correspond to the ADC transfer function shown in <a href="#">Figure 2-9, page 32</a> . The data is MSB justified in the 16-bit register. The supply sensor is used when measuring $V_{CCPAUX}$ .
$V_{CCO\_DDR}^{(1)}$	0Fh	The result of a conversion on the PS supply, $V_{CCO\_DDR}$ is stored in this register. The 12 MSBs correspond to the ADC transfer function shown in <a href="#">Figure 2-9, page 32</a> . The data is MSB justified in the 16-bit register. The supply sensor is used when measuring $V_{CCO\_DDR}$ .
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 12 MSBs correspond to the transfer function shown in <a href="#">Figure 2-1, page 24</a> or <a href="#">Figure 2-2, page 25</a> depending on analog input mode settings.
Max Temp	20h	Maximum temperature measurement recorded since power-up or the last XADC reset.
Max $V_{CCINT}$	21h	Maximum $V_{CCINT}$ measurement recorded since power-up or the last XADC reset.
Max $V_{CCAUX}$	22h	Maximum $V_{CCAUX}$ measurement recorded since power-up or the last XADC reset.
Max $V_{CCBRAM}$	23h	Maximum $V_{CCBRAM}$ measurement recorded since power-up or the last XADC reset.
Min Temp	24h	Minimum temperature measurement recorded since power-up or the last XADC reset.
Min $V_{CCINT}$	25h	Minimum $V_{CCINT}$ measurement recorded since power-up or the last XADC reset.
Min $V_{CCAUX}$	26h	Minimum $V_{CCAUX}$ measurement recorded since power-up or the last XADC reset.
Min $V_{CCBRAM}$	27h	Minimum $V_{CCBRAM}$ measurement recorded since power-up or the last XADC reset.
$V_{CCPINT}^{(1)}$ Max	28h	Maximum $V_{CCPINT}$ measurement recorded since power-up or the last XADC reset.
$V_{CCPAUX}^{(1)}$ Max	29h	Maximum $V_{CCPAUX}$ measurement recorded since power-up or the last XADC reset.

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

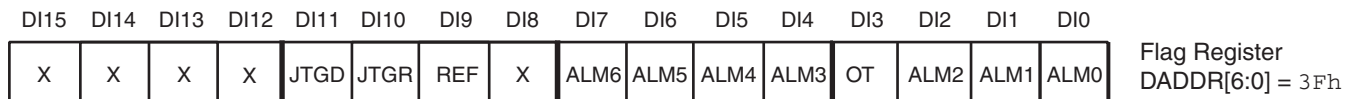
Name	Address	Description
V <sub>CCO_DDR</sub> <sup>(1)</sup> Max	2Ah	Maximum V <sub>CCO_DDR</sub> measurement recorded since power-up or the last XADC reset.
Unassigned	2Bh	
V <sub>CCPINT</sub> <sup>(1)</sup> Min	2Ch	Minimum V <sub>CCPINT</sub> measurement recorded since power-up or the last XADC reset.
V <sub>CCPAUX</sub> <sup>(1)</sup> Min	2Dh	Minimum V <sub>CCPAUX</sub> measurement recorded since power-up or the last XADC reset.
V <sub>CCO_DDR</sub> <sup>(1)</sup> Min	2Eh	Minimum V <sub>CCO_DDR</sub> measurement recorded since power-up or the last XADC reset.
Unassigned	2Fh	
Supply B Offset	30h	The calibration coefficient for the supply sensor offset using ADC B is stored at this location.
ADC B Offset	31h	The calibration coefficient for the ADC B offset is stored at this location.
ADC B Gain	32h	The calibration coefficient for the ADC B gain error is stored at this location.
Undefined	33h to 3Eh	These locations are unused and contain invalid data.
Flag	3Fh	This register contains general status information (see <a href="#">Flag Register</a> ).

**Notes:**

1. These channels are only available in Zynq-7000 AP SoC devices.

## Flag Register

The Flag Register is shown in [Figure 3-2](#). The bit definitions are described in [Table 3-2](#).



UG480\_c3\_02\_032212

Figure 3-2: Flag Registers

Table 3-2: Flag Register Bit Definitions

Bit	Name	Description
DI7 to DI0	ALM6 to ALM0	These bits reflect the status of the alarm output ALM[6:0].
DI3	OT	This bit reflects the status of the Over Temperature logic output.
DI9	REF	When this bit is a logic 1, the ADC is using the internal voltage reference. When this bit is a logic 0, the external reference is being used.

Table 3-2: Flag Register Bit Definitions (Cont'd)

Bit	Name	Description
DI10	JTGR	A logic 1 indicates that the <b>JTAG_XADC</b> BitGen option has been used to restrict JTAG access to read only. See <a href="#">DRP JTAG Interface</a> for more information.
DI11	JTGD	A logic 1 indicates that the <b>JTAG_XADC</b> BitGen option has been used to disable all JTAG access. See <a href="#">DRP JTAG Interface</a> for more information.

## XADC Calibration Coefficients

The XADC can digitally calibrate out any offset and gain errors in the ADCs and power supply sensor. By connecting known voltages (i.e.,  $V_{REFN}$  and  $V_{REFP}$ ) to the ADCs and the supply sensor, the offset and gain errors can be calculated and correction coefficients generated. These calibration coefficients are stored in status registers 08h to 0Ah for ADC A and 30h to 32h for ADC B (see [Table 3-1, page 35](#)).

The XADC 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 XADC *default* operating mode also includes this channel and calibration coefficients are also applied automatically. 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 3-5, page 41](#)).

BUSY transitions High for the duration of the entire calibration sequence (conversion on channel 8). This calibration sequence is six times longer than a regular conversion on a sensor channel as offset and gain are measured for both ADCs and the power supply sensor.

### Calibration Coefficients Definition

As mentioned above, 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 it is not possible to modify the contents via 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 12-bit, 2's complement number and is expressed in LSBs. Similar to other status registers, the 12-bit values are MSB justified in the registers. For example, if the ADC has an offset of +10 LSBs (approximately  $10 \times 250 \mu\text{V} = 2.5 \text{ mV}$ ), the offset coefficient records -10 LSBs or FF6h, and status register 08h records 1111, 1111, 1100 XXXXb. For the supply sensor, the LSB size is approximately 750  $\mu\text{V}$ , thus a +10 LSB offset is equivalent to 7.5 mV of offset in the supply measurement.

#### Gain Coefficients

The ADC gain calibration coefficient stores the correction factor for any gain error in the ADCs. The correction factor is stored in the six LSBs of registers 0Ah and 32h. These six bits store both sign and magnitude information for the gain correction factor. If the sixth bit is a logic 1, the correction factor is positive. If it is 0, the correction factor is negative. The next five bits store the magnitude of the gain correction factor. Each bit is equivalent to 0.1%.

For example, if the ADC A 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 sixth bit is set to zero. The remaining magnitude bits record 1%, where 1% = 10 x 0.1% and 10 = 1010 binary. The status register 0Ah records 0000 0000 0000 1010. With five bits assigned to the magnitude, the calibration can correct errors in the range  $\pm 0.1\% \times (2^5 - 1)$ , or  $\pm 3.1\%$ .

## Control Registers

The XADC has 32 control registers that are located at DRP addresses 40h to 5Fh (see [Table 3-3](#)). These registers are used to configure the XADC operation. All XADC functionality is controlled through these registers.

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

**Table 3-3: XADC Control Registers**

Name	Address	Software Attribute	Description
Configuration Register 0	40h	INIT_40	These are XADC configuration registers (see <a href="#">Configuration Registers (40h to 42h)</a> ).
Configuration Register 1	41h	INIT_41	
Configuration Register 2	42h	INIT_42	
Test Registers 0 to 4	43h to 47h	INIT_43 to INIT_47	These are test registers. The default initialization is 0000h. These registers are used for factory test and should be left at the default initialization.
Sequence Registers	48h to 4Fh	INIT_48 to INIT_4F	These registers are used to program the channel sequencer function (see <a href="#">Chapter 4, XADC Operating Modes</a> ).
Alarm Registers	50h to 5Fh	INIT_50 to INIT_5F	These are the alarm threshold registers for the XADC alarm function (see <a href="#">Automatic Alarms, page 59</a> ).

### Configuration Registers (40h to 42h)

The first three registers in the control register block configure the XADC operating modes. These registers are known as XADC configuration registers. Their bit definitions are illustrated in [Figure 3-3](#).

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

The configuration registers can be modified through the DRP after the FPGA has been configured. For example, a soft microprocessor or state machine can be used to alter the contents of the XADC control registers at any time during normal operation. [Table 3-4](#) through [Table 3-6](#) define the bits for the three configuration registers.

DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Config Reg #0 DADDR[6:0] = 40h
CAVG	0	AVG1	AVG0	MUX	B $\bar{U}$	E $\bar{C}$	ACQ	0	0	0	CH4	CH3	CH2	CH1	CH0	
DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Config Reg #1 DADDR[6:0] = 41h
SEQ3	SEQ2	SEQ1	SEQ0	ALM6 (Note 1)	ALM5 (Note 1)	ALM4 (Note 1)	ALM3	CAL3	CAL2	CAL1	CAL0	ALM2	ALM1	ALM0	OT	
DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	Config Reg #2 DADDR[6:0] = 42h
CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	0	0	PD1	PD0	0	0	0	0	

UG480\_c3\_03\_101112

Figure 3-3: Configuration Registers Bit Definitions

Notes relevant to [Figure 3-3](#):

1. Zynq-7000 AP SoC devices only.

Table 3-4: Configuration Register 0 Bit Definitions

Bit	Name	Description
DI0 to DI3	CH0 to CH4	When operating in single channel mode or external multiplexer mode, these bits are used to select the ADC input channel. See <a href="#">Table 3-7</a> for the channel assignments.
DI8	ACQ	This bit is used to increase the settling time available on external analog inputs in continuous sampling mode by six ADCCLK cycles (see <a href="#">Chapter 2, Analog-to-Digital Converter</a> and <a href="#">Chapter 5, XADC Timing</a> ). The acquisition time is increased by setting this bit to logic 1.
DI9	E $\bar{C}$	This bit is used to select either continuous or event-driven sampling mode for the ADC (see <a href="#">Chapter 5, XADC Timing</a> ). A logic 1 places the ADC in event-driven sampling mode and a logic 0 places the ADC in continuous sampling mode.
DI10	B $\bar{U}$	This bit is used in single channel mode to select either unipolar or bipolar operating mode for the ADC analog inputs (see <a href="#">Analog Inputs, page 26</a> ). A logic 1 places the ADC in bipolar mode and a logic 0 places the ADC in unipolar mode.
DI11	MUX	This bit should be set to a logic 1 to enable external multiplexer mode. See <a href="#">External Multiplexer Mode</a> for more information.
DI12, DI13	AVG0, AVG1	These bits are used to set the amount of sample averaging on selected channels in both single channel and sequence modes (see <a href="#">Table 3-8</a> and <a href="#">Chapter 4, XADC Operating Modes</a> ).
DI15	CAVG	This bit is used to disable averaging for the calculation of the calibration coefficients. Averaging is enabled by default (logic 0). To disable averaging, set this bit to logic 1. Averaging is fixed at 16 samples.



Table 3-5: Configuration Register 1 Bit Definitions

Bit	Name	Description
DI0	OT	This bit is used to disable the over-temperature signal. The alarm is disabled by setting this bit to logic 1.
DI1 to DI3, DI8	ALM0 to ALM3	These bits are used to disable individual alarm outputs for temperature, $V_{CCINT}$ , $V_{CCAUX}$ , and $V_{CCBRAM}$ , respectively. A logic 1 disables an alarm output.
DI9 to DI11	ALM4 to ALM6	These bits are used to disable individual alarm outputs for temperature, $V_{CCPINT}$ , $V_{CCPAUX}$ , and $V_{CCO\_DDRT}$ , respectively. A logic 1 disables an alarm output.
DI4 to DI7	CAL0 to CAL3	These bits enable 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. For bit assignments, see Table 3-10.
DI12 to DI15	SEQ0 to SEQ3	These bits enable the channel-sequencer function. For the bit assignments, see Table 3-9. See Chapter 4, XADC Operating Modes, for more information.

Table 3-6: Configuration Register 2 Bit Definitions

Bit	Name	Description
DI4, DI5	PD0, PD1	Power-down bits for the XADC. The entire XADC block can be powered down by setting PD1 = PD0 = 1. ADC B can also be powered down by setting PD1 = 1 and PD0 = 0. See Table 3-11.
DI8 to DI15	CD0 to CD7	These bits select the division ratio between the DRP clock (DCLK) and the lower frequency ADC clock (ADCCLK) used for the ADC (see Chapter 5, XADC Timing). For bit assignments, see Table 3-12.

Table 3-7: ADC Channel Select

ADC Channel	CH4	CH3	CH2	CH1	CH0	Description
0	0	0	0	0	0	On-chip temperature
1	0	0	0	0	1	$V_{CCINT}$
2	0	0	0	1	0	$V_{CCAUX}$
3	0	0	0	1	1	$V_P$ , $V_N$ - Dedicated analog inputs
4	0	0	1	0	0	$V_{REFP}$ (1.25V) <sup>(1)</sup>
5	0	0	1	0	1	$V_{REFN}$ (0V) <sup>(1)</sup>
6	0	0	1	1	0	$V_{CCBRAM}$
7	0	0	1	1	1	Invalid channel selection
8	0	1	0	0	0	Carry out an XADC calibration
9-12	...	...	...	...	...	Invalid channel selection

Table 3-7: ADC Channel Select (Cont'd)

ADC Channel	CH4	CH3	CH2	CH1	CH0	Description
13	0	1	1	0	1	V <sub>CCPINT</sub> <sup>(3)</sup>
14	0	1	1	1	0	V <sub>CCPAUX</sub> <sup>(3)</sup>
15	0	1	1	1	1	V <sub>CCO_DDR</sub> <sup>(3)</sup>
16	1	0	0	0	0	VAUXP[0], VAUXN[0] - Auxiliary channel 0
17	1	0	0	0	1	VAUXP[1], VAUXN[1] - Auxiliary channel 1
18–31	...	...	...	...	...	VAUXP[2:15], VAUXN[2:15] - Auxiliary channels 2 to 15 <sup>(2)</sup>

**Notes:**

1. These channel selection options are used for XADC self-check and calibration operations. When these channels are selected, the supply sensor is connected to V<sub>REFP</sub> or V<sub>REFN</sub>.
2. Auxiliary channels 6, 7, 13, 14, and 15 are not supported on Kintex™-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC package options. Users should consult the package file for the device.
3. These channels are only supported in Zynq-7000 AP SoC devices.

Table 3-8: 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 3-9: 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)
0	1	X	X	Simultaneous Sampling Mode
1	0	X	X	Independent ADC Mode
1	1	X	X	Default Mode

Table 3-10: Calibration Enables

Name	Description
CAL0	ADCs offset correction enable
CAL1	ADCs offset and gain correction enable

Table 3-10: Calibration Enables (Cont'd)

Name	Description
CAL2	Supply Sensor offset correction enable
CAL3	Supply Sensor offset and gain correction enable

Table 3-11: Power Down Selection

PD1	PD0	Description
0	0	Default. All XADC blocks powered up
0	1	Not valid - do not select
1	0	ADC B powered down
1	1	XADC powered down

Table 3-12: DCLK Division Selection<sup>(1)</sup>

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.

## Test Registers (43h to 47h)

These registers, intended for factory test purposes only, have a default status of zero. The user must not write to these registers.

## Channel Sequencer Registers (48h to 4Fh)

These registers are used to program the channel sequencer functionality. For more information see [Automatic Channel Sequencer, page 49](#).

## Alarm Registers (50h to 5Fh)

These registers are used to program the alarm thresholds for the automatic alarms on the internally monitored channels, temperature, V<sub>CCINT</sub>, V<sub>CCAUX</sub>, and V<sub>CCBRAM</sub>. For Zynq-7000 AP SoC devices, the alarm thresholds for V<sub>CCPINT</sub>, V<sub>CCPAUX</sub>, and V<sub>CCO\_DDR</sub> are also set using these registers. For more information, see [Automatic Alarms, page 59](#).

## DRP JTAG Interface

The XADC uses a full JTAG interface extension to the DRP interface. This allows read/write access to the XADC 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 XADC\_DRP, added to 7 series FPGAs, allows access to the DRP through the JTAG TAP. All XADC JTAG instructions are 32 bits wide. For more information on the 7 series FPGA boundary-scan instructions and usage, see UG470, *7 Series FPGAs Configuration User Guide* [Ref 3]. Read and Write operations using the XADC JTAG DRP interface are described in the next sections. Users unfamiliar with basic JTAG functionality should become familiar with the JTAG standard (IEEE 1149.1) before proceeding.

**Note:** JTAG access can be limited to read only or completely disabled by using the **JTAG\_XADC: [Enable ! Disable ! Status\_only]** BitGen option. If the dedicated PS to XADC interface is used in the Zynq-7000 AP SoC devices, this external JTAG access is also disabled (see UG585, *Zynq+7000 All Programmable SoC Technical Reference Manual* for more details [Ref 4]).

### XADC DRP JTAG Write Operation

Figure 3-4 shows a timing diagram for a write operation to the XADC DRP through the JTAG TAP. The DRP is accessed through the XADC Data Register (XADC DR). Before the XADC DR is accessed, the instruction register (IR) must first be loaded with the XADC instruction. The controller is placed in the IR-scan mode, and the XADC instruction is shifted to the IR.

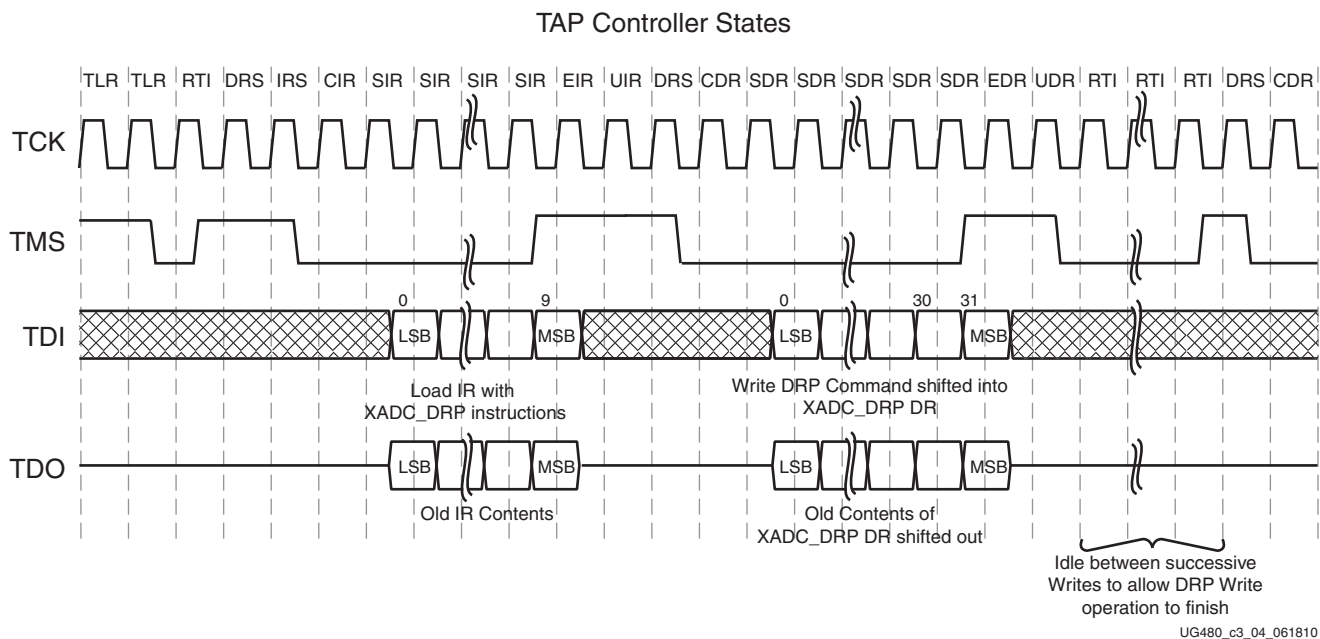


Figure 3-4: XADC JTAG DRP Write

After the XADC instruction is loaded, all data register (DR)-scan operations are carried out on the XADC DR. When the data shifted into XADC DR is a JTAG DRP Write command, the XADC DRP arbitrator carries out a DRP write. The format of this write command is described in [JTAG DRP Commands](#). The XADC DR contents are transferred to the XADC DRP arbitrator (see [DRP Arbitration](#)) during the Update-DR. After the Update-DR

state, the arbitrator manages the new data transfer to the XADC DRP register. This takes up to six DRP Clocks (DCLK) cycles if a DRP access from the FPGA logic is already in progress.

During the Capture-DR phase (just before data is shifted into the XADC 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 [XADC 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 XADC DR are taking place, it might be necessary to idle the TAP controller for several TCK cycles before advancing to the next write operation (see [Figure 3-4](#)). The idle cycles allow the arbitrator to complete the write operation to the XADC DRP register. If DCLK is running approximately 6 x TCK, these idle states are not necessary. However, inserting 10 or so idle states ensures all transfers are inherently safe.

### XADC DRP JTAG Read Operation

[Figure 3-5](#) shows the timing for an XADC DR Read operation. The IR should contain the DR-scan operation (XADC\_DRP instruction). A JTAG read from the XADC DRP is a two-step operation.

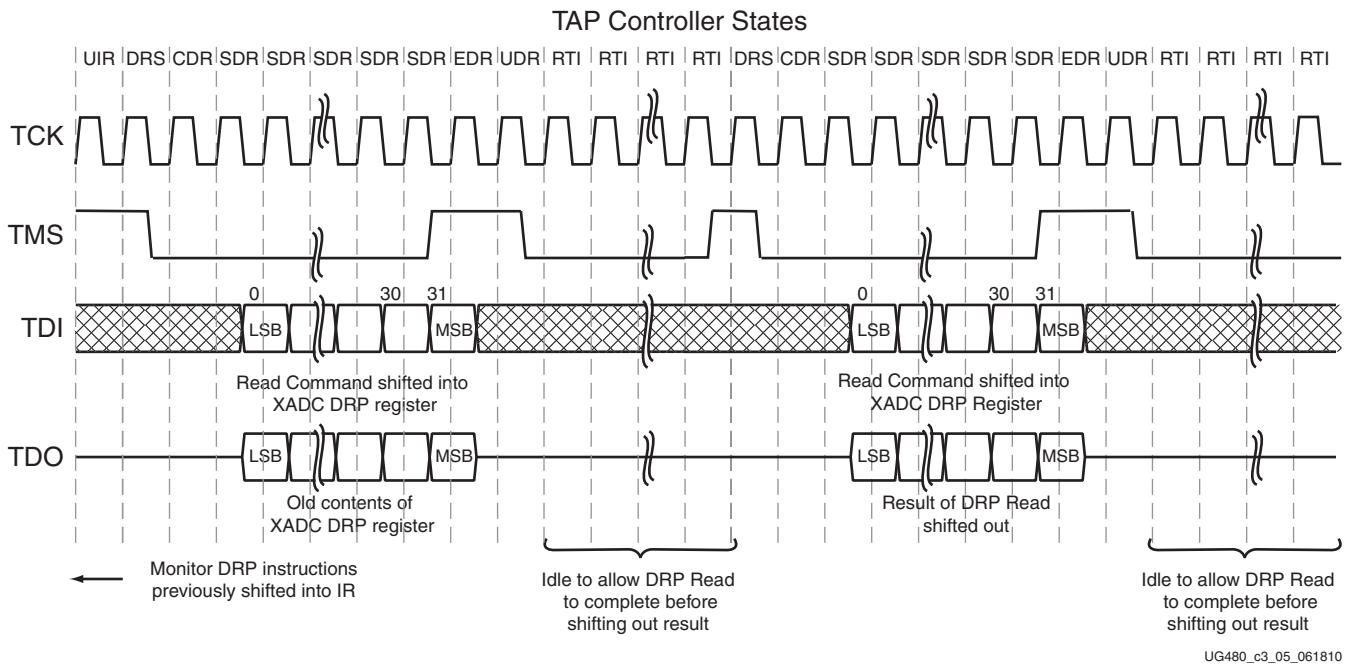


Figure 3-5: XADC JTAG DRP Read

First, the XADC 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 XADC 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 3-5). The timing diagram shows a number of idle states at the end of the first DR-scan operation, allowing the arbitrator enough time to fetch the XADC DRP data.

As mentioned previously, 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 XADC 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 3-5. Thus, as the result of a read operation is being shifted out of the XADC DR, an instruction for the next read can be shifted in.

## JTAG DRP Commands

The data shifted into the 32-bit XADC DR during a DR-scan operation instructs the arbitrator to carry out a write, read, or no operation on the XADC DRP. Figure 3-6 shows the data format of the JTAG DRP command loaded into the XADC DR. The first 16 LSBs of XADC DR [15:0] contain the DRP register data. For both read and write operations, the address bits XADC DR [25:16] hold the DRP target register address. The command bits XADC DR [29:26] specify a read, write, or no operation (see Table 3-13).

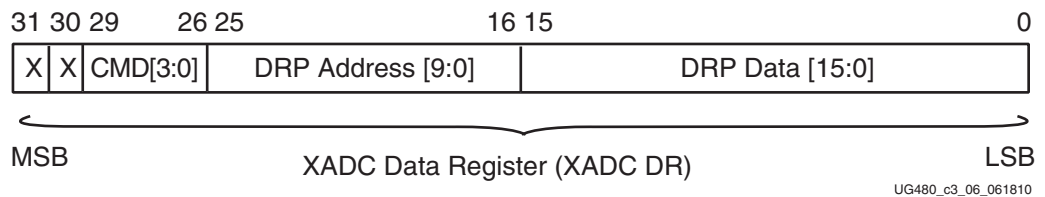


Figure 3-6: XADC JTAG DRP Command

Table 3-13: 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 FPGA, 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.

## DRP Arbitration

Because the DRP registers are accessed from two different ports (interconnect and JTAG TAP), access must be carefully managed. An arbitrator is implemented to manage potential conflicts between the FPGA logic (fabric) and JTAG port. Arbitration is managed

on a per transaction basis (a transaction is a single read/write operation to the DRP). The arbitration rules are as follows:

1. A JTAG transaction cannot be interrupted by the FPGA logic (fabric). The fabric transaction is queued by the arbitrator until the JTAG transaction has finished, and then the fabric transaction is completed.
2. A JTAG transaction cannot interrupt a fabric transaction already in progress. As soon as the fabric transaction is finished, then the JTAG transaction is completed.

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

## JTAGBUSY

JTAGBUSY becomes active during the update phase of a DRP transaction through the JTAG TAP. This signal resets when the JTAG XADC DR transaction is completed. Each read/write to the XADC 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 FPGA logic. DRDY does not go active until JTAGBUSY transitions Low and the interconnect transaction is completed. A second DRP access through the FPGA 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 (FPGA) 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

In some cases, it is simpler for the JTAG user 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 fabric 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 10 DCLK cycles for JTAGLOCKED to go Low after DCLK becomes active.

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

## XADC JTAG Reset

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

## Zynq-7000 AP SoC Processing System (PS) to XADC Dedicated Interface

The DRP JTAG interface described in [DRP JTAG Interface, page 44](#) is also used to provide a dedicated interface between the processor subsystem and the XADC block located in the programmable logic region of the Zynq-7000 AP SoC. This dedicated interface does not require the programmable logic to be configured. All the XADC functionality can be accessed by writing to and reading from the XADC status and configuration registers. For more details on how to enable this interface and programming models, see UG585, *Zynq+7000 All Programmable SoC Technical Reference Manual [Ref 4]*.

**Note:** When this interface is enabled, external JTAG access to the XADC as described in [DRP JTAG Interface, page 44](#) is no longer supported.

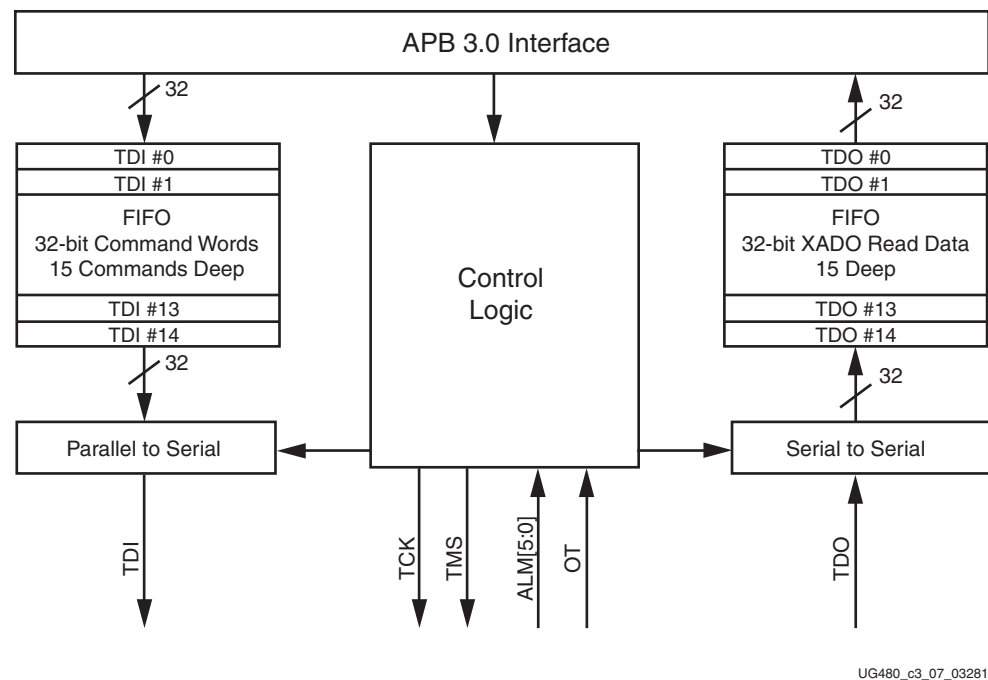


Figure 3-7: Processor to XADC Dedicated Interface

Figure 3-7 illustrates a simplified block diagram of the interface between the processor subsystem and the XADC block. To simplify the interface and also reduce the overhead on the processor, two 32-bit FIFOs have been implemented. Up to 15 32-bit DRP JTAG command words (see [Table 3-13](#)) can be loaded into the command FIFO. The control logic in the interface manages the parallel to serial conversion and writing the commands to the DRP JTAG interface. Serial data shifted out on TDO on the DRP JTAG interface is converted to a 32-bit parallel word and written to the DRP JTAG Read Data FIFO which is also 15 words deep. XADC read data is accessed by the processor by reading from this FIFO. The same timing and protocol as detailed in [DRP JTAG Interface, page 44](#), applies to this interface.

The XADC alarm signals OT and ALM[6:0] are also connected to this dedicated interface. These signals generate interrupts when the XADC automatic alarms go active. For more information, see [Automatic Alarms, page 59](#).



# XADC Operating Modes

---

The XADC includes a number of operating modes that cover some of the most common use cases for this kind of functionality. The most basic mode of operation is called *default mode*, where the XADC monitors all on-chip sensors and requires no configuration of the XADC. In the *simultaneous sampling mode*, the sequencer is used to operate both ADCs in lock step to sample two external analog inputs and store results in the status registers.

## Single Channel Mode

This mode is enabled when bits SEQ3 to SEQ0 in Control Register 41h are set to 0011 (see [Table 3-9, page 42](#)). In this mode, the user must select the channel for analog-to-digital conversion by writing to bit locations CH4 to CH0 in Control Register 40h. Various configurations for single channel mode, such as analog input mode ( $\overline{BU}$ ) 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 3-9](#)).

The channel sequencer functionality is implemented using eight control registers from addresses 48h to 4Fh on the DRP (see [Control Registers, page 39](#)). These eight registers can be viewed as four pairs of 16-bit registers. Each pair of registers controls one aspect of the sequencer functionality. Individual bits in each pair of registers (32 bits) enable specific functionality for a particular ADC channel. The four pairs of registers are:

- [ADC Channel Selection Registers \(48h and 49h\)](#)
- [ADC Channel Averaging \(4Ah and 4Bh\)](#)
- [ADC Channel Analog-Input Mode \(4Ch and 4Dh\)](#)
- [ADC Channel Settling Time \(4Eh and 4Fh\)](#)

### ADC Channel Selection Registers (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 two 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 fixed starting from the LSB (bit 0) of register 48h and ending with the MSB (bit 15) of register 49h. The behavior of the channel sequencer does change for simultaneous sampling mode and independent ADC mode (see [Simultaneous Sampling Mode](#) and [Independent ADC Mode](#), respectively).

**Table 4-1: Sequencer On-Chip Channel Selection (48h)**

Sequence Number	Bit	ADC Channel	Description
1	0	8	XADC calibration
-	1	9	Invalid channel selection
	2	10	
	3	11	
	4	12	
2 <sup>(1)</sup>	5	13	V <sub>CCPINT</sub>
3 <sup>(1)</sup>	6	14	V <sub>CCPAUX</sub>
4 <sup>(1)</sup>	7	15	V <sub>CCO_DDR</sub>
2/5 <sup>(1)</sup>	8	0	On-chip temperature
3/6 <sup>(1)</sup>	9	1	V <sub>CCINT</sub>
4/7 <sup>(1)</sup>	10	2	V <sub>CCAUX</sub>
5/8 <sup>(1)</sup>	11	3	V <sub>P</sub> , V <sub>N</sub> - Dedicated analog inputs
6/9 <sup>(1)</sup>	12	4	V <sub>REFP</sub> (1.25V)
7/10 <sup>(1)</sup>	13	5	V <sub>REFN</sub> (0V)
8/11 <sup>(1)</sup>	14	6	V <sub>CCBRAM</sub>
-	15	7	Invalid channel selection

**Notes:**

1. These channels are only supported on Zynq-7000 AP SoC devices.

**Table 4-2: Sequencer Auxiliary Channel Selection (49h)**

Sequence Number	Bit	ADC Channel	Description
9/12 <sup>(1)</sup>	0	16	VAUXP[0], VAUXN[0] - Auxiliary channel 0
10/13 <sup>(1)</sup>	1	17	VAUXP[1], VAUXN[1] - Auxiliary channel 1
11/14 <sup>(1)</sup>	2	18	VAUXP[2], VAUXN[2] - Auxiliary channel 2
12/15 <sup>(1)</sup>	3	19	VAUXP[3], VAUXN[3] - Auxiliary channel 3
13/16 <sup>(1)</sup>	4	20	VAUXP[4], VAUXN[4] - Auxiliary channel 4
14/17 <sup>(1)</sup>	5	21	VAUXP[5], VAUXN[5] - Auxiliary channel 5

Table 4-2: Sequencer Auxiliary Channel Selection (49h) (Cont'd)

Sequence Number	Bit	ADC Channel	Description
15/18 <sup>(1)</sup>	6	22	VAUXP[6], VAUXN[6] - Auxiliary channel 6 <sup>(1)</sup>
16/19 <sup>(1)</sup>	7	23	VAUXP[7], VAUXN[7] - Auxiliary channel 7 <sup>(1)</sup>
17/20 <sup>(1)</sup>	8	24	VAUXP[8], VAUXN[8] - Auxiliary channel 8
18/21 <sup>(1)</sup>	9	25	VAUXP[9], VAUXN[9] - Auxiliary channel 9
19/22 <sup>(1)</sup>	10	26	VAUXP[10], VAUXN[10] - Auxiliary channel 10
20/23 <sup>(1)</sup>	11	27	VAUXP[11], VAUXN[11] - Auxiliary channel 11
21/24 <sup>(1)</sup>	12	28	VAUXP[12], VAUXN[12] - Auxiliary channel 12
22/25 <sup>(1)</sup>	13	29	VAUXP[13], VAUXN[13] - Auxiliary channel 13 <sup>(1)</sup>
23/26 <sup>(1)</sup>	14	30	VAUXP[14], VAUXN[14] - Auxiliary channel 14 <sup>(1)</sup>
24/27 <sup>(1)</sup>	15	31	VAUXP[15], VAUXN[15] - Auxiliary channel 15 <sup>(1)</sup>

**Notes:**

1. Auxiliary channels 6, 7, 13, 14, and 15 are not supported on Kintex-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC device package options. Users should consult the package file for the device.

## ADC Channel Averaging (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](#), page 39). These registers also have the same bit assignments as the channel sequence registers listed in [Table 4-1](#) and [Table 4-2](#).

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 3-8](#), page 42). 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<sub>AUX</sub>[1], where an averaging of 16 is enabled on V<sub>AUX</sub>[1]. The sequence is Temperature, V<sub>AUX</sub>[1], Temperature, V<sub>AUX</sub>[1], ..., Temperature, V<sub>AUX</sub>[1] for each of the conversions where the temperature status register is updated. The V<sub>AUX</sub>[1] status register is updated after the averaging of the 16 conversions.

If averaging is enabled for the calibration channel by setting CAVG to a logic 0 (see [Control Registers](#), page 39), 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.

## ADC Channel Analog-Input Mode (4Ch and 4Dh)

These registers are used to configure an ADC channel as either unipolar or bipolar in the automatic sequence (see [Analog Inputs, page 26](#)). These registers also have the same bit assignments as the channel sequence registers listed in [Table 4-1](#) and [Table 4-2](#). 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 way. 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.

## ADC Channel Settling Time (4Eh and 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 (see [Chapter 5, XADC Timing](#)). However, by setting the corresponding bits (for external channels) to logic 1 in registers 4Eh and 4Fh, the associated channel can have its settling time extended to 10 ADCCLK cycles. The bit definitions (which bits correspond to which external channels) for these registers are the same as the sequencer channel selection shown in [Table 4-1](#) and [Table 4-2](#).

## Sequencer Modes

There are a number of sequencer modes as defined by [Table 3-9, page 42](#). These modes are described in this section.

### Default Mode

The default mode is enabled by setting the sequence bits to 0. In this mode of operation, the XADC automatically monitors the on-chip sensors and stores the results in the status registers. Both ADCs are calibrated in this mode and an averaging of 16 samples is applied to all sensors. The XADC operates independently of any other control register settings in this mode. The XADC also operates in default mode after initial power up and during FPGA configuration. [Table 4-3](#) shows the default sequence for the XADC.

**Note:** All alarm outputs (ALM[7:0]) except OT are disabled in default mode. ADC calibration is automatically enabled in default mode.

**Table 4-3: Default Mode Sequence**

Order	Channel	Address	Description
1	Calibration	08h	Calibration of ADC A and ADC B
2 <sup>(1)</sup>	V <sub>CCPINT</sub>	0Dh	V <sub>CCPINT</sub> supply sensor
3 <sup>(1)</sup>	V <sub>CCPAUX</sub>	0Eh	V <sub>CCPAUX</sub> supply sensor
4 <sup>(1)</sup>	V <sub>CCO_DDR</sub>	0Eh	V <sub>CCO_DDR</sub> supply sensor
2/5 <sup>(1)</sup>	Temp	00h	Temperature sensor
3/6 <sup>(1)</sup>	V <sub>CCINT</sub>	01h	V <sub>CCINT</sub> supply sensor
4/7 <sup>(1)</sup>	V <sub>CCAUX</sub>	02h	V <sub>CCAUX</sub> supply sensor

Table 4-3: Default Mode Sequence (Cont'd)

Order	Channel	Address	Description
5/8 <sup>(1)</sup>	V <sub>CCBRAM</sub>	06h	V <sub>CCBRAM</sub> supply sensor

**Notes:**

1. Only available on the Zynq-7000 AP SoC devices.

## Single Pass Mode

In single pass mode, the sequencer operates for one pass through the sequencer channel select registers (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 3-9, page 42](#) are set so as to enable the automatic channel sequencer in single pass mode, the sequence starts. The settings in sequencer registers 48h to 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 XADC defaults to Single Channel Mode described at the start of this chapter. Thus the XADC converts the channel selected by bits CH5 to CH0 in Configuration Register 0.

## Continuous Sequence Mode

The continuous sequence mode is similar to single pass mode; however, the sequence automatically restarts as long as the mode is enabled.

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. It is recommended that the XADC is placed in default mode by writing zeros to SEQ0 and SEQ1 while updating these registers. The XADC 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.

## Simultaneous Sampling Mode

When placed in simultaneous sampling mode, the sequencer automatically sequences through eight pairs of auxiliary analog input channels for simultaneous sampling and conversion as shown in [Table 4-4](#). This is useful in applications where it is necessary to preserve the phase relationship between two signals.

Auxiliary analog channels 0 to 7 are assigned to ADC A and are nominated as *A channels*. Auxiliary analog channels 8 to 15 are assigned to ADC B and are nominated as *B channels* (see [Figure 1-1, page 12](#)). One A channel and one B channel are always sampled and converted at the same time in simultaneous sampling mode. [Table 4-4](#) shows how A and B pairs are selected using sequencer channel register 49h. Other sequencer registers that define averaging, analog input mode, and settling time are also available for use in this mode.

Channel Averaging (4Ah and 4Bh) and Analog-Input Modes (4Ch and 4Dh) can be set on a per channel basis in simultaneous sampling mode. Thus it is possible to have an A channel configured as unipolar and a B channel configured as bipolar. Bit definitions as shown in [Table 4-1](#) and [Table 4-2](#) apply. The settling time is applied to a channel pair only;

thus setting bit 0 to 1 in register 4Eh sets the settling time to 10 ADCCLKs for both auxiliary channel 0 and channel 8.

**Table 4-4: Sequencer Register (49h) Bit Definitions for Simultaneous Sampling Mode**

Sequence Number	Bit	ADC Channel	Description
1	0	16, 24	Auxiliary channels 0 and 8
2	1	17, 25	Auxiliary channels 1 and 9
3	2	18, 26	Auxiliary channels 2 and 10
4	3	19, 27	Auxiliary channels 3 and 11
5	4	20, 28	Auxiliary channels 4 and 12
6	5	21, 29	Auxiliary channels 5 and 13 <sup>(1)</sup>
7	6	22, 30	Auxiliary channels 6 and 14 <sup>(1)</sup>
8	7	23, 31	Auxiliary channels 7 and 15 <sup>(1)</sup>
x	8	x	Undefined
x	9	x	Undefined
x	10	x	Undefined
x	11	x	Undefined
x	12	x	Undefined
x	13	x	Undefined
x	14	x	Undefined
x	15	x	Undefined

**Notes:**

1. These simultaneous sampling channels are not supported in Kintex-7 devices. Some other auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC device package options. Users should consult the package file for the device.

The on-chip temperature and power supplies can also be included in the sequence of channels monitored by setting the appropriate bits in Sequencer Register 48h. The second ADC (ADC B) does not carry out a conversion in simultaneous sampling mode when an internal sensor channel is being converted.

Automatic calibration cannot be enabled in simultaneous sampling mode. The XADC must be placed in default mode or other sequencer mode to carry out a calibration. The ADCs are automatically calibrated on power-up, which is sufficient for most applications.

## Timing

The timing for simultaneous sampling mode is identical to other XADC modes (see [Chapter 5, XADC Timing](#)). Because the ADCs run in lock step, both status registers are updated at exactly the same time. Both continuous and event driven timing modes can be used.

## Independent ADC Mode

In the independent ADC mode, ADC A is used to implement a fixed “monitoring mode” similar to the default mode except the alarm functions are enabled (see [Automatic Alarms](#),

page 59). In this mode, the alarm outputs are active, and the alarm threshold must be correctly configured by the user. Like default sequencer mode, the averaging is fixed at 16 samples.

ADC B is available to be used with the external analog input channels only. This mode frees up the second ADC for use in a customer application where only a single ADC is required while maintaining monitoring of the FPGA for reliability, safety, and anti-tamper applications.

Only the dedicated channel and auxiliary analog input channels can be assigned to ADC B in this sequencer mode. The internal channels (sensors) are automatically assigned to ADC A, which automatically monitors these channels and generates alarms based on the user-defined alarm thresholds.

As with simultaneous sampling mode, it is not possible to select an automatic calibration of ADC B in this sequencer mode. ADC A is automatically calibrated in this mode. To carry out a calibration on ADC B and maintain on-chip monitoring, the default sequence mode should be selected. The XADC can then be returned to Independent ADC mode after EOS goes High at least once.

### Sequencer Operation

Channel selection for independent ADC mode is defined using Sequencer Channel Registers 48h and 49h (see Table 4-5 and Table 4-6). Internal sensors channels are automatically monitored and calibrated and cannot be assigned to the sequence for ADC B by users. Other sequencer registers that define settling time, analog input mode, and averaging remain unchanged.

Table 4-5: Independent ADC Sequencer Mode Bit Definitions (48h)

Sequence Number	Bit	ADC Channel	Description
-	0	-	Not Defined
	1		Not Defined
	2		Not Defined
	3		Not Defined
	4		Not Defined
	5		Not Defined
	6		Not Defined
	7		Not Defined
	8		Not Defined
	9		Not Defined
	10		Not Defined
1	11	3	$V_P/V_N$
--	12	-	Not Defined
	13		Not Defined
	14		Not Defined
	15		Not Defined

Table 4-6: Independent ADC Sequencer Mode Bit Definitions (49h)

Sequence Number	Bit	ADC Channel	Description
2	0	16	Auxiliary channel 0
3	1	17	Auxiliary channel 1
4	2	18	Auxiliary channel 2
5	3	19	Auxiliary channel 3
6	4	20	Auxiliary channel 4
7	5	21	Auxiliary channel 5
8	6	22	Auxiliary channel 6 <sup>(1)</sup>
9	7	23	Auxiliary channel 7 <sup>(1)</sup>
10	8	24	Auxiliary channel 8
11	9	25	Auxiliary channel 9
12	10	26	Auxiliary channel 10
13	11	27	Auxiliary channel 11
14	12	28	Auxiliary channel 12
15	13	29	Auxiliary channel 13 <sup>(1)</sup>
16	14	30	Auxiliary channel 14 <sup>(1)</sup>
17	15	31	Auxiliary channel 15 <sup>(1)</sup>

**Notes:**

1. Auxiliary channels 6, 7, 13, 14, and 15 are not supported on Kintex-7 devices. Some auxiliary analog channels might also not be supported in certain Virtex, Artix, and Zynq-7000 AP SoC device package options. Users should consult the package file for the device.

## Timing

The timing in this mode of operation is also the same for both ADCs. ADC A monitors the internal sensors at the same sample rate and mode selected for ADC B. Again, ADC B can be operated in continuous or event mode timing. In the case of event mode timing, internal sensor monitoring also stops if the CONVST signal is stopped. Therefore care should be taken to ensure periodic pulsing of CONVST even when ADC B is not being used by the application, if constant coverage of the on-chip environment is required.

Timing for continuous and event modes remain the same as shown in [Chapter 5, XADC Timing](#).

## External Multiplexer Mode

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

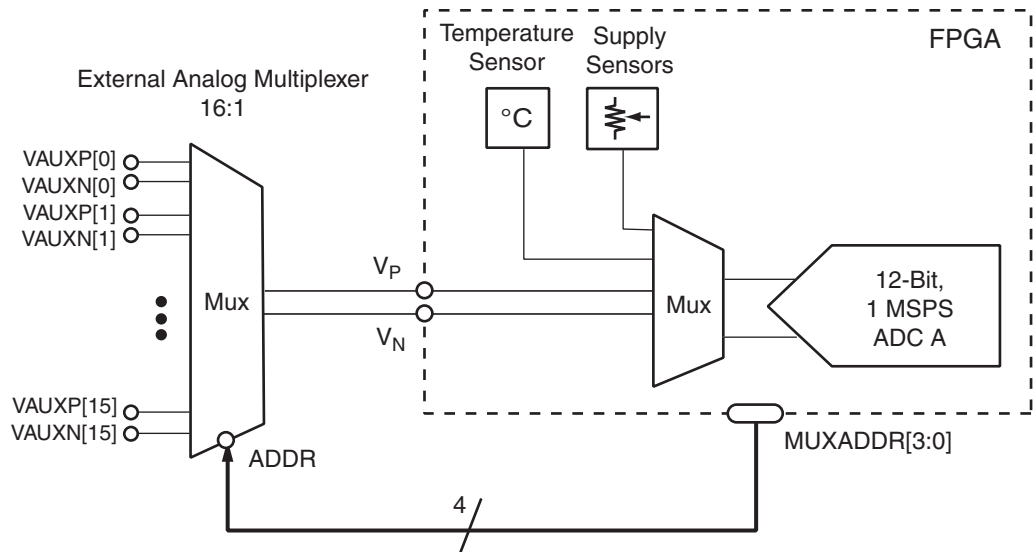
The XADC track/hold amplifiers return to track mode as soon as a conversion starts (see [Figure 4-3](#)). Therefore the acquisition on the next channel can start during the current conversion cycle. An output bus called MUXADDR[4:0] allows the XADC to control an external multiplexer. The address on this bus reflects the channel currently being acquired,



and it changes state as soon as the XADC enters acquisition mode. Users can also nominate the channel to be used with an external multiplexer.

## External Multiplexer Operation

Figure 4-1 illustrates the external multiplexer concept. In this example an external 16:1 analog multiplexer is used instead of consuming the 32 FPGA I/Os required to implement the 16 auxiliary analog input channels using the internal multiplexer. Any four FPGA I/Os can be used for the external multiplexer decode operation. As shown in Figure 4-1, the dedicated analog inputs ( $V_P/V_N$ ) are used to connect the external multiplexer to the XADC 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](#), page 39).

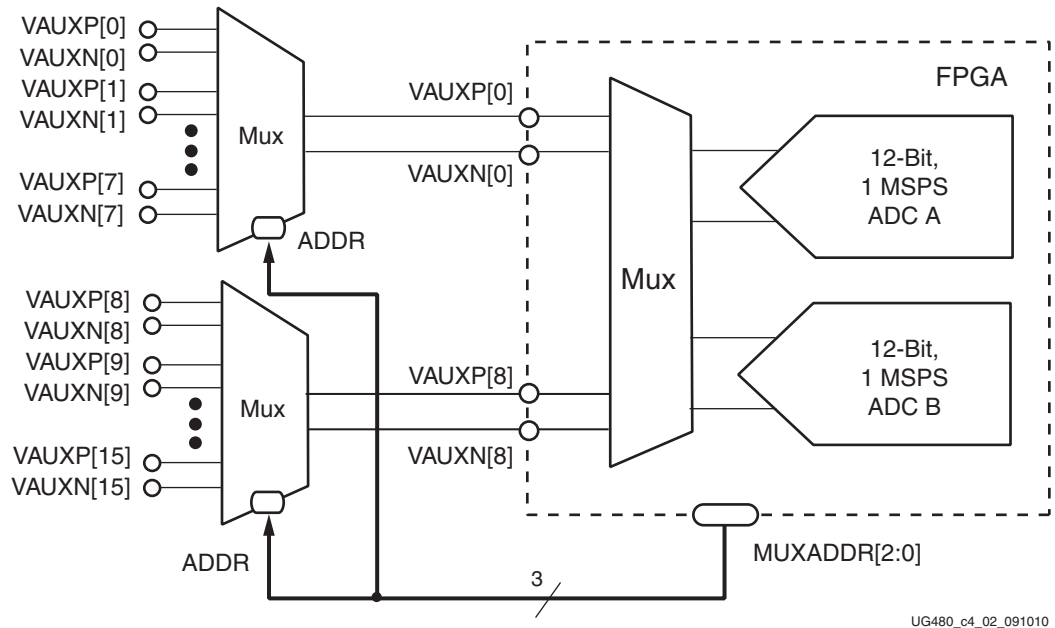


UG480\_c4\_01\_091010

Figure 4-1: External Multiplexer Mode

When the MUX bit is set to a 1, the channel selection bits (CH0 to CH4) in Configuration Register 0 are used to nominate the channel for connection to the external multiplexer. For example, as shown in Figure 4-1, the dedicated analog input channel  $V_P/V_N$  is used. In this case, channel 3 (00011b) should be written to CH4 to CH0 in Control Register 40h. Any one of the auxiliary channels can also be used for connection to the external multiplexer. In the case of simultaneous sampling mode (see [Simultaneous Sampling Mode](#), page 53), two channels must be allocated to two external multiplexers to support simultaneous sampling.

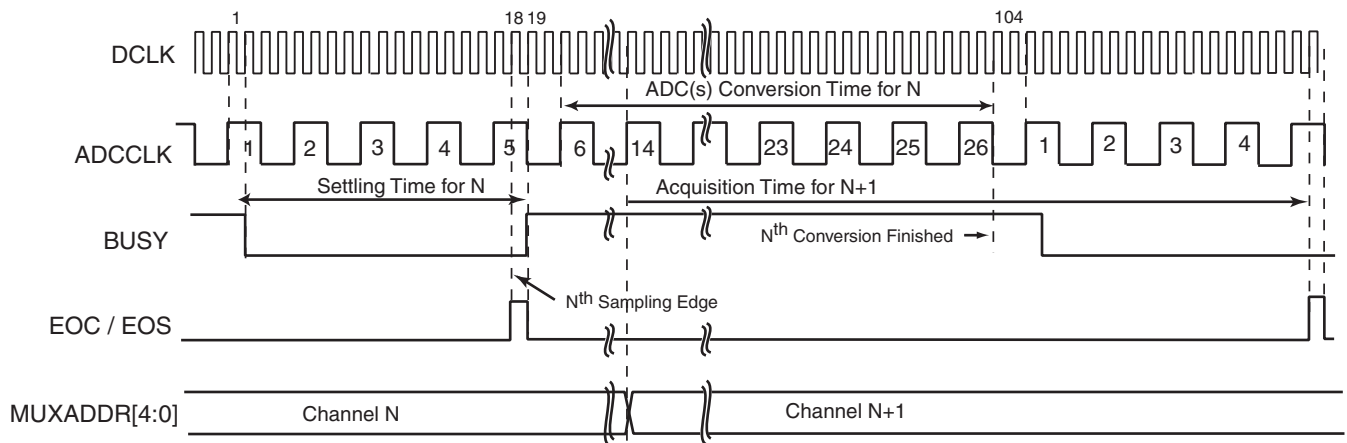
Figure 4-2 illustrates how the external multiplexer mode is implemented for simultaneous sampling mode. The channels selected for connection are also selected by writing to CH4 to CH0 but are allocated in pairs as defined in Table 4-4. For example, writing 16 (10000) to CH4 to CH0 would select auxiliary channels 0 and 8 for connection to external multiplexers as shown in Figure 4-2.



UG480\_c4\_02\_091010

Figure 4-2: External Multiplexer Mode for Simultaneous Sampling

In both cases, the MUXADDR[4:0] bus is used to automatically select the external multiplexer channel. Figure 4-3 shows how the MUXADDR bus reflects the next channel selection ( $N + 1$ ). MUXADDR changes state eight ADCCLK cycles after BUSY goes High.



UG480\_c4\_03\_110310

Figure 4-3: External Multiplexer Timing (Continuous Sampling Mode)

## Maximum and Minimum Status Registers

The XADC also tracks the minimum and maximum values recorded for the internal sensors since the last power-up or since the last reset of the XADC control logic. The maximum and minimum values recorded are stored in the DRP status registers starting at address 20h (see [Status Registers](#), page 34). 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.

Table 4-7: Maximum and Minimum Registers

Status Register	Description
20h	Temp Max - Maximum Temperature Recorded
21h	V <sub>CCINT</sub> Max - Maximum V <sub>CCINT</sub> Recorded
22h	V <sub>CCAUX</sub> Max - Maximum V <sub>CCAUX</sub> Recorded
23h	V <sub>CCBRAM</sub> Max - Maximum V <sub>CCBRAM</sub> Recorded
24h	Temp Min - Minimum Temperature Recorded
25h	V <sub>CCINT</sub> Min - Minimum V <sub>CCINT</sub> Recorded
26h	V <sub>CCAUX</sub> Min - Minimum V <sub>CCAUX</sub> Recorded
27h	V <sub>CCBRAM</sub> Min - Minimum V <sub>CCBRAM</sub> Recorded
28h	V <sub>CCPINT</sub> <sup>(1)</sup> Max - Maximum V <sub>CCPINT</sub> Recorded
29h	V <sub>CCPAUX</sub> <sup>(1)</sup> Max - Maximum V <sub>CCPAUX</sub> Recorded
2Ah	V <sub>CCO_DDR</sub> <sup>(1)</sup> Max - Maximum V <sub>CCO_DDR</sub> Recorded
2Bh	Unassigned
2Ch	V <sub>CCPINT</sub> <sup>(1)</sup> Min - Minimum V <sub>CCPINT</sub> Recorded
2Dh	V <sub>CCPAUX</sub> <sup>(1)</sup> Min - Minimum V <sub>CCPAUX</sub> Recorded
2Eh	V <sub>CCO_DDR</sub> <sup>(1)</sup> Min - Minimum V <sub>CCO_DDR</sub> Recorded
2Fh	Unassigned
<b>Notes:</b>	
1. Only available on the Zynq-7000 AP SoC devices.	

## Automatic Alarms

The XADC also generates an alarm signal on the logic outputs ALM[7:0] when an internal sensor measurement (Temperature, V<sub>CCINT</sub>, V<sub>CCAUX</sub>, V<sub>CCBRAM</sub>, V<sub>CCPINT</sub>, V<sub>CCPAUX</sub>, or V<sub>CCO\_DDR</sub>) 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 ALM6 to ALM0 in Configuration Register 1. The alarm thresholds are stored in Control Registers 50h to 5Fh. Table 4-8 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 2-8, page 31 and Figure 2-9, page 32).

Table 4-8: Alarm Threshold Registers

Control Register	Description	Alarm
50h	Temperature Upper	ALM[0]
51h	V <sub>CCINT</sub> Upper	ALM[1]
52h	V <sub>CCAUX</sub> Upper	ALM[2]
53h	OT Upper	OT
54h	Temperature Lower	ALM[0]
55h	V <sub>CCINT</sub> Lower	ALM[1]
56h	V <sub>CCAUX</sub> Lower	ALM[2]
57h	OT Lower	OT
58h	V <sub>CCBRAM</sub> Upper	ALM[3]
59h	V <sub>CCPINT</sub> Upper <sup>(1)</sup>	ALM[4]
5Ah	V <sub>CCPAUX</sub> Upper <sup>(1)</sup>	ALM[5]
5Bh	V <sub>CCO_DDR</sub> Upper <sup>(1)</sup>	ALM[6]
5Ch	V <sub>CCBRAM</sub> Lower	ALM[3]
5Dh	V <sub>CCPINT</sub> Lower <sup>(1)</sup>	ALM[4]
5Eh	V <sub>CCPAUX</sub> Lower <sup>(1)</sup>	ALM[5]
5Fh	V <sub>CCO_DDR</sub> Lower <sup>(1)</sup>	ALM[6]

**Notes:**

1. Only Zynq-7000 AP SoC devices support these channels.

## Supply Sensor Alarms

When the measured value on the supply sensor for V<sub>CCINT</sub>, V<sub>CCAUX</sub>, V<sub>CCBRAM</sub>, V<sub>CCPINT</sub>, V<sub>CCPAUX</sub>, or V<sub>CCO\_DDR</sub> is greater than the thresholds in Control Registers 51h, 52h, 58h, 59h, 5Ah, and 5Bh or less than the thresholds in Control Registers 55h, 56h, 5Ch, 5Bh, 5D, and 5Eh, respectively, the output alarms go active. The alarms are reset when a subsequently measured value falls inside the threshold.

## Thermal Management

The on-chip temperature measurement is used for critical temperature warnings. The default Over Temperature threshold is 125°C. This threshold is used for pre-configuration or when the contents of the OT Upper alarm register have not been configured (see Table 4-8). To override this default condition, the 12 MSBs of the OT Upper Register (Control Register 53h) must be set using the temperature sensor transfer function (see Figure 2-8). In addition, the four LSBs must be set to 0011b.

When the die temperature exceeds a factory set limit of 125°C or a user-defined threshold, the Over-Temperature alarm logic output (OT) becomes active. This feature can be disabled by the user when the XADC is instantiated in a design. The OT function is disabled by writing a 1 to the OT bit in Configuration Register 1. The OT signal resets when the FPGA temperature has fallen below a user-programmable limit in Control Register 57h (see Table 4-8). When the automatic power-down feature is enabled, the OT

signal can be used to trigger a device power down. When OT goes High, the FPGA enters power down approximately 10 ms later. The power-down feature initiates a configuration shutdown sequence, disabling the device when finished and asserts GHIGH to prevent any contention (see UG470, *7 Series FPGAs Configuration User Guide* [Ref 3]). When OT is deasserted, GHIGH also is deasserted and the start-up sequence is initiated releasing all global resources. By default this functionality is disabled and must be explicitly enabled.

The automatic power down is enabled by using a configuration option in the ISE® design tools. To generate a programming file, the user checks the “Power Down Device if Over Safe Temperature” option under “Configuration Options” on the Process Properties GUI. Alternatively, the user can use the `bitgen -g` command line option `OverTempPowerDown: [Enable | Disable]`. When the FPGA enters power down, the XADC continues to operate in whatever mode was configured prior to power down using an internal clock oscillator. The FPGA automatically powers up once the temperature has fallen below the user-programmable lower limit. The XADC OT signal can also be reset by writing a 1 to the OT bit in XADC Configuration Register 1 via the JTAG DRP interface. On-chip sensors are monitored via the JTAG TAP during device power down. During power down, the XADC uses an internal oscillator instead of DCLK. On power-up, several DCLK cycles are required to resynchronize the DRP (JTAGLOCKED remains High until the DRP is ready for use again).

A second user-programmable temperature threshold level (Control Register 50h) is used to carry out a user-defined thermal management procedure, such as powering on or controlling the speed of a fan. An alarm signal (ALM[0]) becomes active when the FPGA temperature exceeds the limit in this register. The signal is available through the interconnect and is routed using the FPGA resources. The alarm signal resets when the temperature falls below the threshold in Control Register 54h. This operation differs for the supply sensor alarm, because the supply alarm resets when the measurement is between the upper and lower thresholds.

## XADC Enhanced Linearity Mode

Starting with ISE design tools, version 13.4, a new BitGen option called `XADCEnhancedLinearity` is available. It enhances the linearity (INL) and dynamic performance (SNR and THD) of the ADCs in the XADC block. This new `bitgen -g` command line option is `XADCEnhancedLinearity = [ON | OFF]`. By default this BitGen option is OFF. This mode was also OFF by default in all previous releases of the ISE design tools.

## Changes to Offset and Gain Calibration Operation

If the new `XADCEnhancedLinearity` BitGen option is not enabled (i.e., OFF), there is no change to the XADC operation when offset and gain calibration is enabled. See [XADC Calibration Coefficients, page 38](#), for an explanation of how the ADC offset and gain calibration is enabled. However if this new BitGen option is enabled, some differences in the ADC transfer functions are observed when offset and gain calibration coefficients are applied.

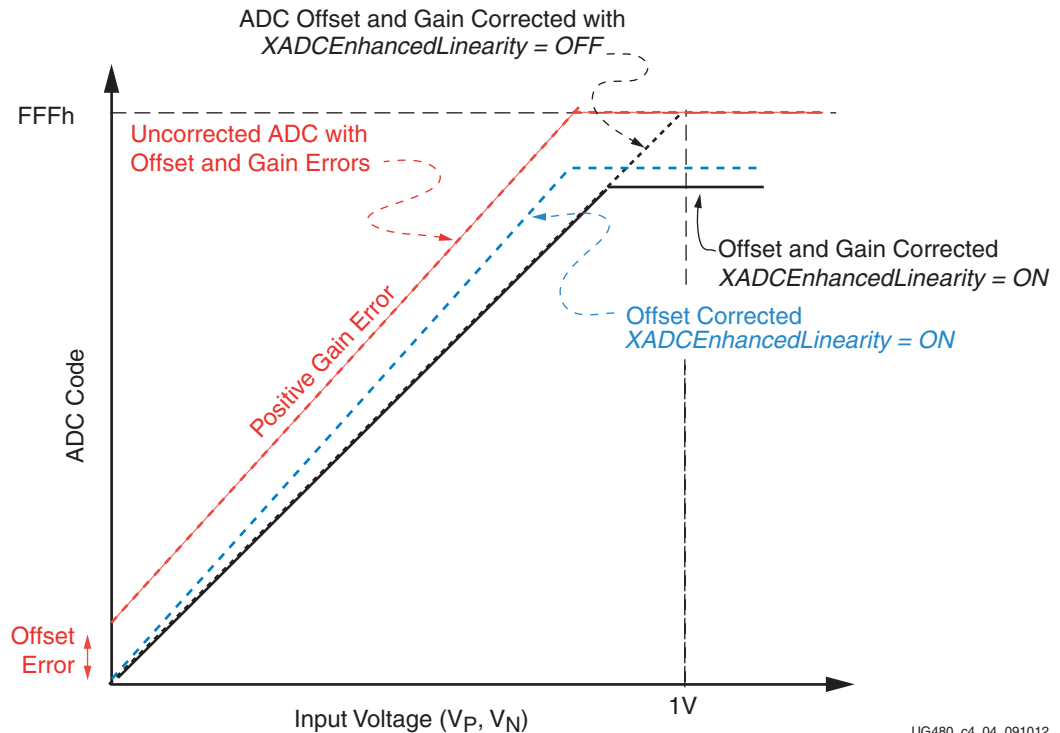


Figure 4-4: **ADC Calibration with *XADCEnhancedLinearity = ON***

Figure 4-4 illustrates the impact of enabling offset and gain error correction with *XADCEnhancedLinearity = ON*. In this example, the red line shows the impact of offset and gain errors on the ADC transfer function. This is exaggerated for clarity. To remove the offset error, a constant correction (offset calibration coefficient) is subtracted from the ADC output. This is shown as the blue line in Figure 4-4. The black solid line shows the impact of the gain error and offset correction. Because the ADC uncalibrated output reached full scale (FFFh) before the input reached 1V (due to offset and gain errors), part of the ADC transfer function is no longer available even after calibration. Thus the ADC maximum code stays at a value less than FFFh even for input voltages of 1V. This clipping range is typically no more than about 10 mV around either 0V or 1V input depending on the ADC offset and gain errors. For example, if the offset error was negative in the example shown in Figure 4-4, ADC would produce a 0 code out until the analog input exceeded this offset with offset correction enabled.

**Note:** This clipping effect is not seen when the BitGen option *XADCEnhancedLinearity = OFF*.

# XADC Timing

---

All XADC timing is synchronized to the DRP clock (DCLK). The ADCCLK is generated by dividing DCLK by the user selection in Configuration Register 2 (see [Control Registers, page 39](#)). ADCCLK is an internal clock used by the ADCs and is not available externally. ADCCLK is only included here to aid in describing the timing.

The ADC block is operated in one of two possible timing modes, continuous sampling mode and event driven sampling mode.

In continuous sampling mode, the ADC automatically starts a new conversion at the end of a current conversion cycle. In event sampling mode, the user must initiate the next conversion after the current conversion cycle ends by using the CONVST or CONVSTCLK inputs. The operating mode is selected by writing to Configuration Register 0 (see [Control Registers](#)).

Refer to the respective 7 series FPGAs data sheet for the latest XADC timing specifications. The robust nature of the XADC ensures continued and correct operation even if the external clock input DCLK is stopped. In this situation, the XADC automatically switches over to an internal clock oscillator to continue to operate as configured. After configuration, the DCLK input requires 20 DCLKs to resynchronize to the external clock. The JTAGLOCKED signal is High during this period. It is not possible to access the DRP until resynchronization has occurred.

## Continuous Sampling

In continuous sampling mode, the ADCs continue to carry out a conversion on the selected analog inputs. [Figure 5-1](#) shows the timing associated with continuous sampling mode. The ADCCLK is generated by a clock divider. The analog-to-digital conversion process is made up of two parts, the acquisition phase and the conversion phase.

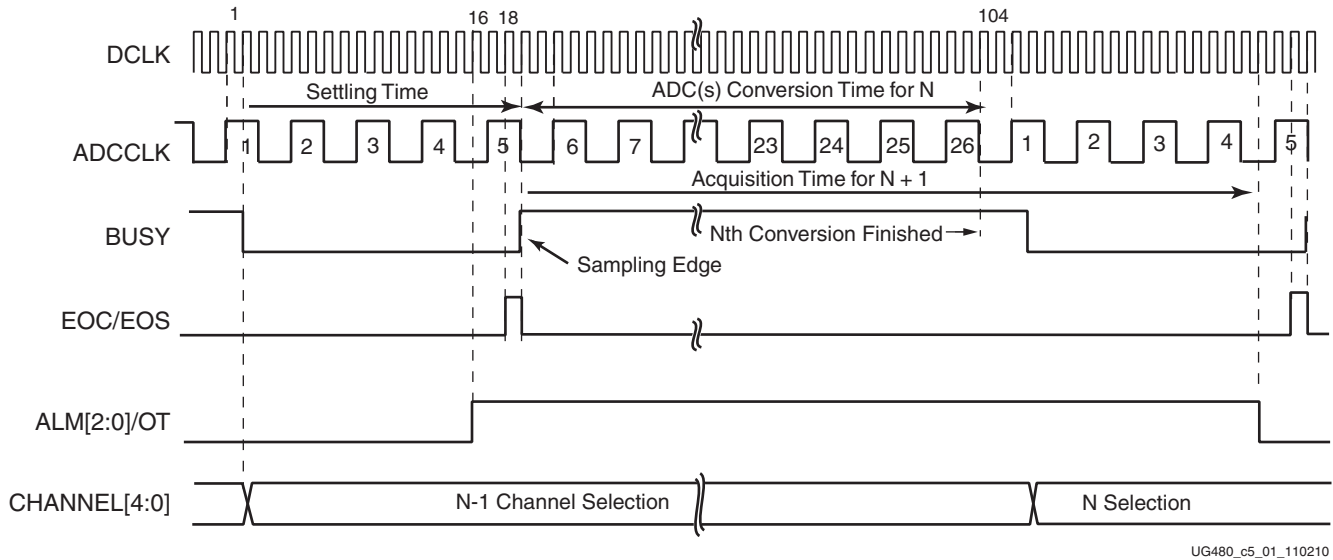


Figure 5-1: Continuous Sampling Mode

## Acquisition Phase

During the acquisition phase, the ADC acquires the voltage on a selected channel to perform the conversion. The acquisition phase involves charging a capacitor in the ADC to the voltage on the selected channel (see [Analog Input Description, page 27](#) for more information). The time required to charge this capacitor depends on the selected input-channel source impedance. The XADC allows the acquisition of the next channel to start during the current conversion. This is possible because the ADC has a separate Track-and-Hold amplifier (T/H). Thus when the ADC starts to convert an input voltage, the T/H is free to start precharging to the next voltage to be converted. This allows for a faster sampling rate because the ADC does not need to acquire the full input voltage at the end of the current conversion. By default a *settling period* of four ADCCLK cycles is left between the end of the current conversion and the start of the next conversion. This can be extended to 10 ADCCLK cycles by setting the ACQ bits in either the control registers or the sequence registers.

When operating in single channel mode (see [Chapter 4, XADC Operating Modes](#)), the user must write to Configuration Register 0 to select the next channel for conversion. The address of the next channel to convert is read from the control registers at the end of the previous conversion (when BUSY goes Low), and the analog signal is acquired during the current conversion. Additional acquisition or settling time after the end of the current conversion is configured by the ACQ bit in Configuration Register 0.

For more information on the effects of source impedance on the acquisition, see [Analog Input Description](#).

## Conversion Phase

The conversion phase starts on the sampling edge (next rising edge of DCLK) at the end of the 4 or 10 ADCCLK cycles settling time. The BUSY signal transitions to an active High on the next rising edge of DCLK to indicate the ADC is carrying out a conversion. The conversion phase is 22 ADCCLK cycles long. The conversion result transfers to the status registers 16 DCLK cycles after BUSY transitions Low, and the EOC logic output pulses High for one DCLK cycle (see [Figure 5-1](#)). If the channel being converted is also being



averaged, then the averaged data is only transferred to the status registers when the last sample result has been converted. Thus, if a channel is being averaged, an EOC pulse is only generated for the last conversion result, such as the 16th, 64th, 256th sample. When averaging is enabled, write operations to the configuration registers are not acted on until EOC is pulsed, i.e., when the averaging operation is complete.

When XADC is being operated in a sequence mode, the user identifies the channel being converted by monitoring the channel address (CHANNEL[4:0]) logic outputs. The multiplexer channel address of the channel being converted is updated on these logic outputs when BUSY transitions Low at the end of the conversion phase. The channel address outputs can be used with the EOC and DRDY signals to automatically latch the contents of the output data registers into a FIFO or block RAM. This is accomplished by connecting the CHANNEL[4:0] outputs to DADDR[4:0] (with DADDR[6:5] = 0), using EOC as a DEN (enable) for the DRP, and using DRDY as a WE (write enable) for the block RAM.

XADC EOS signal has the same timing as EOC. This signal is pulsed when the output data register for the last channel in a programmed channel sequence is updated.

## Event-Driven Sampling

Figure 5-2 illustrates the event-driven sampling mode. In this operating mode, the sampling instant and subsequent conversion process are initiated by a trigger signal called convert start (CONVST). Event-driven sampling mode is used when precise control over the sampling instant is necessary.

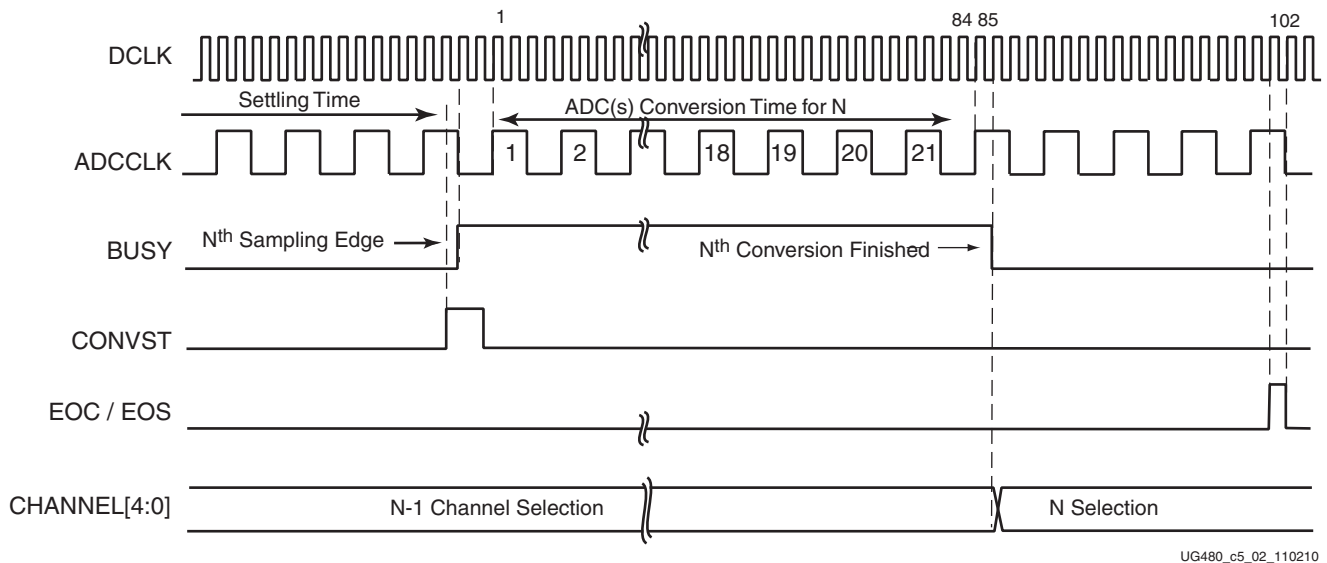


Figure 5-2: Event Driven Sampling Mode

The DCLK must always be present when using event-driven sampling mode. If no DCLK is present, the XADC reverts to continuous mode timing using an internal clock oscillator.

A Low-to-High transition (rising edge) on CONVST or CONVSTCLK defines the exact sampling instant for the selected analog-input channel. The BUSY signal transitions High just after the sampling instant on the next rising edge of DCLK. CONVST can be an asynchronous external signal in which case the XADC automatically resynchronizes the conversion process to the ADCCLK.

As with the continuous sampling mode, enough time must be allowed for the acquisition phase, that is, the time between a channel change and the sampling edge (the rising edge of CONVST or CONVSTCLK, see [Analog Input Description, page 27](#)). This allows the ADC to acquire the new signal before it is sampled by the CONVST signal and the conversion phase starts. The ACQ bit has no meaning in event-sampling mode because the sampling instant is controlled by CONVST/CONVSTCLK. Therefore, the acquisition time on a selected channel is also controlled by CONVST/CONVSTCLK. CONVST and CONVSTCLK are logically ORed within the XADC. The T/H starts to acquire the voltage on the next channel as soon as BUSY goes High and a conversion starts.

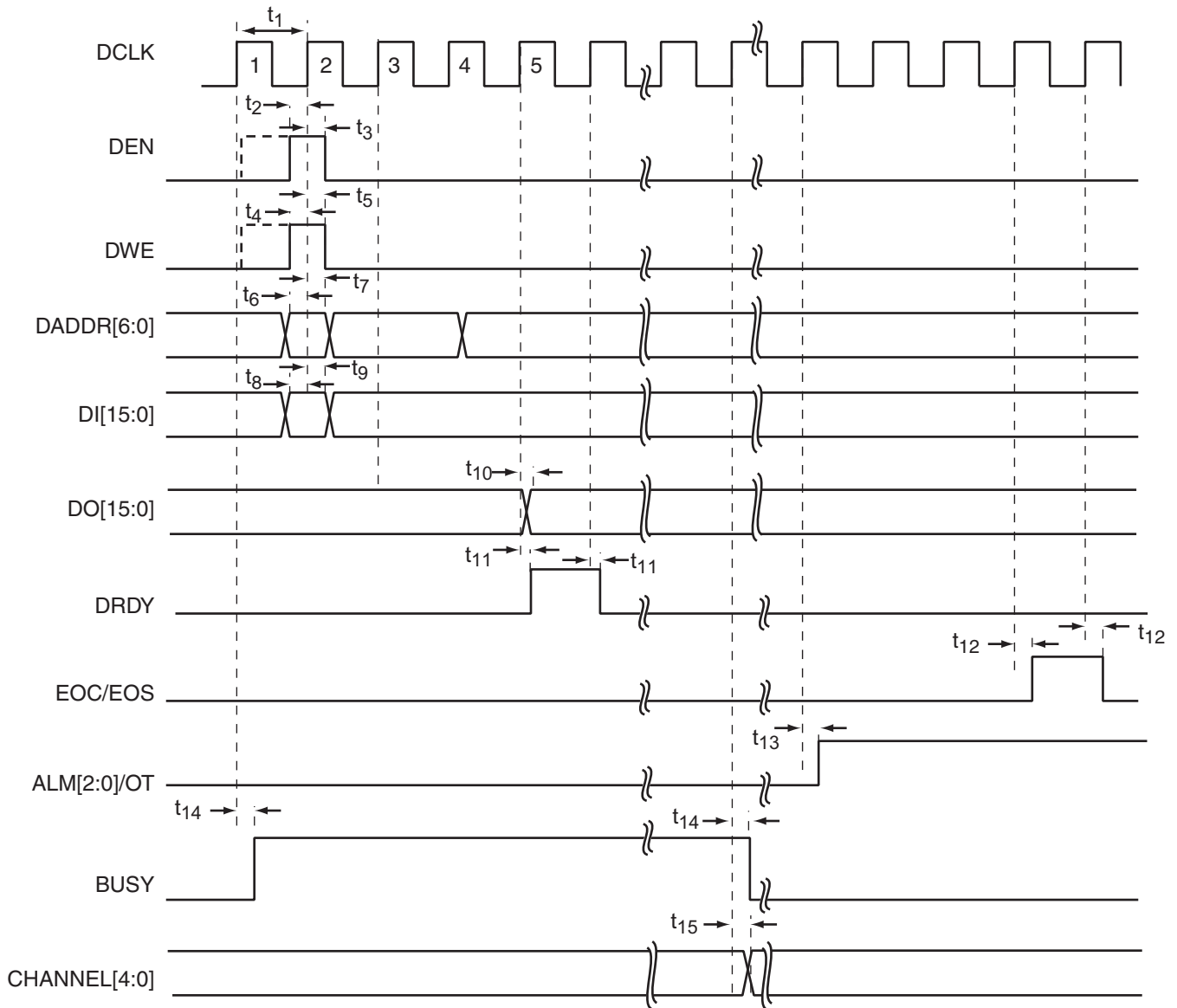
After the analog input has been sampled by a rising edge on CONVST/CONVSTCLK, a conversion is initiated on the next rising edge of ADCCLK. After a conversion has been initiated by CONVST, it is not possible to interrupt the conversion or start a new conversion until BUSY goes Low.

After BUSY goes Low, the conversion result is transferred to the output status registers 16 DCLK cycles later, and the EOC logic output pulses High for one DCLK cycle at this time. If the channel being converted is also being filtered, the averaged data is only transferred to the status registers when the last sample result has been converted. Thus, if a channel is being averaged, no EOC pulse is generated for all but the last conversion result (such as the 16th, 64th, and 256th sample), depending on the filter setting. The EOC, EOS, and CHANNEL[4:0] outputs operate in the same way as in the continuous sampling mode, (see [Continuous Sampling](#)).

## Dynamic Reconfiguration Port (DRP) Timing

[Figure 5-3](#) illustrates a DRP read and write operation. When the DEN is logic High, the DRP address (DADDR) and write enable (DWE) inputs are captured on the next rising edge of DCLK. DEN should only go high for one DCLK period.

If DWE is logic Low, a DRP read operation is carried out. The data for this read operation is valid on the DO bus when DRDY goes high. Thus DRDY should be used to capture the DO bus. For a write operation, the DWE signal is logic High and the DI bus and DRP address (DADDR) is captured on the next rising edge of DCLK. The DRDY signal goes 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 has gone low.



UG480\_c5\_03\_062010

Figure 5-3: DRP Detailed Timing



# Application Guidelines

---

The 7 series FPGAs XADC is a precision analog measurement system based on a 12-bit analog-to-digital converter (ADC) with an LSB size approximately equal to 250  $\mu\text{V}$ . To achieve the best possible performance and accuracy with all measurements (both on-chip and external), a number of dedicated pins for the ADC reference and power supply are provided. Care must be taken with the connection of these pins 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 ( $V_{\text{REFP}}$ and $V_{\text{REFN}}$ )

These high-impedance inputs are used to deliver a differential reference voltage for the analog-to-digital conversion process. ADCs are only as accurate as the reference provided. Any reference-voltage error results in a gain error vs. the ideal ADC transfer function (see [Chapter 2, Analog-to-Digital Converter](#)). 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  $V_{\text{REFP}}$  and  $V_{\text{REFN}}$  should be maintained at  $1.25\text{V} \pm 0.2\%$  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). The 1.25V reference should be placed as close as possible to the reference pins and connected directly to the  $V_{\text{REFP}}$  input, using the decoupling capacitors recommended in the reference IC data sheet. The recommended reference connections are illustrated in [Figure 6-1](#).

The XADC also has an on-chip reference option which is selected by connecting  $V_{\text{REFP}}$  and  $V_{\text{REFN}}$  to ADCGND as shown in [Figure 6-1](#). Due to reduced accuracy, the on-chip reference does impact the measurement performance of the XADC as explained previously. The performance with on-chip reference is specified in the respective 7 series FPGAs data sheet.

## Analog Power Supply and Ground ( $V_{\text{CCADC}}$ and GNDADC)

These inputs provide the power supply and ground reference for the analog circuitry in the XADC. 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 would not be advisable to use the digital ground as an analog ground reference for XADC.

Similarly, for the digital supplies for the FPGA 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 6-1 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 which needs to be considered. The filtering should ensure no more than 1 LSB (250  $\mu$ V) of noise on the reference output to minimize any impact on ADC accuracy at 12 bits.

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, page 26, where the common ground impedance  $R_G$  converts digital switching currents into a noise voltage for the analog circuitry. While a separate analog ground plane is recommended for 12-bit operation, it is often not possible or practical to implement a separate analog ground plane in a design. For example, if a user only intends to use the on-chip sensors, 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 6-1. This is illustrated in PC Board Design Guidelines.

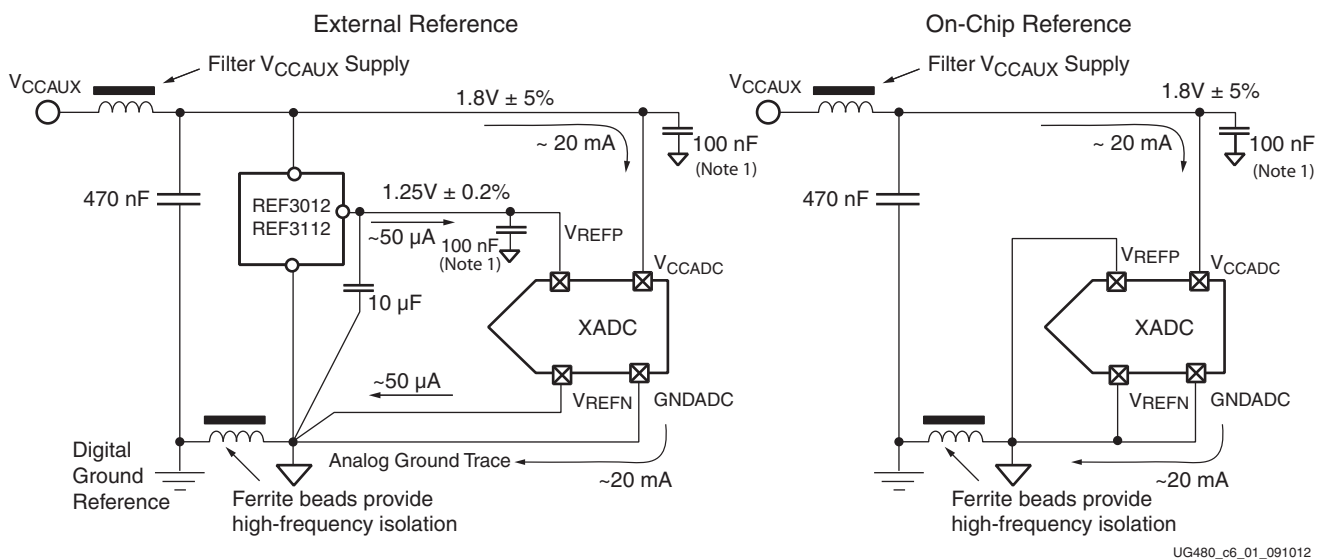


Figure 6-1: ADC Power and Ground Connections

Notes relevant to Figure 6-1:

1. Place the 100 nF capacitor as close as possible to the package balls (see PC Board Design Guidelines, page 72).

The ferrite bead behaves like a resistor at high frequencies and functions as a lossy inductor. A typical ferrite impedance vs. frequency plot is shown in Figure 6-2. 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.

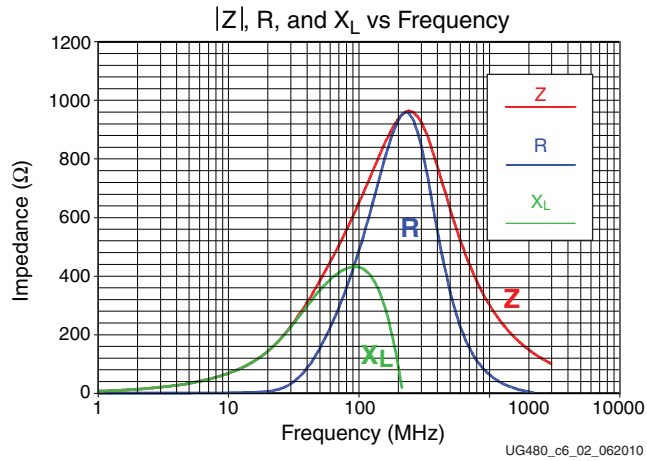


Figure 6-2: Ferrite Bead Characteristic

## 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 6-3 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.

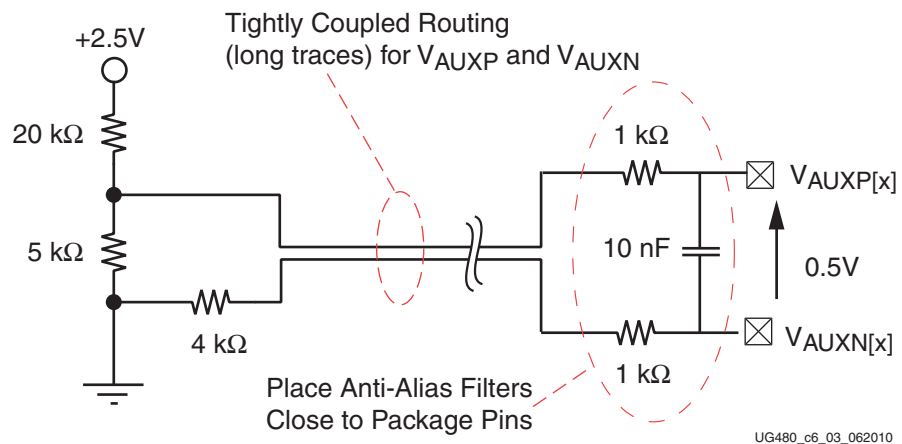


Figure 6-3: Voltage Attenuation

## Anti-Alias Filters

Also shown in Figure 6-3, 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. A discussion of aliasing in sampled systems is beyond the scope of this document. A good data converter reference book can provide more information on this topic.

## Over and Under Voltages

The input voltage can exceed VCCADC (1.8V) or go below GNDADC by as much as 100 mV without damage to the XADC. A current-limiting resistor of at least 100Ω should be placed in series with the analog inputs. 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, page 24 or Figure 2-2, page 25, depending on the analog input mode. Negative input voltages clip at zero code.

## PC Board Design Guidelines

Figure 6-4 and Figure 6-5 illustrate one possible way to address the requirements outlined in the previous sections. Figure 6-4 shows how, by staggering the vias with respect to the pads, north-south and east-west routing channels through the via field are created. These routing channels can be used to bring tightly coupled differential pairs into the center of the via field—even when using 5 mil tolerances.

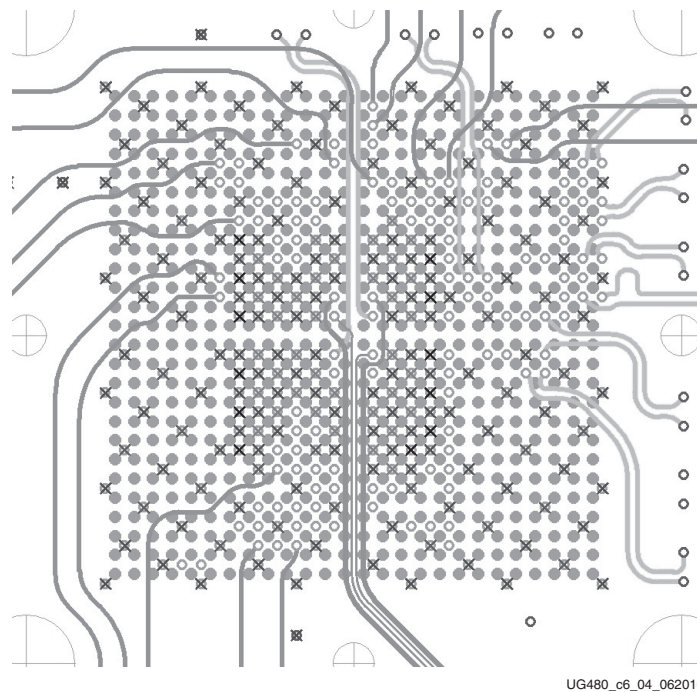
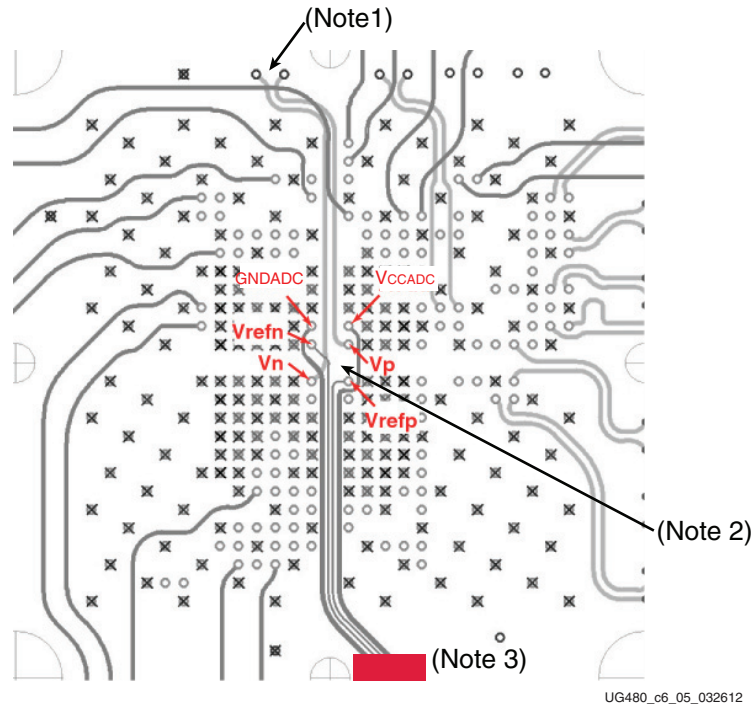


Figure 6-4: Routing Channels to Center of Array Created by Staggering Vias





**Figure 6-5: Reference Inputs ( $V_{REFP}$  and  $V_{REFN}$ ) should be Routed as Differential Pairs into the Center of the BGA**

Notes relevant to [Figure 6-5](#):

1. Place anti-alias filters for analog inputs close to the FPGA.
2. Place 100 nF decoupling for  $V_{REFP}$  and  $V_{CCADC}$  here.
3. Place external reference IC and any dedicated analog power supply regulation as close to the FPGA as possible.

In [Figure 6-5](#), the pads have been removed for clarity. The reference inputs ( $V_{REFP}$  and  $V_{REFN}$ ) are routed as a tightly coupled differential pair from an external 1.25V reference IC at the bottom edge of the FPGA (refer to [Figure 6-1](#) for the connections). The analog power supply and ground reference are also routed into the center of the BGA using traces. No power planes are required to supply a ground reference for the XADC. The analog supply and ground reference are connected to the external reference IC as shown in [Figure 6-1](#). In the PCB implementation shown, the supply and ground traces are routed on either side of the reference traces on the same signal layer and act as guards between the reference traces and any potential aggressors (e.g., clocks and switching I/Os). It is not a requirement that the supply and ground traces are routed on the same signal layer as shown, but they should be routed on an adjacent layer. The  $V_{REFN}$  and GNDADC traces should be connected at (or close to) the ground pin of the reference IC. The ferrite bead that connects the analog ground trace to system ground should also be placed close to the reference IC. [Figure 6-5](#) also shows the routing (from the top) of the dedicated analog input pair ( $V_P$  and  $V_N$ ). These inputs are also routed as a differential pair.

The external reference IC should be placed as close as possible to the FPGA to reduce the opportunities for coupled noise and to minimize any impedances in the reference traces. The staggered via field also allows the 100 nF decoupling on  $V_{REFP}$  and  $V_{CCADC}$  to be placed in the center of the array close to the package balls.  $V_{REFP}$  should be decoupled to  $V_{REFN}$  and  $V_{CCADC}$  to GNDADC near the package balls.

When using the on-chip reference, the layout of the PCB is greatly simplified. The  $V_{REFP}$  and  $V_{REFN}$  pins should be shorted to AGND locally at the package balls. The ferrite beads used to separate AGND and digital GND should be placed close to the XADC balls in the center of the array along with a 100 nF decoupling capacitor for  $V_{CCADC}$ .

These suggestions are only intended to guide a low-cost PC board implementation. If it is feasible to create an analog reference plane, then there is no issue with doing this and it helps improve the ADC performance. However, the reference inputs should still be routed as differential pairs as shown. See the schematics and layout for the KC705 and AC701 evaluation boards for an example of a practical layout implementation. For more information, see XAPP554, *XADC Layout Guidelines Application Note* [Ref 6].

## XADC Software Support

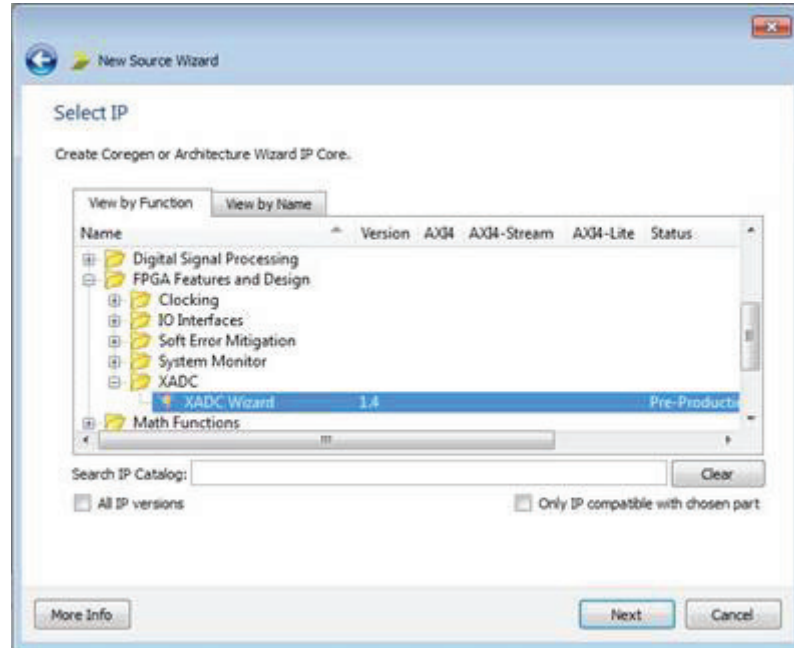
This section explains in more detail the example shown in [Example Instantiation, page 19](#). The Verilog and VHDL source code and test benches for this example can be found in the ZIP file `ug480_7Series_XADC.zip` that accompanies this user guide:

[Design file for UG480 ug480\\_7Series\\_XADC.zip](#)

In addition to instantiating a basic design, the use of the Analog Stimulus file as a means to introduce analog signals into the simulation is also explained. Finally, the output of a simple behavioral simulation of the design is shown. This ZIP file also includes a Vivado 2012.2 project. The project illustrates an instantiation and simulation of the XADC in Vivado.

## Using XADC Instantiation Wizard

One of the easiest methods to add the XADC to a design is to use the XADC instantiation wizard in the CORE Generator™ tool. In Vivado the XADC Wizard can be found under the IP Catalogue. The wizard is a GUI that allows designers to select the required block I/O and initialize the control registers for the required operation. [Figure 6-6](#) shows the location of the XADC wizard in the CORE Generator tool.



UG480\_c6\_06\_032712

Figure 6-6: XADC Wizard in the CORE Generator Tool

For more information on the XADC wizard, see UG772, *LogiCORE IP XADC Wizard User Guide* [Ref 5].

## Example Design Instantiation

The following HDL example sets up the XADC to monitor all the on-chip sensors, i.e., Temperature,  $V_{CCINT}$ ,  $V_{CCAUX}$ , and  $V_{CCBRAM}$  of a 7 series FPGA. See [Temperature Sensor, page 21](#) and [Power Supply Sensors, page 21](#). In addition, four auxiliary analog input channels are also monitored. The XADC is also set to automatically generate alarm outputs when the defined operating ranges for the FPGA supply voltages and temperature are exceeded (see [Automatic Alarms, page 59](#)). The XADC is operated in Continuous Sequence mode for this example (see [Sequencer Modes, page 52](#)). For clarity (and shorter simulations), the averaging function is disabled by the test bench. Averaging does not have any impact on the simulation results because an ideal model of the XADC is used. It is recommended that averaging is enabled when monitoring the on-chip sensors in a typical application to minimise any noise impacts. This is especially true if the automatic alarm functions are used. Averaging has been enabled for the on-chip sensors in the instantiation shown below.

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 \(40h to 42h\), page 39](#) for more information). Here is an instantiation in Verilog of the XADC example design:

```

`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company: Xilinx Inc.
//
// Create Date: 26/02/2012

```

```

//
// Module Name:    My_XADC
// Project Name:   XADC UG480 Example
// Target Devices: Kintex-7 325T
// Tool versions:  13.4
// Description:    Example XADC instantiation monitoring on chip sensors
// and auxiliary analog input channels 0 to 3
// Dependencies:
//
// Revision: 1.1
// Comments: See 7 series XADC User Guide UG480
// XADC Software Support Section
//
////////////////////////////////////

module My_XADC(
    DADDR_IN, // Address bus for the dynamic reconfiguration port
    DCLK_IN, // Clock input for the dynamic reconfiguration port
    DEN_IN, // Enable Signal for the dynamic reconfiguration port
    DI_IN, // Input data bus for the dynamic reconfiguration port
    DWE_IN, // Write Enable for the dynamic reconfiguration port
    RESET_IN, // Reset signal for the System Monitor control logic
    VAUXP0, // Auxiliary channel 0
    VAUXN0,
    VAUXP1, // Auxiliary channel 1
    VAUXN1,
    VAUXP2, // Auxiliary channel 2
    VAUXN2,
    VAUXP3, // Auxiliary channel 3
    VAUXN3,
    BUSY_OUT, // ADC Busy signal
    CHANNEL_OUT, // Channel Selection Outputs
    DO_OUT, // Output data bus for dynamic reconfiguration port
    DRDY_OUT, // Data ready signal for the dynamic reconfiguration port
    EOC_OUT, // End of Conversion Signal
    EOS_OUT, // End of Sequence Signal
    OT_OUT, // Over-Temperature alarm output
    VCCAUX_ALARM_OUT, // VCCAUX-sensor alarm output
    VCCINT_ALARM_OUT, // VCCINT-sensor alarm output
    USER_TEMP_ALARM_OUT, // Temperature-sensor alarm output
    VCCBRAM_ALARM_OUT, // VCCBRAM-sensor alarm output
    ALARM_OUT, // OR'ed output of all the Alarms
    VP_IN, // Dedicated and Hardwired Analog Input Pair
    VN_IN); // Not strictly required in the instantiation

    input [6:0] DADDR_IN;
    input DCLK_IN;
    input DEN_IN;
    input [15:0] DI_IN;
    input DWE_IN;
    input RESET_IN;
    input VAUXP0;
    input VAUXN0;
    input VAUXP1;
    input VAUXN1;
    input VAUXP2;
    input VAUXN2;
    input VAUXP3;

```

```
input VAUXN3;
input VP_IN;
input VN_IN;

output BUSY_OUT;
output [4:0] CHANNEL_OUT;
output [15:0] DO_OUT;
output DRDY_OUT;
output EOC_OUT;
output EOS_OUT;
output OT_OUT;
output VCCAUX_ALARM_OUT;
output VCCINT_ALARM_OUT;
output USER_TEMP_ALARM_OUT;
output VCCBRAM_ALARM_OUT;
output ALARM_OUT;

wire GND_BIT;
assign GND_BIT = 0;
wire [15:0] aux_channel_p;
wire [15:0] aux_channel_n;
wire [7:0] alm_int;

assign ALARM_OUT = alm_int[7];
assign VCCBRAM_ALARM_OUT = alm_int[3];
assign VCCAUX_ALARM_OUT = alm_int[2];
assign VCCINT_ALARM_OUT = alm_int[1];
assign USER_TEMP_ALARM_OUT = alm_int[0];

assign aux_channel_p[0] = VAUXP0;// Connection to top level enables these as Analog inputs
assign aux_channel_n[0] = VAUXN0;// No UCF setting are required

assign aux_channel_p[1] = VAUXP1;// Connection to top level enables these as Analog inputs
assign aux_channel_n[1] = VAUXN1;// No UCF setting are required

assign aux_channel_p[2] = VAUXP2;// Connection to top level enables these as Analog inputs
assign aux_channel_n[2] = VAUXN2;// No UCF setting are required

assign aux_channel_p[3] = VAUXP3;// Connection to top level enables these as Analog inputs
assign aux_channel_n[3] = VAUXN3;// No UCF setting are required

assign aux_channel_p[4] = 1'b0;// Not connected to top level
assign aux_channel_n[4] = 1'b0;// Available as regular digital IO

assign aux_channel_p[5] = 1'b0;// Not connected to top level
assign aux_channel_n[5] = 1'b0;

assign aux_channel_p[6] = 1'b0;// Not connected to top level
assign aux_channel_n[6] = 1'b0;

assign aux_channel_p[7] = 1'b0;// Not connected to top level
assign aux_channel_n[7] = 1'b0;

assign aux_channel_p[8] = 1'b0;// Not connected to top level
assign aux_channel_n[8] = 1'b0;

assign aux_channel_p[9] = 1'b0;// Not connected to top level
assign aux_channel_n[9] = 1'b0;
```

```

assign aux_channel_p[10] = 1'b0;// Not connected to top level
assign aux_channel_n[10] = 1'b0;

assign aux_channel_p[11] = 1'b0;// Not connected to top level
assign aux_channel_n[11] = 1'b0;

assign aux_channel_p[12] = 1'b0;// Not connected to top level
assign aux_channel_n[12] = 1'b0;

assign aux_channel_p[13] = 1'b0;// Not connected to top level
assign aux_channel_n[13] = 1'b0;

assign aux_channel_p[14] = 1'b0;// Not connected to top level
assign aux_channel_n[14] = 1'b0;

assign aux_channel_p[15] = 1'b0;// Not connected to top level
assign aux_channel_n[15] = 1'b0;

XADC #( // Initializing the XADC Control Registers
.INIT_40(16'h9000)// Calibration coefficient averaging disabled
    // averaging of 16 selected for external channels
.INIT_41(16'h2ef0)// Continuous Sequencer Mode, Disable unused ALMs, Enable calibration
.INIT_42(16'h0400)// Set DCLK divider to 4, ADC = 500Ksps, DCLK = 50MHz
.INIT_48(16'h4701)// Sequencer channel - enable Temp sensor, VCCINT, VCCAUX, VCCBRAM, and
calibration
.INIT_49(16'h000f)// Sequencer channel - enable aux analog channels 0 - 3
.INIT_4A(16'h4700)// Averaging enabled for Temp sensor, VCCINT, VCCAUX, VCCBRAM
.INIT_4B(16'h0000)// No averaging on external channels
.INIT_4C(16'h0000)// Sequencer Bipolar selection
.INIT_4D(16'h0000)// Sequencer Bipolar selection
.INIT_4E(16'h0000)// Sequencer Acq time selection
.INIT_4F(16'h0000)// Sequencer Acq time selection
.INIT_50(16'hb5ed)// Temp upper alarm trigger 85°C
.INIT_51(16'h5999)// Vccint upper alarm limit 1.05V
.INIT_52(16'hA147)// Vccaux upper alarm limit 1.89V
.INIT_53(16'hdddd)// OT upper alarm limit 125°C
.INIT_54(16'ha93a)// Temp lower alarm reset 60°C
.INIT_55(16'h5111)// Vccint lower alarm limit 0.95V
.INIT_56(16'h91Eb)// Vccaux lower alarm limit 1.71V
.INIT_57(16'hae4e)// OT lower alarm reset 70°C
.INIT_58(16'h5999)// VCCBRAM upper alarm limit 1.05V
.INIT_5C(16'h5111)// VCCBRAM lower alarm limit 0.95V
.SIM_MONITOR_FILE("sensor_input.txt")// Analog Stimulus file. Analog input values for
simulation
)

XADC_INST (// Connect up instance IO. See UG480 for port descriptions
.CONVST(GND_BIT)// not used
.CONVSTCLK(GND_BIT), // not used
.DADDR(DADDR_IN[6:0]),
.DCLK(DCLK_IN),
.DEN(DEN_IN),
.DI(DI_IN[15:0]),
.DWE(DWE_IN),
.RESET(RESET_IN),
.VAUXN(aux_channel_n[15:0]),
.VAUXP(aux_channel_p[15:0]),
.ALM(alm_int),

```

```

.BUSY(BUSY_OUT),
.CHANNEL(CHANNEL_OUT[4:0]),
.DO(DO_OUT[15:0]),
.DRDY(DRDY_OUT),
.EOC(EOC_OUT),
.EOS(EOS_OUT),
.JTAGBUSY(), // not used
.JTAGLOCKED(), // not used
.JTAGMODIFIED(), // not used
.OT(OT_OUT),
.MUXADDR(), // not used
.VP(VP_IN),
.VN(VN_IN)
);

endmodule

```

## Example Design Test Bench

The following is a simple test bench that sets up a DCLK of 50 MHz and then reads the Status registers (conversion results) at the end of a sequence when EOS goes High. The test bench also does a write to the DRP after the reset to disable the averaging on the sensor channels. The test bench monitors the alarm outputs so alarm events can be seen. Analog signals are not introduced into the simulation using the test bench. Analog signals are read from a file by the simulation model. The SIM\_MONITOR\_FILE attribute used in the XADC instantiation points the model to the location of this file known as the *Analog Stimulus file*. The Analog Stimulus file used for this example is shown here:

```

// It is used to inject analog signals (e.g., volts) input the simulation
// Time is in nanoseconds & inputs are volts or degrees C for temperature
// In this example the VCCAUX supply moves outside the 1.89V upper alarm limit at 27us
// An alarm is generate when the VCCAUX channel is sampled and converted by the ADC

```

TIME	VAUXP[0]	VAUXN[0]	VAUXP[1]	VAUXN[1]	VAUXP[2]	VAUXN[2]	VAUXP[3]	VAUXN[3]	Temp	VCCINT	VCCAUX	VBRAM
00000	0.005	0.0	0.001	0.0	0.5	0.0	0.1	0.0	25	1.0	1.8	1.0
27000	0.020	0.0	0.022	0.0	0.49	0.0	0.2	0.0	85	1.05	1.90	1.05
50000	0.049	0.0	0.051	0.0	0.51	0.0	0.5	0.0	50	0.95	1.71	0.95
73000	0.034	0.0	0.067	0.0	0.53	0.0	0.0	0.0	60	1.00	1.8	1.0

The format of the Analog Stimulus file is based on space- or tab-delimited data and can be created in a spreadsheet, for example. Many tools such as SPICE simulators or even equipment such as oscilloscopes export comma-separated value (CSV) formats, which are easily 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 every time stamp added to the first column, a corresponding value is added to the other columns. Only the required analog input channel columns need to be listed. In this example, only the on-chip sensors and auxiliary channels 0 to 3 are listed in the Analog Stimulus file. In this stimulus file, the voltage on  $V_{CCAUX}$  moves from nominal supply (1.8V) to 1.9V at 27  $\mu$ s after the start of the simulation. The  $V_{CCAUX\_ALARM}$  output can be seen to go active High shortly after this event when the  $V_{CCAUX}$  supply is measured by the ADC during the sequence (see [Figure 6-7, page 85](#)). The upper alarm threshold for  $V_{CCAUX}$  has been set at 1.89V in the instantiation above.

```

`timescale 1ns / 1ps

/////////////////////////////////////////////////////////////////
// Company:Xilinx Inc.
//
// Create Date:26/02/2012
//
// Module Name:My_XADC_tb
// Project Name: XADC UG Example Test Bench
// Target Devices:Kintex-7 325T
// Tool versions:13.4
// Description: Test bench for XADC UG480 Example design
//
// Dependencies:
//
// Revision: 1.1
// Comments:See 7 series XADC User Guide
//          This test bench contains some useful tasks & functions:
//          - Writing & Reading the DRP
//          - Converter ADC output to degrees C and volts
//          Tested with iSim 13.4
//
/////////////////////////////////////////////////////////////////

module My_XADC_tb;

    // Inputs
    reg VAUXP0;
    reg VAUXN0;
    reg VAUXP1;
    reg VAUXN1;
    reg VAUXP2;
    reg VAUXN2;
    reg VAUXP3;
    reg VAUXN3;

    reg [6:0] DADDR;
    reg DCLK;
    reg DEN, DWE;
    reg RESET;
    reg [15:0] DI;

    // Outputs
    wire [15:0] DO;
    wire DRDY;
    wire EOS;
    wire BUSY;
    wire [4:0] CHANNEL;
    wire VCCAUX_ALARM;
    wire VCCINT_ALARM;
    wire USER_TEMP_ALARM;
    wire VCCBRAM_ALARM;
    wire ALARM;

    // Testbench variables

    wire GND_BIT;

```



```

assign GND_BIT = 0;
reg [15:0] reg_contents;

real temperature;// display real numbers for ADC output
real Vccint;
real Vccaux;
real Vccbram;

real Vaux0;
real Vaux1;
real Vaux2;
real Vaux3;

reg [15:0] temp_output; //Actual 16-bit ADC output values
reg [15:0] Vccint_output;
reg [15:0] Vccaux_output;
reg [15:0] Vccbram_output;

reg [15:0] Aux0_output;
reg [15:0] Aux1_output;
reg [15:0] Aux2_output;
reg [15:0] Aux3_output;

// Instantiate the Unit Under Test (UUT)

My_XADC uut (
    .VAUXP0(VAUXP0),
    .VAUXN0(VAUXN0),
    .VAUXP1(VAUXP1),
    .VAUXN1(VAUXN1),
    .VAUXP2(VAUXP2),
    .VAUXN2(VAUXN2),
    .VAUXP3(VAUXP3),
    .VAUXN3(VAUXN3),
    .DADDR_IN(DADDR),
    .DCLK_IN(DCLK),
    .DEN_IN(DEN),
    .DWE_IN(DWE),
    .RESET_IN(RESET),
    .DO_OUT(DO),
    .DI_IN(DI),
    .DRDY_OUT(DRDY),
    .EOS_OUT(EOS),
    .BUSY_OUT(BUSY),
    .CHANNEL_OUT(CHANNEL),
    .VCCAUX_ALARM_OUT(VCCAUX_ALARM),
    .VCCINT_ALARM_OUT(VCCINT_ALARM),
    .USER_TEMP_ALARM_OUT(USER_TEMP_ALARM),
    .VCCBRAM_ALARM_OUT(VCCBRAM_ALARM),
    .ALARM_OUT(ALARM)
);

// Clock Generator
`define DCLK_MHZ50

```

```

always #(1000.0 / (2*`DCLK_MHZ)) DCLK= ~DCLK;

initial begin

// Initialize Inputs
DI = 16'h0000;
DADDR = 0;
DCLK = 0;
DEN = 0;
DWE = 0;
RESET = 1;

// Wait 4 us for global reset to finish
repeat(200) @(posedge DCLK);
RESET = 0;

// Example of DRP write to Control Registers to make a change
// Change attributes initialized in the original instantiation (reg 40h)
// Reading the Configuration Reg #0(40h) and disabling the averaging for quicker simulation

drp_read(7'h40); // Read register 40h
reg_contents = DO;
assign reg_contents = (reg_contents & 16'h0FFF); // Set AVG bits to zero
drp_write(7'h40, reg_contents); // Write updated configuration to register 40h

end

// Read back results from Status Registers via DRP when EOS pulses
// EOS indicates channel sequence is complete.

always@(posedge EOS) begin

drp_read(0); // Read the result for temp
temp_output = DO; // Save the result for temp in hex
temperature = bits2temp(DO); // Save the result for temp as "degrees C"

drp_read(1); // Read the result for Vccint
Vccint_output = DO; // Save the result for Vccint
Vccint = bits2supply(DO); // Save the result for Vccint as a "volts"

drp_read(2); // Read the result for Vccaux
Vccaux_output = DO; // Save the result for Vccaux
Vccaux = bits2supply(DO); // Save the result for Vccaux as a "volts"

drp_read(6); // Read the result for Vccbram
Vccbram_output = DO; // Save the result for Vccbram
Vccbram = bits2supply(DO); // Save the result for Vccbram as a "volts"

drp_read(16); // Read the result for Aux0
Aux0_output = DO; // Save the result for Aux0
Vaux0 = bits2univolts(DO); // Save the result for Vaux0 as a "volts"

drp_read(17); // Read the result for Aux1
Aux1_output = DO; // Save the result for Aux1
Vaux1 = bits2univolts(DO); // Save the result for Vaux1 as a "volts"

```

```

drp_read(18); // Read the result for Aux2
Aux2_output = DO; // Save the result for Aux2
Vaux2 = bits2univolts(DO); // Save the result for Vaux2 as a "volts"

drp_read(19); // Read the result for Aux2
Aux3_output = DO; // Save the result for Aux2
Vaux3 = bits2univolts(DO); // Save the result for Vaux3 as a "volts"

end

// Some Helper tasks
// Tasks to do DRP read & write
// Tasks to convert ADC output to Degrees C and Volts

// Note: this task assumes there are variables defined: DCLK, DADDR, DEN, DWE, and DRDY
task drp_read;
input [6:0] addr;
begin
    @(negedge DCLK);
    DEN = 1'b1;
    DWE = 1'b0;
    DADDR = addr;
    @(negedge DCLK); // DEN only pulsed for one clock cycle
    DEN = 0;
    wait(DRDY==1); // Wait for the drdy output to go hi before exiting.
    @(negedge DCLK);
end
endtask

// Note: this task assumes there are variables defined: DCLK, DADDR, DEN, DWE, and DRDY
task drp_write;
input [6:0] addr;
input [15:0] data;
begin
    @(negedge DCLK);
    DI = data;
    DEN = 1'b1;
    DWE = 1'b1;
    DADDR = addr;
    @(negedge DCLK); // DEN only pulsed for one clock cycle
    DEN = 0;
    DWE = 0;
    wait(DRDY==1); // Wait for the drdy output to go hi before exiting.
    @(negedge DCLK);
end
endtask

// Implements the temperature transfer function
// implement the transfer function from temperature in C to a 16-bit number

function [15:0] temp2bits;
input real temperature;

```

```

real r_temp2bits;
begin
  r_temp2bits = (temperature + 273.15)*65535.0*0.00198421639;
  temp2bits = r_temp2bits;// 'cast' the output to a 16 bit word
end
endfunction

// Inverse of the function above - convert bits to a temperature value
function real bits2temp;
input [15:0] bits;
begin
  bits2temp = ((bits/65536.0)/0.00198421639) - 273.15;
end
endfunction

// Implements the supply sensor transfer function
// implement the transfer function from a 3V range to a 16-bit number
function [15:0] supply2bits;
input real supply_val;
real r_supply2bits;
begin
  r_supply2bits = supply_val/3.0*65535.0;
  supply2bits = r_supply2bits;// 'cast' the output to a 16 bit word
end
endfunction

// Implements the inverse supply sensor transfer function
function real bits2supply;
input [15:0] bits;
begin
  bits2supply = ((bits/65536.0)* 3);
end
endfunction

// Implements the ADC unipolar transfer function
// implement the transfer function from a 3V range to a 16-bit number
function [15:0] univolts2bits;
input real supply_val;
real r_supply2bits;
begin
  r_supply2bits = supply_val/65535.0;
  univolts2bits = r_supply2bits; // 'cast' the output to a 16 bit word
end
endfunction

// Implements the ADC unipolar inverse transfer function
function real bits2univolts;
input [15:0] bits;

```

```

begin
  bits2univolts = ((bits/65536.0));
end
endfunction

endmodule
    
```

### Simulation Output

The simulation output shown in Figure 6-7 shows two passes through the user-defined sequence in continuous sampling mode. The channels monitored in the sequence can be seen by looking at the CHANNEL[4:0] bus. The sequence is 8, 0, 1, 2, 3, 6, 10, 11, 12, and 13, which corresponds to Calibration, Temperature, V<sub>CCINT</sub>, V<sub>CCAUX</sub>, V<sub>CCBRAM</sub>, Aux0, Aux1, Aux2, and Aux3. Then the sequence repeats. Note how the calibration channel takes longer to complete than the other channels. This is because the calibration routine involves three conversions (measurements) using the ADCs.

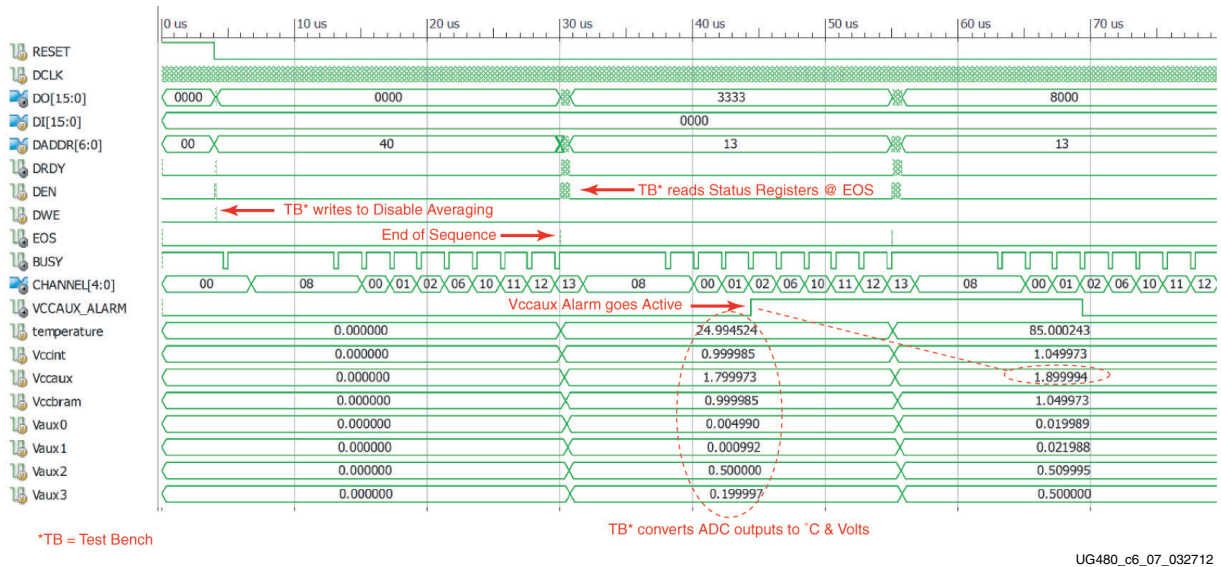


Figure 6-7: XADC Simulation Output (iSim)

After RESET goes low, the test bench carries out a DRP write to register 40h (Configuration Register 0) to set the AVG1 and AVG0 bits to 00 (see Figure 6-8). This disables the averaging functionality for simulation.

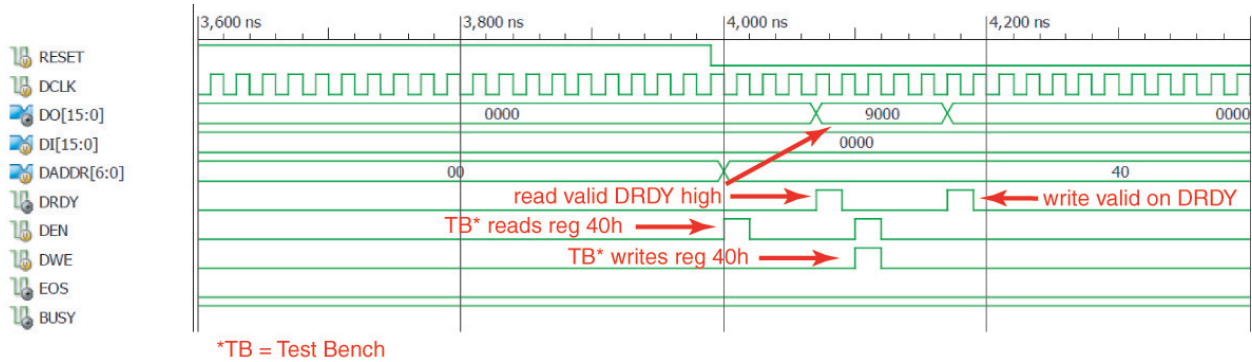


Figure 6-8: DRP Write at 4 μs

When the EOS signal pulses high for one DCLK period at the end of a sequence (at approximately 30 μs) the test bench reads the Status registers (see Figure 6-9). Note that the simulation model uses the full 16 ADC conversion result because it is an ideal model of the ADC. Thus for example, the result for the  $V_{CCINT}$  measurement is 5555h, which corresponds to 1V. In Figure 2-9, page 32 (Ideal Power Supply Transfer Function) 1V is shown as 555h. This is a 12-bit MSB justified result. However, the 4 LSBs of the Status register also contain data that would be 5h if the ADC was an ideal 16-bit ADC.

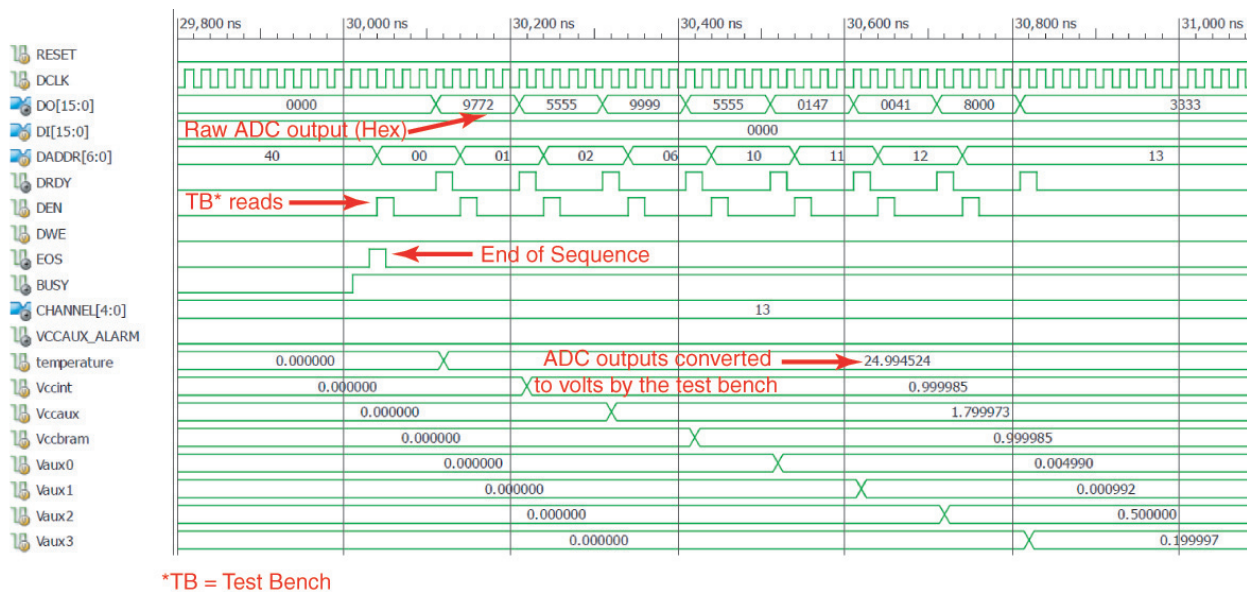


Figure 6-9: DRP Read of Status Register at EOS

Note how the  $V_{CCAUX\_ALARM}$  output goes high during the second pass through the sequence (Figure 6-7). The alarm goes 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 goes high. The  $V_{CCAUX}$  is ~1.9V (as set in the stimulus file) and is greater than the 1.89V limit set when the XADC was instantiated in the design which activates the alarm.