

# LogiCORE IP AXI XADC (v1.00.a)

## *Product Guide*

PG019 October 19, 2011

# Table of Contents

---

## Chapter 1: Overview

|                                     |   |
|-------------------------------------|---|
| Operating System Requirements ..... | 7 |
| Feature Summary .....               | 7 |
| Applications .....                  | 7 |
| Licensing .....                     | 7 |
| Resource Utilization.....           | 8 |

## Chapter 2: Core Interfaces and Register Space

|  |    |
|--|----|
| Port Descriptions.....                                 | 9  |
| AXI XADC Register Descriptions.....                    | 11 |
| AXI XADC Local Register Grouping.....                  | 16 |
| AXI XADC Interrupt Controller Register Grouping .....  | 19 |
| XADC Hard Macro Register (DRP Register) Grouping ..... | 23 |

## Chapter 3: Customizing and Generating the Core

|                        |    |
|------------------------|----|
| Output Generation..... | 25 |
|------------------------|----|

## Chapter 4: Designing with the Core

|                                     |    |
|-------------------------------------|----|
| AXI XADC Design Parameters .....    | 26 |
| Parameter - Port Dependencies ..... | 27 |
| Clocking.....                       | 27 |
| Resets.....                         | 27 |

## Chapter 5: Constraining the Core

|                         |    |
|-------------------------|----|
| Design Constraints..... | 28 |
|-------------------------|----|

## Chapter 6: Detailed Example Design

## Appendix 7: Additional Resources

|                            |    |
|----------------------------|----|
| Xilinx Resources .....     | 30 |
| List of Acronyms.....      | 30 |
| Solution Centers .....     | 31 |
| References .....           | 31 |
| Technical Support.....     | 31 |
| Ordering Information ..... | 32 |
| Revision History .....     | 32 |
| Notice of Disclaimer ..... | 32 |

## Introduction

The Advanced eXtensible Interface (AXI) Xilinx® Analog-to-Digital Converter (XADC) Intellectual Property (IP) Core is a 32-bit slave peripheral that connects to the AXI4 and provides the controller interface for System Monitor XADC hard macro on the Virtex®-7 and Kintex™-7 Field Programmable Gate Arrays (FPGAs). This document describes the specifications for the AXI XADC IP Core. It is assumed that the user is familiar with the XADC hard macro. For information on the XADC hard macro, see the [XADC Hard Macro](#) section.

## Features

- AXI4-Lite interface is based on the AXI4 specification.
- Connects as a 32-bit AXI4-Lite slave
- Supports two 12-bit, 1 Mega-Samples Per Second (MSPS) Analog-to-Digital Converters (ADC)
- Supports optional interrupt request generation

| LogiCORE IP Facts Table   |  |
|---|--|
| Core Specifics  |  |
| Supported Device Family <sup>(1)</sup>  | Virtex-7, Kintex-7                         |
| Supported User Interfaces   | AXI4-Lite                                  |
| Resources   | See <a href="#">Table 1-1</a> .            |
| Provided with Core  |  |
| Design Files  | VHSIC Hardware Description Language (VHDL) |
| Example Design  | VHDL                                       |
| Test Bench  | Not Provided                               |
| Constraints File  | No t Applicable                            |
| Simulation Model  | Not Provided                               |
| Tested Design Tools   |  |
| Design Entry Tools  | Xilinx Platform Studio (XPS) 13.3          |
| Simulation <sup>(2)</sup>   | ModelSim                                   |
| Synthesis Tools <sup>(2)</sup>  | Xilinx Synthesis Technology (XST).         |
| Support   |  |
| Provided by Xilinx @ <a href="http://www.xilinx.com/support">www.xilinx.com/support</a> |  |

1. For a complete list of supported derivative devices, see the [IDS Embedded Edition Derivative Device Support](#).
2. For the supported versions of the tools, see the [ISE Design Suite 13: Release Notes Guide](#).

## Overview

The top-level block diagram for the AXI XADC IP Core is shown in Figure 1-1.

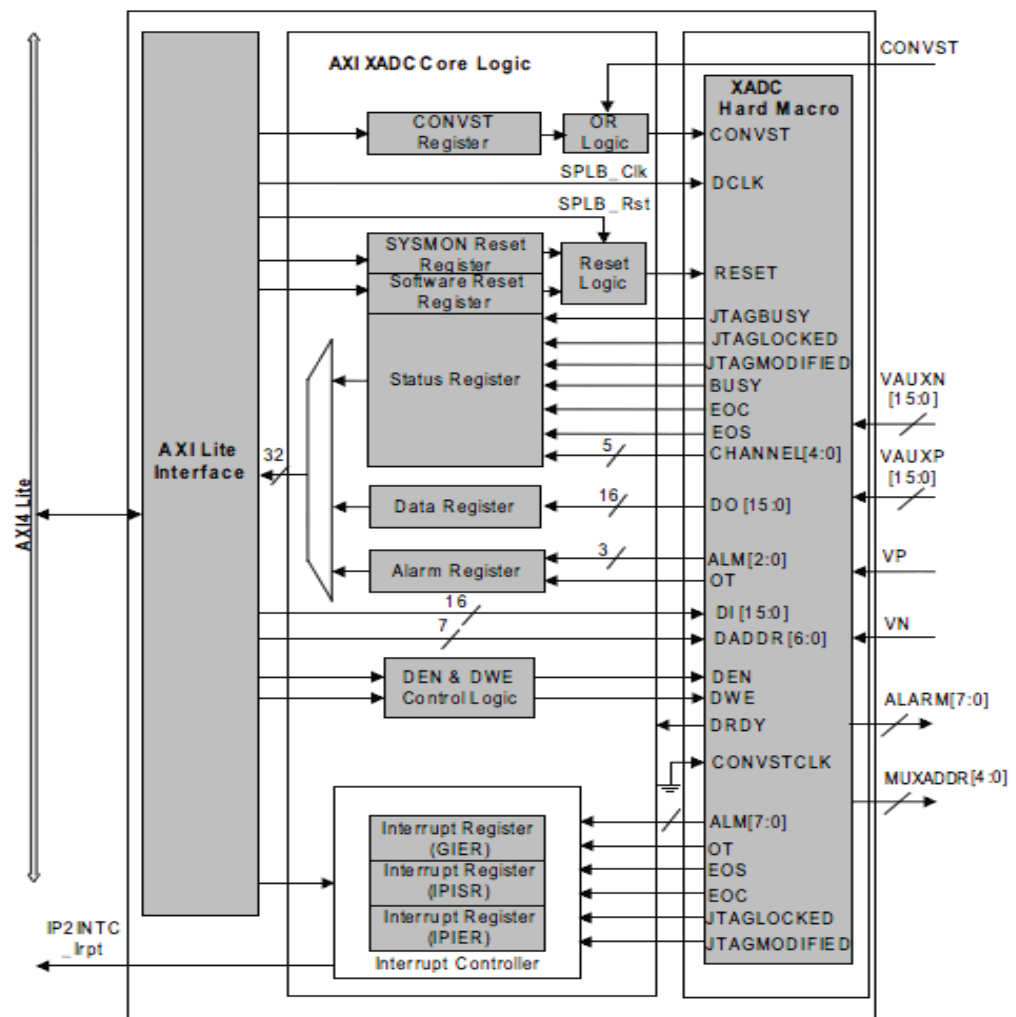


Figure 1-1: Block Diagram of AXI XADC IP Core

The AXI XADC IP Core consists of following major blocks.

- AXI4-Lite Interface Module
- XADC Core Logic
- XADC Hard Macro

## AXI4-Lite Interface Module

The AXI4-Lite Interface Module provides the AXI4-Lite interface to the AXI4. Read and write transactions at the AXI4 are translated into equivalent XADC core logic and XADC hard macro transactions. The register interfaces of the XADC Core Logic connect to the AXI4 Interface Module. The AXI4 Interface Module also provides an address decoding service.

## XADC Core Logic

The XADC core logic provides necessary address decoding logic, control signal generation and the interface between AXI4-Lite and XADC hard macro. The read/write requests along with the address and data (in case of write) from the AXI4 Interface Module are transferred to either Dynamic Reconfiguration Port (DRP) registers of the XADC hard macro or local registers in the IP along with the necessary control signals like DEN and DWE.

The XADC core logic supports including/excluding the Interrupt Controller based on generic `C_INCLUDE_INTR`. If `C_INCLUDE_INTR = 1` then the Interrupt Controller is included in the design.

## XADC Hard Macro

The XADC hard macro can be accessed via both Joint Test Action Group (JTAG) TAP (Test Access Port) and the AXI XADC IP Core. When simultaneous access of the XADC hard macro occurs, JTAGLOCKED port can be asserted high by JTAG TAP. In this scenario the AXI XADC IP core is not allowed to do any read/write access from/to DRP or FPGA logic. When JTAGLOCKED port is again deasserted through JTAG TAP, AXI XADC IP core can do read/write operation from/to DRP.

This functionality is especially useful in applications where the user is configuring DRP through JTAG TAP and does not want the FPGA logic (AXI XADC IP core) to alter the configuration. The user can make JTAGLOCKED = '1' through JTAG TAP which blocks any read/write transactions from/to DRP through the FPGA logic and thus ensures a non-destructive access through JTAG TAP.XADC Hard Macro.

The XADC hard macro is present in every 7 series family of FPGAs. The block diagram for the XADC hard macro is shown in [Figure 1-2](#).

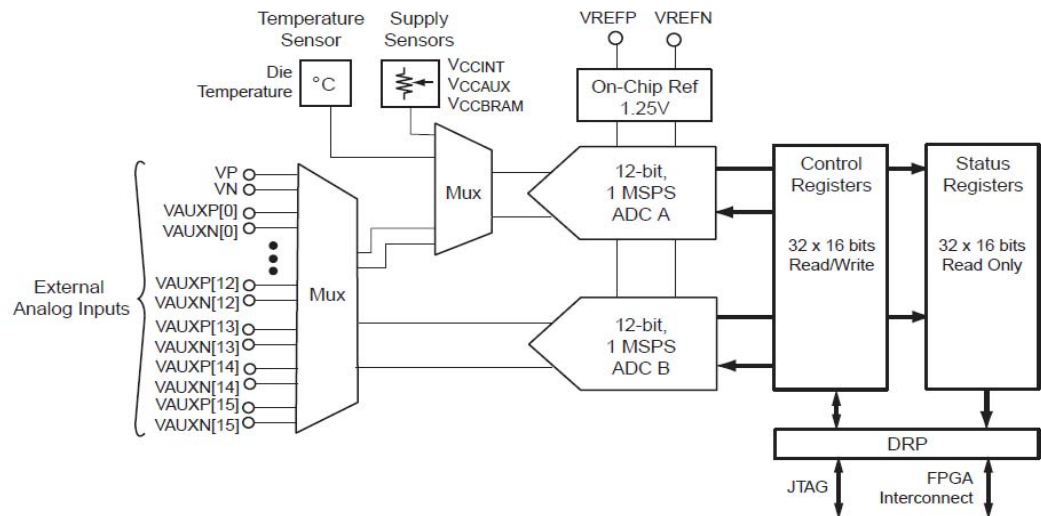


Figure 1-2: Block Diagram of XADC Hard Macro

The AXI XADC IP core is built around a dedicated XADC hard macro on the device family. It uses two 12-bit, 1 MSPS ADCs internally for conversion of various analog data. The AXI XADC IP Core is used to measure die temperature and voltage. Additionally, the AXI XADC IP Core provides analog-to-digital conversion of up to 16 external channels in simultaneous sequencer mode. From a user point of view, the core is defined as an AXI XADC that can operate to monitor on-device voltage and temperature and/or external analog voltages.

The XADC hard macro consists of the Register File Interface (RFI) which in turn consists of Status and Control registers. Status registers are read-only and contain the results of analog-to-digital conversion of the on-device sensors and external channels. The status registers also store the maximum and minimum temperature and VCCAUX/VCCINT voltages.

The control registers are used to configure the XADC hard macro operation. XADC hard macro functionality (ADC operating modes, Channel Sequencer, and Alarm limits) is controlled through these registers. The first three registers in the control register block are also called as Configuration Registers and are used to configure XADC hard macro operating modes. In addition to the RFI of the hard macro, the AXI XADC IP core consists of a set of local registers and optional interrupt registers.

The XADC hard macro provides channel sequencing, averaging and filtering functions. Many of the 16-bit registers are not defined in the XADC hard macro RFI. Accessing a location that is undefined returns an undefined value.

In the XADC hard macro, in individual ADC mode, a channel sequencer allows the user to specify the channels monitored but the sequence order is fixed. Users can specify an averaging filter to reduce noise. There are programmable alarm thresholds for the on-device sensors and if an on-device temperature or voltage is enabled outside the specified limit an alarm is activated. If simultaneous sequencer mode is enabled, both the ADCs will monitor one channel each and the results are provided at the end of conversion. The conversion time for both the ADCs is same.

Structurally, the AXI XADC IP core consists of the XADC hard macro, the AXI4 Interface Module, Optional Interrupt Source Controller Module, Soft Reset Module, XADC Reset Register and additional logic to interface to the core. The Soft Reset Module provides a way for resetting the entire IP without disturbing the entire system. The XADC Reset Register is provided to reset the XADC hard macro only.

All read and write operations to the configuration and limit registers are synchronized to DCLK (DCLK input of XADC hard macro is connected to (AXI\_Clk). The XADC hard macro has an internal clock divider that divides DCLK by any integer ranging from 2–255 to generate ADCCLK. ADCCLK is an internal clock used by the ADC. Because an internal clock divider is provided, DCLK frequency can be in the range of 2 MHz to 200 MHz. See the 7 series FPGA data sheets for the maximum operating frequency of XADC.

The XADC hard macro operates either in event-driven or continuous sampling mode. In event-driven sampling mode, the conversion process is initiated on the rising edge of CONVST. The AXI XADC supports this operation by providing a rising edge signal on the external CONVST port or by writing into the CONVST Register. In continuous sampling mode, the ADC continues to carry out a conversion on the selected analog inputs as long as the ADCCLK (DCLK) is present.

## Operating System Requirements

For a list of System Requirements, see [ISE Design Suite 13: Release Notes Guide](#).

## Feature Summary

- AXI4-Lite interface is based on the AXI4 specification.
- Connects as a 32-bit AXI4-Lite slave
- Supports two 12-bit, 1 Mega-Samples Per Second (MSPS) Analog-to-Digital Converters (ADC)
- Supports on-chip monitoring of supply voltages and temperature
- Supports simultaneous sequencer mode when both the ADCs are in use
- Supports one dedicated high bandwidth differential analog-input pair and 16 auxiliary low bandwidth differential analog-input pairs
- Supports automatic alarms based on user-defined limits
- Supports optional interrupt request generation

## Applications

AXI XDAC is ideally suited for high-volume applications such as multi-function printers, digital SLR cameras, Motor Control, Power Conversion, Touch/Gesture based HMI, anti-tamper security and system management.

## Licensing

This Xilinx® LogiCORE™ IP module is provided at no additional cost with the Xilinx ISE® Design Suite Embedded Edition software under the terms of the [Xilinx End User License](#). The core is generated using the Xilinx ISE Embedded Edition software (EDK).

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx [Intellectual Property page](#). For information on pricing and availability of other Xilinx LogiCORE modules and software, contact your local Xilinx [sales representative](#).

## Resource Utilization

Because the AXI XADC IP core can be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the AXI XADC IP core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI XADC IP core design varies from the results reported here.

The AXI XADC IP Core resource utilization for various parameter combinations measured with the 7 series FPGAs as the target device are detailed in [Table 1-1](#).

**Table 1-1: Performance and Resource Utilization Benchmarks on the 7 Series FPGAs**

| Parameter Values<br>(Other parameters at default values) | Device Resources |      | Performance            |
|--|------------------|------|------------------------|
|  | Slice Flip-Flops | LUTs | F <sub>Max</sub> (MHz) |
| 0  | 120              | 148  | 100                    |
| 1  | 200              | 239  | 100                    |



## Core Interfaces and Register Space

This chapter provides detailed descriptions for each interface. In addition, detailed information about configuration and control registers is included.

### Port Descriptions

The AXI XADC I/O signals are listed and described in [Table 2-1](#).

**Table 2-1: AXI XADC I/O Signal Descriptions**

| Port                                     | Signal Name                                  | Interface | I/O | Initial State | Description   |
|--|--|-----------|-----|---------------|---|
| <b>AXI Global System Signals</b>         |  |           |     |               |   |
| P1                                       | S_AXI_ACLK                                   | AXI       | I   | -             | AXI Clock   |
| P2                                       | S_AXI_ARESETN                                | AXI       | I   | -             | AXI Reset, active LOW   |
| <b>AXI Write Address Channel Signals</b> |  |           |     |               |   |
| P3                                       | S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0]   | AXI       | I   | -             | AXI Write address. The write address bus gives the address of the write transaction.                                    |
| P4                                       | S_AXI_AWVALID                                | AXI       | I   | -             | Write address valid. This signal indicates that a valid write address and control information are available.            |
| P5                                       | S_AXI_AWREADY                                | AXI       | O   | 0             | Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals. |
| <b>AXI Write Channel Signals</b>         |  |           |     |               |   |
| P6                                       | S_AXI_WDATA[(C_S_AXI_DATA_WIDTH - 1) : 0]    | AXI       | I   | -             | Write data  |
| P7                                       | S_AXI_WSTB[((C_S_AXI_DATA_WIDTH/8) - 1) : 0] | AXI       | I   | -             | Write strobes. This signal indicates which byte lanes to update in memory.  |
| P8                                       | S_AXI_WVALID                                 | AXI       | I   | -             | Write valid. This signal indicates that valid write data and strobes are available.                                     |
| P9                                       | S_AXI_WREADY                                 | AXI       | O   | 0             | Write ready. This signal indicates that the slave can accept the write data.  |

Table 2-1: AXI XADC I/O Signal Descriptions (Cont'd)

| Port                                      | Signal Name                                | Interface | I/O | Initial State | Description  |
|---|--|-----------|-----|---------------|--|
| <b>AXI Write Response Channel Signals</b> |  |           |     |               |  |
| P10                                       | S_AXI_BRESP[1 : 0]                         | AXI       | O   | 0             | Write response. This signal indicates the status of the write transaction<br>"00" - OKAY (normal response)<br>"10" - SLVERR (error response)<br>"11" - DECERR (not issued by core)                 |
| P11                                       | S_AXI_BVALID                               | AXI       | O   | 0             | Write response valid. This signal indicates that a valid write response is available.  |
| P12                                       | S_AXI_BREADY                               | AXI       | I   | -             | Response ready. This signal indicates that the master can accept the response information.   |
| <b>AXI Read Address Channel Signals</b>   |  |           |     |               |  |
| P13                                       | S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0] | AXI       | I   | -             | Read address. The read address bus gives the address of a read transaction.  |
| P14                                       | S_AXI_ARVALID                              | AXI       | I   | -             | Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and remains stable until the address acknowledgement signal, S_AXI_ARREADY, is high.  |
| P15                                       | S_AXI_ARREADY                              | AXI       | O   | 1             | Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals.   |
| <b>AXI Read Data Channel Signals</b>      |  |           |     |               |  |
| P16                                       | S_AXI_RDATA[(C_S_AXI_DATA_WIDTH - 1) : 0]  | AXI       | O   | 0             | Read data  |
| P17                                       | S_AXI_RRESP[1 : 0]                         | AXI       | O   | 0             | Read response. This signal indicates the status of the read transfer.<br>"00" - OKAY (normal response)<br>"10" - SLVERR (error condition)<br>"11" - DECERR (not issued by core)                    |
| P18                                       | S_AXI_RVALID                               | AXI       | O   | 0             | Read valid. This signal indicates that the required read data is available and the read transfer can complete.   |
| P19                                       | S_AXI_RREADY                               | AXI       | I   | -             | Read ready. This signal indicates that the master can accept the read data and response information.   |
| <b>AXI XADC IP Core Interface Signals</b> |  |           |     |               |  |
| P20                                       | VAUXP[15 : 0]                              | XADC      | I   | -             | Positive auxiliary differential analog inputs  |
| P21                                       | VAUXN[15 : 0]                              | XADC      | I   | -             | Negative auxiliary differential analog inputs  |
| P22                                       | CONVST                                     | XADC      | I   | -             | Convert Start input port is used to control the sampling instant on the ADC input and is used only in event-driven sampling mode. This port is auto connected to ground internally, if not in use. |

Table 2-1: AXI XADC I/O Signal Descriptions (Cont'd)

| Port | Signal Name  | Interface | I/O | Initial State | Description                               |
|------|--------------|-----------|-----|---------------|---|
| P23  | ALARM[7:0]   | XADC      | O   | 0             | XADC hard macro Alarm output signals      |
| P24  | MUXADDR[4:0] | XADC      | O   | 0             | XADC external multiplexer address outputs |

## AXI XADC Register Descriptions

Table 2-2 shows the AXI XADC IP Core registers and their corresponding addresses.

Table 2-2: IP Core Registers

| Base Address + Offset (hex)                            | Register Name                           | Access Type          | Default Value (hex) | Description                               |
|--|---|----------------------|---------------------|---|
| <b>AXI XADC Local Register Grouping</b>                |   |                      |                     |   |
| C_BASEADDR + 0x00                                      | Software Reset Register (SRR)           | Write <sup>(1)</sup> | N/A                 | Software reset register                   |
| C_BASEADDR + 0x04                                      | Status Register (SR)                    | Read <sup>(2)</sup>  | N/A                 | Status register                           |
| C_BASEADDR + 0x08                                      | Alarm Output Status Register (AOSR)     | Read <sup>(2)</sup>  | 0x0                 | Alarm output status register              |
| C_BASEADDR + 0x0C                                      | CONVST Register (CONVSTR)               | Write <sup>(1)</sup> | N/A                 | ADC convert start register <sup>(3)</sup> |
| C_BASEADDR + 0x10                                      | XADC Reset Register (SYSMONRR)          | Write <sup>(1)</sup> | N/A                 | XADC hard macro reset register            |
| <b>AXI XADC Interrupt Controller Register Grouping</b> |   |                      |                     |   |
| C_BASEADDR + 0x5C                                      | Global Interrupt Enable Register (GIER) | R/W                  | 0x0                 | Global interrupt enable register          |
| C_BASEADDR + 0x60                                      | IP Interrupt Status Register (IPISR)    | R/TOW <sup>(4)</sup> | N/A                 | IP interrupt status register              |
| C_BASEADDR + 0x68                                      | IP Interrupt Enable Register (IPIER)    | R/W                  | 0x0                 | IP interrupt enable register              |

Table 2-2: IP Core Registers (Cont'd)

| Base Address + Offset (hex)                            | Register Name                               | Access Type         | Default Value (hex) | Description  |
|--|---|---------------------|---------------------|--|
| <b>XADC Hard Macro Register Grouping<sup>(5)</sup></b> |   |                     |                     |  |
| C_BASEADDR + 0x200                                     | Temperature                                 | Read <sup>(6)</sup> | N/A                 | The 12-bit Most Significant Bit (MSB) justified result of on-device temperature measurement is stored in this register.  |
| C_BASEADDR + 0x204                                     | V <sub>CCINT</sub>                          | Read <sup>(6)</sup> | N/a                 | The 12-bit MSB justified result of on-device V <sub>CCINT</sub> supply monitor measurement is stored in this register.   |
| C_BASEADDR + 0x208                                     | V <sub>CCAUX</sub>                          | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified result of on-device V <sub>CCAUX</sub> Data supply monitor measurement is stored in this register.  |
| C_BASEADDR + 0x20C                                     | V <sub>P</sub> /V <sub>N</sub>              | R/W <sup>(7)</sup>  | 0x0                 | When read: The 12-bit MSB justified result of A/D conversion on the dedicated analog input channel (V <sub>P</sub> /V <sub>N</sub> ) is stored in this register.<br>When written: Write to this register resets the XADC hard macro. No specific data is required. |
| C_BASEADDR + 0x210                                     | V <sub>REFP</sub>                           | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the reference input V <sub>REFP</sub> is stored in this register.   |
| C_BASEADDR + 0x214                                     | V <sub>REFN</sub>                           | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the reference input V <sub>REFN</sub> is stored in this register.   |
| C_BASEADDR + 0x218                                     | V <sub>BRAM</sub>                           | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the reference input V <sub>BRAM</sub> is stored in this register.   |
| C_BASEADDR + 0x21C                                     | Undefined                                   | N/A                 | Undefined           | These locations are unused and contain invalid data.   |
| C_BASEADDR + 0x220                                     | Supply Offset                               | Read <sup>(6)</sup> | N/A                 | The calibration coefficient for the supply sensor offset is stored in this register.   |
| C_BASEADDR + 0x224                                     | ADC Offset                                  | Read <sup>(6)</sup> | N/A                 | The calibration coefficient for the ADC offset calibration is stored in this register.   |
| C_BASEADDR + 0x228                                     | Gain Error                                  | Read <sup>(6)</sup> | N/A                 | The calibration coefficient for the gain error is stored in this register.   |
| C_BASEADDR + 0x22C to C_BASEADDR + 0x230               | Undefined                                   | N/A                 | Undefined           | These locations are unused and contain invalid data.   |
| C_BASEADDR + 0x240                                     | V <sub>AUXP</sub> [0]/V <sub>AUXN</sub> [0] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 0 is stored in this register.  |
| C_BASEADDR + 0x244                                     | V <sub>AUXP</sub> [1]/V <sub>AUXN</sub> [1] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 1 is stored in this register.  |

Table 2-2: IP Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name                                     | Access Type         | Default Value (hex) | Description  |
|-----------------------------|---|---------------------|---------------------|--|
| C_BASEADDR + 0x248          | V <sub>AUXP</sub> [2]/<br>V <sub>AUXN</sub> [2]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 2 is stored in this register.  |
| C_BASEADDR + 0x24C          | V <sub>AUXP</sub> [3]/<br>V <sub>AUXN</sub> [3]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 3 is stored in this register.  |
| C_BASEADDR + 0x250          | V <sub>AUXP</sub> [4]/<br>V <sub>AUXN</sub> [4]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 4 is stored in this register.  |
| C_BASEADDR + 0x254          | V <sub>AUXP</sub> [5]/<br>V <sub>AUXN</sub> [5]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 5 is stored in this register.  |
| C_BASEADDR + 0x258          | V <sub>AUXP</sub> [6]/<br>V <sub>AUXN</sub> [6]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 6 is stored in this register.  |
| C_BASEADDR + 0x25C          | V <sub>AUXP</sub> [7]/<br>V <sub>AUXN</sub> [7]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 7 is stored in this register.  |
| C_BASEADDR + 0x260          | V <sub>AUXP</sub> [8]/<br>V <sub>AUXN</sub> [8]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 8 is stored in this register.  |
| C_BASEADDR + 0x264          | V <sub>AUXP</sub> [9]/<br>V <sub>AUXN</sub> [9]   | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 9 is stored in this register.  |
| C_BASEADDR + 0x268          | V <sub>AUXP</sub> [10]/<br>V <sub>AUXN</sub> [10] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 10 is stored in this register. |
| C_BASEADDR + 0x26C          | V <sub>AUXP</sub> [11]/<br>V <sub>AUXN</sub> [11] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 11 is stored in this register. |
| C_BASEADDR + 0x270          | V <sub>AUXP</sub> [12]/<br>V <sub>AUXN</sub> [12] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 12 is stored in this register. |
| C_BASEADDR + 0x274          | V <sub>AUXP</sub> [13]/<br>V <sub>AUXN</sub> [13] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 13 is stored in this register. |
| C_BASEADDR + 0x278          | V <sub>AUXP</sub> [14]/<br>V <sub>AUXN</sub> [14] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 14 is stored in this register. |
| C_BASEADDR + 0x27C          | V <sub>AUXP</sub> [15]/<br>V <sub>AUXN</sub> [15] | Read <sup>(6)</sup> | 0x0                 | The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 15 is stored in this register. |

Table 2-2: IP Core Registers (Cont'd)

| Base Address + Offset (hex)              | Register Name            | Access Type         | Default Value (hex) | Description   |
|--|--------------------------|---------------------|---------------------|---|
| C_BASEADDR + 0x280                       | Max Temp                 | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified maximum temperature measurement.   |
| C_BASEADDR + 0x284                       | Max V <sub>CCINT</sub>   | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified maximum V <sub>CCINT</sub> measurement.  |
| C_BASEADDR + 0x288                       | Max V <sub>CCAUX</sub>   | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified maximum V <sub>CCAUX</sub> measurement.  |
| C_BASEADDR + 0x28C                       | Max V <sub>BRAM</sub>    | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified maximum V <sub>BRAM</sub> measurement.   |
| C_BASEADDR + 0x290                       | Min Temp                 | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified minimum temperature measurement  |
| C_BASEADDR + 0x294                       | Min V <sub>CCINT</sub>   | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified minimum V <sub>CCINT</sub> measurement   |
| C_BASEADDR + 0x298                       | Min V <sub>CCAUX</sub>   | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified minimum V <sub>CCAUX</sub> measurement.  |
| C_BASEADDR + 0x29C                       | Min V <sub>BRAM</sub>    | Read <sup>(6)</sup> | N/A                 | The 12-bit MSB justified minimum V <sub>BRAM</sub> measurement.   |
| C_BASEADDR + 0x2AC                       | Undefined                | N/A                 | Undefined           | These locations are unused and contain invalid data.  |
| C_BASEADDR + 0x2BC to C_BASEADDR + 0x2F8 | Undefined                | N/A                 | Undefined           | These locations are unused and contain invalid data.  |
| C_BASEADDR + 0x2FC                       | Flag Register            | Read <sup>(6)</sup> | N/A                 | The 16-bit register gives general status information of ALARM, Over Temperature (OT), Disable information of XADC and information whether the XADC is using internal reference voltage or external reference voltage. |
| C_BASEADDR + 0x300                       | Configuration Register 0 | R/W <sup>(8)</sup>  | 0x0                 | XADC Configuration register 0   |
| C_BASEADDR + 0x304                       | Configuration Register 1 | R/W <sup>(8)</sup>  | 0x0                 | XADC Configuration register 1   |
| C_BASEADDR + 0x308                       | Configuration Register 2 | R/W <sup>(8)</sup>  | 0x1E00              | XADC Configuration register 2   |
| C_BASEADDR + 0x30C to C_BASEADDR + 0x31C | Test register 0 to 4     | N/A                 | N/A                 | XADC Test register 0 to 4 (for factory test only)   |
| C_BASEADDR + 0x320                       | Sequence Register 0      | R/W                 | 0x0                 | XADC Sequence register 0 (ADC channel selection)  |
| C_BASEADDR + 0x324                       | Sequence Register 1      | R/W                 | 0x0                 | XADC Sequence register 1 (ADC channel selection)  |
| C_BASEADDR + 0x328                       | Sequence Register 2      | R/W                 | 0x0                 | XADC Sequence register 2 (ADC channel averaging enable)   |

Table 2-2: IP Core Registers (Cont'd)

| Base Address + Offset (hex) | Register Name              | Access Type           | Default Value (hex) | Description  |
|-----------------------------|----------------------------|-----------------------|---------------------|--|
| C_BASEADDR + 0x32C          | Sequence Register 3        | R/W                   | 0x0                 | XADC Sequence register 3 (ADC channel averaging enable)                        |
| C_BASEADDR + 0x330          | Sequence Register 4        | R/W                   | 0x0                 | XADC Sequence register 4 (ADC channel analog-input mode)                       |
| C_BASEADDR + 0x334          | Sequence Register 5        | R/W                   | 0x0                 | XADC Sequence register 5 (ADC channel analog-input mode)                       |
| C_BASEADDR + 0x338          | Sequence Register 6        | R/W                   | 0x0                 | XADC Sequence register 6 (ADC channel acquisition time)                        |
| C_BASEADDR + 0x33C          | Sequence Register 7        | R/W                   | 0x0                 | XADC Sequence register 7 (ADC channel acquisition time)                        |
| C_BASEADDR + 0x340          | Alarm Threshold Register 0 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 0 (Temperature Upper)        |
| C_BASEADDR + 0x344          | Alarm Threshold Register 1 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 1 (V <sub>CCINT</sub> Upper) |
| C_BASEADDR + 0x348          | Alarm Threshold Register 2 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 2 (V <sub>CCAUX</sub> Upper) |
| C_BASEADDR + 0x34C          | Alarm Threshold Register 3 | R/W <sup>(8)(9)</sup> | 0x0                 | The 12-bit MSB justified alarm threshold register 3 (OT Upper)                 |
| C_BASEADDR + 0x350          | Alarm Threshold Register 4 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 4 (Temperature Lower)        |
| C_BASEADDR + 0x354          | Alarm Threshold Register 5 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 5 (V <sub>CCINT</sub> Lower) |
| C_BASEADDR + 0x358          | Alarm Threshold Register 6 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 6 (V <sub>CCAUX</sub> Lower) |
| C_BASEADDR + 0x35C          | Alarm Threshold Register 7 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 7 (OT Lower)                 |
| C_BASEADDR + 0x360          | Alarm Threshold Register 8 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 8 (VBRAM Upper)              |
| C_BASEADDR + 0x370          | Alarm Threshold Register 8 | R/W                   | 0x0                 | The 12-bit MSB justified alarm threshold register 8 (VBRAM Lower)              |



Table 2-2: IP Core Registers (Cont'd)

| Base Address + Offset (hex)  | Register Name | Access Type | Default Value (hex) | Description                      |
|--|---------------|-------------|---------------------|----------------------------------|
| C_BASEADDR + 0x380 to C_BASEADDR + 0x3FC   | Undefined     | N/A         | Undefined           | Do not Read/Write these register |
| <b>Note:</b> <ol style="list-style-type: none"> <li>1. Reading of this register returns an undefined value.</li> <li>2. Writing into this register has no effect.</li> <li>3. Used in event-driven sampling mode only.</li> <li>4. TOW = Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.</li> <li>5. These are 16-bit registers internal to XADC. These are mapped to the lower half word boundary on 32-bit AXI XADC IP core registers.</li> <li>6. Writing to this XADC hard macro register is not allowed. The XADC hard macro data registers are 16-bits in width. The XADC hard macro specification guarantees the first 12-MSB bits accuracy; so only these bits are used for reference.</li> <li>7. Writing to this register resets the XADC hard macro. No specific data pattern is required to reset the XADC hard macro. Reading of this register gives the details of Vp/Vn port.</li> <li>8. Read the XADC User Guide, for setting the different bits available in configuration registers for 7 series devices.</li> <li>9. The OT upper register is a user-configurable register for the upper threshold level of temperature. If this register is left unconfigured, then the XADC considers 125°C as the upper threshold value for OT. While configuring this register, the last 4-bits must be set to 0011, that is, Alarm Threshold Register 3[3:0] = 0011. The upper 12 bits of this register are user configurable.</li> </ol> |               |             |                     |                                  |

## AXI XADC Local Register Grouping

It is expected that the AXI XADC IP core registers are accessed in their preferred-access mode only. If the write attempt is made to read-only registers, then there is not any effect on register contents. If the write-only registers are read, then it results in undefined data. All the internal registers of the core have to be accessed in 32-bit format. If any other kind of access (like half word or byte access) is done for AXI XADC IP core's local 32-bit registers, the transaction is completed with a generation of errors for the corresponding transaction.

### Software Reset Register (SRR)

The Software Reset Register permits the programmer to reset the AXI XADC IP core including the XADC hard macro output ports (except JTAG related outputs), independently of other IP cores in the systems. To activate software reset, the value 0x0000\_000A must be written to the register. Any other access, read or write, has undefined results. The bit assignment in the software reset register is shown in Figure 2-1 and described in Table 2-3.

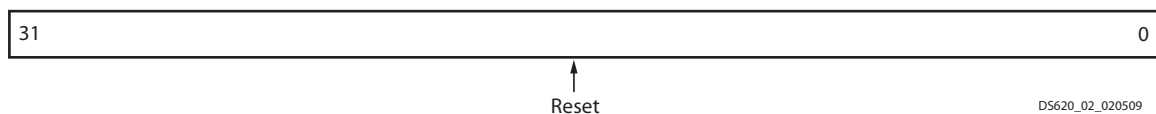


Figure 2-1: Software Reset Register



Table 2-3: Software Reset Register Description (C\_BASEADDR + 0x00)

| Bit(s) | Name  | Core Access | Reset Value | Description   |
|--------|-------|-------------|-------------|---|
| 0-31   | Reset | Write only  | N/A         | The only allowed operation on this register is a write of 0x0000_000A, which resets the AXI XADC IP Core. The reset is active only for 16 clock cycles. |

## Status Register (SR)

Status Register contains the AXI XADC IP core channel status, EOC, EOS, and JTAG access signals. This register is read only. Any attempt to write the bits of the register is not able to change the bits. The Status Register bit definitions are shown in Figure 2-2 and explained in Table 2-4.

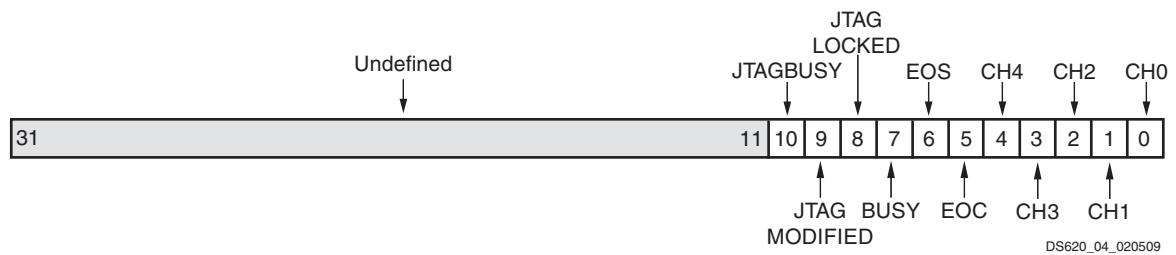


Figure 2-2: Status Register

Table 2-4: Status Register (C\_BASEADDR + 0x04)

| Bit(s)  | Name            | Core Access | Reset Value | Description  |
|---------|-----------------|-------------|-------------|--|
| 31 - 11 | Undefined       | N/A         | N/A         | Undefined  |
| 10      | JTAGBUSY        | Read        | 0           | Used to indicate that a JTAG DRP transaction is in progress.   |
| 9       | JTAG MODIFIED   | Read        | 0           | Used to indicate that a write to DRP through JTAG interface has occurred. This bit is cleared when a successful DRP read/write operation through the FPGA logic is performed. The DRP read/write through the FPGA logic fails, if JTAGLOCKED = '1'       |
| 8       | JTAG LOCKED     | Read        | 0           | Used to indicate that a DRP port lock request has been made by the Joint Test Action Group (JTAG) interface.   |
| 7       | BUSY            | Read        | N/A         | ADC busy signal. This signal transitions high during an ADC conversion.  |
| 6       | EOS             | Read        | N/A         | End of Sequence. This signal transitions to an active High when the measurement data from the last channel in the auto sequence is written to the status registers. This bit is cleared when a read operation is performed on status register.           |
| 5       | EOC             | Read        | N/A         | End of Conversion signal. This signal transitions to an active High at the end of an ADC conversion when the measurement is written to the XADC hard macro's status register. This bit is cleared when a read operation is performed on status register. |
| 4 - 0   | CHANNEL [4 : 0] | Read        | N/A         | 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.  |

## Alarm Output Status Register (AOSR)

Alarm Output Status Register contains all the alarm outputs for the AXI XADC IP core. This register is read only. Any attempt to write the bits of the register is not able to change the bits. The Alarm Output Status Register bit definitions are shown in Figure 2-3 and explained in Table 2-5.

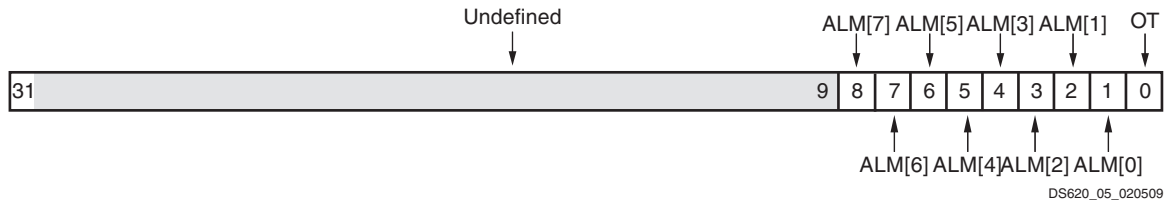


Figure 2-3: Alarm Output Status Register

Table 2-5: Alarm Output Status Register (C\_BASEADDR + 0x08)

| Bit(s) | Name      | Core Access | Reset Value | Description   |
|--------|-----------|-------------|-------------|---|
| 31 - 9 | Undefined | N/A         | N/A         | Undefined   |
| 8      | ALM[7]    | Read        | '0'         | Logical ORing of ALARM bits 0 to 7. This is direct output from the XADC macro.  |
| 7-5    | ALM[4-6]  | N/A         | N/A         | Reserved  |
| 4      | ALM[3]    | Read        | '0'         | <b>XADC V<sub>BRAM</sub>-sensor Status.</b> XADC V <sub>BRAM</sub> -sensor alarm output interrupt occurs when V <sub>BRAM</sub> exceeds user-defined threshold.       |
| 3      | ALM[2]    | Read        | '0'         | <b>XADC V<sub>CCAUX</sub>-sensor Status.</b> XADC V <sub>CCAUX</sub> -sensor alarm output interrupt occurs when V <sub>CCAUX</sub> exceeds user-defined threshold.    |
| 2      | ALM[1]    | Read        | '0'         | <b>XADC V<sub>CCINT</sub>-sensor Status.</b> XADC V <sub>CCINT</sub> -sensor alarm output interrupt occurs when V <sub>CCINT</sub> exceeds user-defined threshold.    |
| 1      | ALM[0]    | Read        | '0'         | <b>XADC temperature-sensor Status.</b> XADC temperature-sensor alarm output interrupt occurs when device temperature exceeds user-defined threshold.                  |
| 0      | OT        | Read        | '0'         | <b>XADC Over-Temperature alarm Status.</b> Over-Temperature alarm output interrupt occurs when the die temperature exceeds a factory set limit of 125 degree celsius. |

## CONVST Register (CONVSTR)

The CONVST Register is used for initiating a new conversion in the event-driven sampling mode. The output of this register is logically ORed with the external CONVST input signal. The attempt to read this register results in undefined data. The CONVST Register bit definitions are shown in Figure 2-4 and explained in Table 2-6.

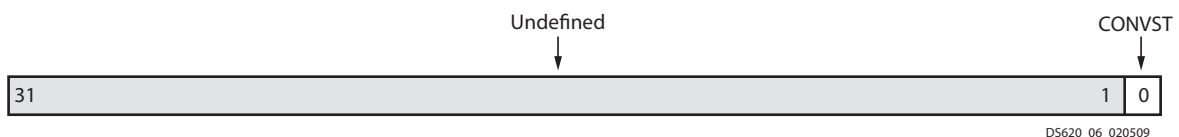


Figure 2-4: CONVST Register

Table 2-6: CONVST Register (C\_BASEADDR + 0x0C)

| Bit(s) | Name      | Core Access | Reset Value | Description  |
|--------|-----------|-------------|-------------|--|
| 31 - 1 | Undefined | N/A         | N/A         | Undefined  |
| 0      | CONVST    | Write       | '0'         | A rising edge on the CONVST input initiates start of ADC conversion in event-driven sampling mode. For the selected channel the CONVST bit in the register needs to be set to '1' and again reset to '0' to start a new conversion cycle. The conversion cycle ends with EOC bit going high. |

## XADC Reset Register

The XADC Reset Register is used to reset only the XADC hard macro. As soon as the reset is released the ADC begins with a new conversion. If sequencing is enabled this conversion is the first in the sequence. This register resets the OT and ALM[n] output from the XADC hard macro. This register does not reset the interrupt registers if they are included in the design. Also any reset from the FPGA logic does not affect the RFI (Register File Interface) contents of XADC hard macro. The attempt to read this register results in undefined data. The XADC Reset Register bit definitions are shown in Figure 2-5 and explained in Table 2-7.

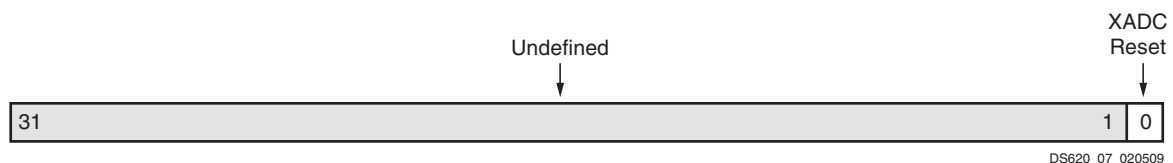


Figure 2-5: XADC Reset Register

Table 2-7: XADC Reset Register (C\_BASEADDR + 0x10)

| Bit(s) | Name       | Core Access | Reset Value | Description  |
|--------|------------|-------------|-------------|--|
| 31 - 1 | Undefined  | N/A         | N/A         | Undefined  |
| 0      | XADC Reset | Write       | '0'         | Writing '1' to this bit position resets the XADC hard macro. The reset is released only after '0' is written to this register. |

## AXI XADC Interrupt Controller Register Grouping

The Interrupt Controller Module is included in the AXI XADC IP core design when C\_INCLUDE\_INTR = '1'. The AXI XADC has a number of distinct interrupts that are sent to the Interrupt Controller Module which is one of the submodules of AXI XADC IP Core. The Interrupt Controller Module allows each interrupt to be enabled independently (via the IP interrupt enable register (IPIER)). All the interrupt signals are rising-edge sensitive.

Interrupt registers are strictly 32-bit accessible. If byte/half-word or without byte-enables kind of access is made, the core behavior is not guaranteed.

The interrupt registers are in the Interrupt Controller Module. The AXI XADC IP core permits multiple conditions for an interrupt or an interrupt strobe which occurs only after the completion of a transfer.

## Global Interrupt Enable Register (GIER)

The Global Interrupt Enable Register (GIER) is used to globally enable the final interrupt output from the Interrupt Controller as shown in Figure 2-6 and described in Table 2-8. This bit is a read/write bit and is cleared upon reset.

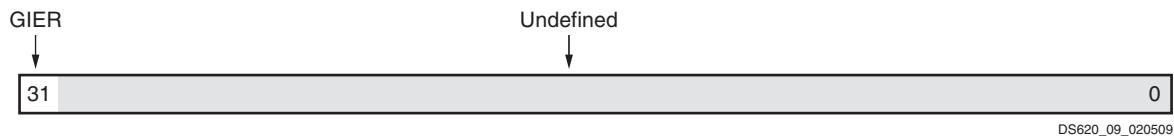


Figure 2-6: Global Interrupt Enable Register (GIER)

Table 2-8: Global Interrupt Enable Register (GIER) Description (C\_BASEADDR + 0x5C)

| Bit(s) | Name      | Access | Reset Value | Description  |
|--------|-----------|--------|-------------|--|
| 31     | GIER      | R/W    | '0'         | <b>Global Interrupt Enable Register.</b> It enables all individually enabled interrupts to be passed to the interrupt controller.<br>'0' = Disabled<br>'1' = Enabled |
| 30 - 0 | Undefined | N/A    | N/A         | Undefined.   |

## IP Interrupt Status Register (IPISR)

Six unique interrupt conditions are possible in the AXI XADC IP core.

The Interrupt Controller has a register that can enable each interrupt independently. Bit assignment in the Interrupt register for a 32-bit data bus is shown in Figure 2-7 and described in Table 2-9. The interrupt register is a read/toggle on write register and by writing a '1' to a bit position within the register causes the corresponding bit position in the register to 'toggle'. All register bits are cleared upon reset.

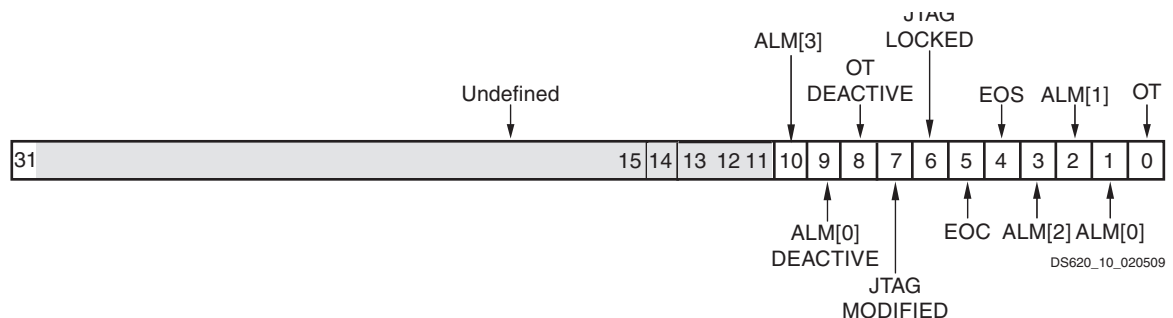


Figure 2-7: IP Interrupt Status Register (IPISR)

Table 2-9: IP Interrupt Status Register (IPISR) Description (C\_BASEADDR + 0x60)

| Bit(s)  | Name                | Access                  | Reset Value | Description   |
|---------|---------------------|-------------------------|-------------|---|
| 31 - 11 | Undefined           | N/A                     | N/A         | Undefined   |
| 10      | ALM[3]              | R/TOW <sup>(1)(2)</sup> | '0'         | <b>XADC V<sub>BRAM</sub>-sensor Interrupt.</b> XADC V <sub>BRAM</sub> -sensor alarm output interrupt occurs when V <sub>BRAM</sub> exceeds user-defined threshold.  |
| 9       | ALM[0]<br>De-active | R/TOW                   | '0'         | <b>ALM[0] Deactive Interrupt.</b> This signal indicates that the falling edge of the Over Temperature signal is detected. It is cleared by writing '1' to this bit position.<br><br>The ALM[0] signal is generated locally from the core. This signal indicates that the XADC macro has deactivated the Over Temperature signal output. |
| 8       | OT<br>Deactive      | R/TOW <sup>(1)</sup>    | '0'         | <b>OT Deactive Interrupt.</b> This signal indicates that falling edge of the Over Temperature signal is detected. It is cleared by writing '1' to this bit position.<br><br>The OT Deactive signal is generated locally from the core. This signal indicates that the XADC macro has deactivated the Over Temperature signal output.    |
| 7       | JTAG<br>MODIFIED    | R/TOW <sup>(1)(2)</sup> | '0'         | <b>JTAGMODIFIED Interrupt.</b> This signal indicates that a write to DRP through the JTAG interface has occurred. It is cleared by writing '1' to this bit position.  |
| 6       | JTAG<br>LOCKED      | R/TOW <sup>(1)(2)</sup> | '0'         | <b>JTAGLOCKED Interrupt.</b> This signal is used to indicate that a DRP port lock request has been made by the Joint Test Action Group (JTAG) interface.  |
| 5       | EOC                 | R/TOW <sup>(1)(2)</sup> | N/A         | <b>End of Conversion signal Interrupt.</b> This signal transitions to an active High at the end of an ADC conversion when the measurement is written to the XADC hard macro's status register.  |
| 4       | EOS                 | R/TOW <sup>(1)(2)</sup> | N/A         | <b>End of Sequence Interrupt.</b> This signal transitions to an active High when the measurement data from the last channel in the auto sequence is written to the status registers.  |
| 3       | ALM[2]              | R/TOW <sup>(1)(2)</sup> | '0'         | <b>XADC V<sub>CCAUX</sub>-sensor Interrupt.</b> XADC V <sub>CCAUX</sub> -sensor alarm output interrupt occurs when V <sub>CCAUX</sub> exceeds the user-defined threshold.   |
| 2       | ALM[1]              | R/TOW <sup>(1)(2)</sup> | '0'         | <b>XADC V<sub>CCINT</sub>-sensor Interrupt.</b> XADC V <sub>CCINT</sub> -sensor alarm output interrupt occurs when V <sub>CCINT</sub> exceeds the user-defined threshold.   |
| 1       | ALM[0]              | R/TOW <sup>(1)(2)</sup> | '0'         | <b>XADC temperature-sensor Interrupt.</b> XADC temperature-sensor alarm output interrupt occurs when device temperature exceeds the user-defined threshold.   |
| 0       | OT                  | R/TOW <sup>(1)(2)</sup> | '0'         | <b>Over-Temperature alarm Interrupt.</b> Over-Temperature alarm output interrupt occurs when the die temperature exceeds a factory set limit of 125 degree celsius.   |

**Note:**

1. TOW = Toggle On Write. Writing a '1' to a bit position within the register causes the corresponding bit position in the register to toggle.
2. This interrupt signal is directly generated from the XADC hard macro.

## IP Interrupt Enable Register (IPIER)

The IPIER has an enable bit for each defined bit of the IPISR as shown in Figure 2-8 and described in Table 2-10. All bits are cleared upon reset.

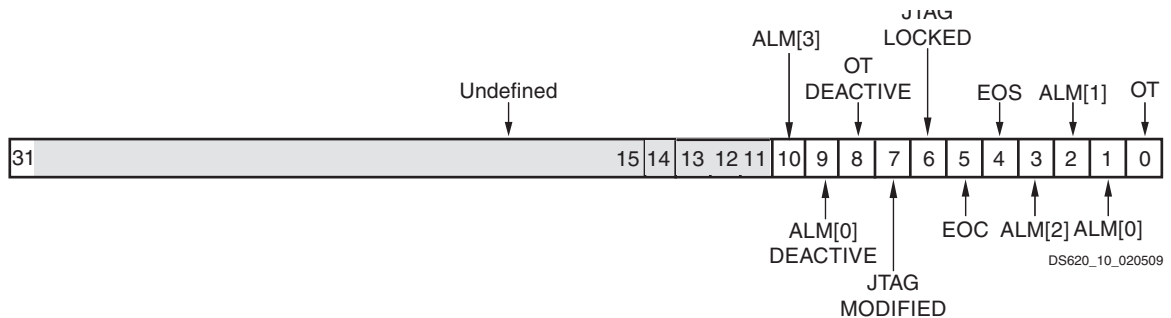


Figure 2-8: IP Interrupt Enable Register (IPIER)

Table 2-10: IP Interrupt Enable Register (IPIER) Description (C\_BASEADDR + 0x68)

| Bit(s)  | Name            | Access | Reset Value | Description  |
|---------|-----------------|--------|-------------|--|
| 31 - 11 | Undefined       | N/A    | N/A         | Undefined  |
| 10      | ALM[3]          | R/W    | '0'         | XADC V <sub>BRAM</sub> -sensor Interrupt<br>'0' = Disabled<br>'1' = Enabled  |
| 9       | ALM[0] Deactive | R/W    | '0'         | ALM[0] Deactive Interrupt<br>'0' = Disabled<br>'1' = Enabled                 |
| 8       | OT Deactive     | R/W    | '0'         | OT Deactive Interrupt<br>'0' = Disabled<br>'1' = Enabled                     |
| 7       | JTAG MODIFIED   | R/W    | '0'         | JTAGMODIFIED Interrupt<br>'0' = Disabled<br>'1' = Enabled                    |
| 6       | JTAG LOCKED     | R/W    | '0'         | JTAGLOCKED Interrupt<br>'0' = Disabled<br>'1' = Enabled                      |
| 5       | EOC             | R/W    | '0'         | End of Conversion signal Interrupt<br>'0' = Disabled<br>'1' = Enabled        |
| 4       | EOS             | R/W    | '0'         | End of Sequence Interrupt<br>'0' = Disabled<br>'1' = Enabled                 |
| 3       | ALM[2]          | R/W    | '0'         | XADC V <sub>CCAUX</sub> -sensor Interrupt<br>'0' = Disabled<br>'1' = Enabled |

Table 2-10: IP Interrupt Enable Register (IPIER) Description (C\_BASEADDR + 0x68) (Cont'd)

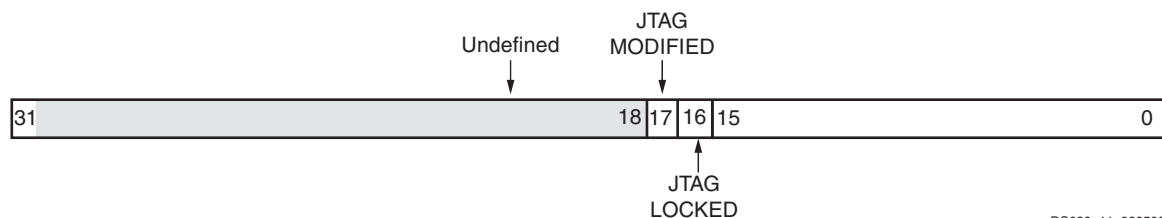
| Bit(s) | Name   | Access | Reset Value | Description  |
|--------|--------|--------|-------------|--|
| 2      | ALM[1] | R/W    | '0'         | XADC V <sub>CCINT</sub> -sensor Interrupt<br>'0' = Disabled<br>'1' = Enabled |
| 1      | ALM[0] | R/W    | '0'         | XADC temperature-sensor Interrupt<br>'0' = Disabled<br>'1' = Enabled         |
| 0      | OT     | R/W    | '0'         | Over-Temperature alarm Interrupt<br>'0' = Disabled<br>'1' = Enabled          |

## More about Locally Generated Interrupt Bits in IPIER and IPISR

The interrupt bits ranging from the bit-16 to bit-0 in IPISR as well as IPIER are direct output signals of the XADC hard macro. The signals like OT Deactive (bit-8), ALM[0] Deactive (bit-9), are locally generated in the core. These interrupts are generated on the falling edge of the Over Temperature and AML[0] signals. The falling edge of these signals can be used in controlling external things like controlling the fan or air-conditioning of the system.

## XADC Hard Macro Register (DRP Register) Grouping

The XADC hard macro register set consists of all the registers present in the XADC hard macro on 7 series FPGAs. The addresses of these registers are mentioned in Table 2-2. Because these registers are 16-bits wide but the processor data bus is 32-bit wide, the hard macro register data resides on the lower 16 bits of the 32-bit data bus as shown in Figure 2-9. The 12-bit MSB aligned A/D converted value of different channels from XADC hard macro are left-shifted and reside from bit position 15 to 6 of the processor data bus. The remaining bit positions from 5 to 0 should be ignored while considering the ADC data for different channels. Along with 16-bit data, the JTAGMODIFIED and JTAGLOCKED bits are passed that can be used by the software driver application for determining the validity of the DRP read data. The JTAGMODIFIED bit is cleared when a DRP read/write operation through the FPGA logic is successful. A DRP read/write through the FPGA logic fails, if JTAGLOCKED = '1'. The JTAGLOCKED signal is independently controlled through JTAG TAP. It is expected that these XADC hard macro registers should be accessed in their preferred access-mode only. The AXI XADC IP core is not able to differentiate any non-preferred access to the XADC hard macro registers.



DS620\_11\_020509

Figure 2-9: XADC Hard Macro Register

DRP registers are accessed as part of the core's local registers. So it is a must to access these registers through the core local registers and any attempt to access these registers in byte or half-word manner returns the error response from core.



# *Customizing and Generating the Core*

---

## **Output Generation**

This core can be customized in the XPS tool to suit user needs. Based on the selections, the parameters in the MHS file are updated.

## Designing with the Core

This chapter includes guidelines and additional information to make designing with the core easier.

### AXI XADC Design Parameters

To allow the user to obtain an AXI XADC IP core that is uniquely tailored for the system, certain features can be parameterized in the AXI XADC design. This allows the user to configure a design that utilizes the resources required by the system only and that operates with the best possible performance. The features that can be parameterized are as shown in [Table 4-1](#).

Table 4-1: AXI XADC Design Parameters

| Generic                    | Feature/Description                | Parameter Name     | Allowable Values   | Default Value       | VHDL Type        |
|----------------------------|------------------------------------|--------------------|--|---------------------|------------------|
| <b>System Parameters</b>   |                                    |                    |  |                     |                  |
| G1                         | Target FPGA family                 | C_FAMILY           | "Kintex7, Virtex7"   | "Kintex7"           | string           |
| <b>AXI4 Parameters</b>     |                                    |                    |  |                     |                  |
| G2                         | AXI Base Address                   | C_BASEADDR         | Valid Address <sup>[1]</sup>                                   | None <sup>[2]</sup> | std_logic_vector |
| G3                         | AXI High Address                   | C_HIGHADDR         | Valid Address <sup>[1]</sup>                                   | None <sup>[2]</sup> | std_logic_vector |
| G4                         | AXI Address Bus Width              | C_S_AXI_ADDR_WIDTH | 32   | 32                  | integer          |
| G5                         | AXI Data Bus Width                 | C_S_AXI_DATA_WIDTH | 32   | 32                  | integer          |
| <b>AXI XADC Parameters</b> |                                    |                    |  |                     |                  |
| G6                         | Include/Exclude interrupt support  | C_INCLUDE_INTR     | 0 = Exclude interrupt support<br>1 = Include interrupt support | 1                   | integer          |
| G7                         | File name for Analog input stimuli | C_SIM_MONITOR_FILE | string   | Design.txt          | string           |

**Note:**

1. The range C\_BASEADDR to C\_HIGHADDR is the address range for the AXI XADC. This range is subject to restrictions to accommodate the simple address decoding scheme that is employed. The size, C\_HIGHADDR - C\_BASEADDR + 1, must be a power of two and must be at least 0x400 to accommodate all AXI XADC registers. However, a larger power of two can be chosen to reduce decoding logic. C\_BASEADDR must be aligned to a multiple of the range size.
2. No default value is specified to ensure that an actual value appropriate to the system is set.

## Parameter - Port Dependencies

The dependencies between the AXI XADC IP Core design parameters and I/O signals are described in [Table 4-2](#).

**Table 4-2: Parameter-Port Dependencies**

| Generic or Port          | Name   | Affects     | Depends | Relationship Description   |
|--------------------------|--|-------------|---------|--|
| <b>Design Parameters</b> |  |             |         |  |
| G4                       | C_S_AXI_ADDR_WIDTH                           | P3, P13     | -       | Affects the number of bits in address bus                        |
| G5                       | C_S_AXI_DATA_WIDTH                           | P6, P7, P16 | -       | Affects the number of bits in data bus                           |
| <b>I/O Signals</b>       |  |             |         |  |
| P3                       | S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0]   | -           | G4      | Width of the S_AXI_AWADDR varies with C_S_AXI_ADDR_WIDTH.        |
| P6                       | S_AXI_WDATA[(C_S_AXI_DATA_WIDTH - 1) : 0]    | -           | G5      | Width of the S_AXI_WDATA varies according to C_S_AXI_DATA_WIDTH. |
| P7                       | S_AXI_WSTB[((C_S_AXI_DATA_WIDTH/8) - 1) : 0] | -           | G5      | Width of the S_AXI_WSTB varies according to C_S_AXI_DATA_WIDTH.  |
| P13                      | S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH - 1) : 0]   | -           | G4      | Width of the S_AXI_ARADDR varies with C_S_AXI_ADDR_WIDTH.        |
| P16                      | S_AXI_RDATA[(C_S_AXI_DATA_WIDTH - 1) : 0]    | -           | G5      | Width of the S_AXI_RDATA varies according to C_S_AXI_DATA_WIDTH. |

In addition to the parameters listed in this table, there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see DS768, AXI Interconnect IP Data Sheet.

## Clocking

The clock to XADC is the S\_AXI\_ACLK clock. Hence the ADCCLK division factor must be programmed taking into consideration the S\_AXI\_ACLK frequency.

## Resets

Certain registers of the IP can be reset by writing a value 0xA to register 0x00. The AXI4-lite interface also has its own reset pin.

# Constraining the Core

---

## Design Constraints

An example User Constraints File for this core is shown below and must be modified for use in the system.

### Timing Constraints

For complete timing coverage, it is recommended that the following timing constraints on the XADC hard macro ports are specified along with the S\_AXI\_ACLK input constraint. Following is an example.

```
NET "S_AXI_AClk" TNM NET = "S_AXI_AClk";  
TIMESPEC "TS_AXI_AClk" = PERIOD "S_AXI_AClk" 5 ns HIGH 50%;
```

**Note:** This particular PERIOD constraint is typically already covered by system-level clocking constraints.

## *Detailed Example Design*

---

There is no example design for this core.

## Additional Resources

---

### Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

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

For a glossary of technical terms used in Xilinx documentation, see:

[http://www.xilinx.com/support/documentation/sw\\_manuals/glossary.pdf](http://www.xilinx.com/support/documentation/sw_manuals/glossary.pdf).

### List of Acronyms

Table 7-1: List of Acronyms

| Acronym | Description                               |
|---------|---|
| A/D     | Analog-to-Digital                         |
| ADC     | Analog-to-Digital Converter               |
| ALM     | Alarm                                     |
| AMBA®   | Advanced Microcontroller Bus Architecture |
| AOSR    | Alarm Output Status Register              |
| ARM®    | Advanced RISC Machine                     |
| AXI     | Advanced eXtensible Interface             |
| CONVSTR | CONVST Register                           |
| DRP     | Dynamic Reconfiguration Port              |
| EOC     | End of Conversion                         |
| EOS     | End of Sequence                           |
| FF      | flip-flop                                 |
| FPGA    | Field Programmable Gate Array             |
| GIER    | Global Interrupt Enable Register          |
| I/O     | Input/Output                              |
| IP      | Intellectual Property                     |
| IPIER   | IP interrupt enable register              |
| IPIF    | IP Interface                              |

Table 7-1: List of Acronyms (Cont'd)

| Acronym  | Description  |
|----------|--|
| IPISR    | IP Interrupt Status Register   |
| JTAG     | Joint Test Action Group  |
| LUT      | Lookup Table   |
| MHz      | Mega Hertz   |
| MSB      | Most Significant Bit   |
| MSPS     | Mega-Samples Per Second  |
| OT       | Over Temperature   |
| RAM      | Random Access Memory   |
| RFI      | Register File Interface  |
| SR       | Status Register  |
| SRR      | Software Reset Register  |
| SYSMONRR | XADC Reset Register  |
| TAP      | Test Access Port   |
| TOW      | Toggle On Write  |
| VHDL     | VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits) |
| XADC     | Xilinx Analog-to-Digital Converter   |
| XPS      | Xilinx Platform Studio   |
| XST      | Xilinx Synthesis Technology  |

## Solution Centers

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

## References

To search for Xilinx documentation, go to <http://www.xilinx.com/support>.

1. 7 Series FPGA XADC User Guide (latest version)
2. AXI4 Advanced Microcontroller Bus Architecture (AMBA®) AXI Protocol Version: 2.0 Specification
3. DS765 LogiCORE IP AXI Lite IPIF (axi\_lite\_ipif) (v1.01a) Data Sheet

## Technical Support

Xilinx provides technical support at [www.xilinx.com/support](http://www.xilinx.com/support) for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

## Ordering Information

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx [Intellectual Property page](#). For information on pricing and availability of other Xilinx LogiCORE modules and software, contact your local Xilinx [sales representative](#).

## Revision History

The following table shows the revision history for this document.

| Date     | Version | Revision                |
|----------|---------|-------------------------|
| 10/19/11 | 1.0     | Initial Xilinx release. |

## Notice of Disclaimer

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 notify you of updates to the Materials or to product specifications. 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 2010–2011 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. AMBA and ARM are registered trademarks of ARM in the EU and other countries. All other trademarks are the property of their respective owners.