LogiCORE IP AXI XADC v1.00a

Product Guide

PG019 October 16, 2012

XILINX®

XILINX®

Table of Contents

IP Facts

Chapter 1: Overview

Operating System Requirements 8
Feature Summary 8
Applications 9
Licensing and Ordering Information 9

Chapter 2: Product Specification

Standards Compliance 10
Resource Utilization 10
Port Descriptions 11
AXI XADC Register Descriptions 13
AXI XADC Local Register Grouping 19
AXI XADC Interrupt Controller Register Grouping 23
XADC Hard Macro Register (DRP Register) Grouping 28

Chapter 3: Designing with the Core

AXI XADC Design Parameters 30
Parameter - Port Dependencies 31
Clocking 31
Resets 31

Chapter 4: Customizing and Generating the Core

GUI 32
Output Generation 35

Chapter 5: Constraining the Core

Appendix A: Debugging

Appendix B: Additional Resources

Xilinx Resources 38

Solution Centers 38
References 38
Technical Support 39
Revision History 39
Notice of Disclaimer 39

Introduction

The LogiCORE™ IP Advanced eXtensible Interface (AXI) Xilinx® Analog-to-Digital Converter (XADC) core is a 32-bit slave peripheral that connects to the AXI4 interface and provides the controller interface for System Monitor XADC hard macro on Zynq™-7000 devices and 7 series FPGAs. This section provides the facts and features for the AXI XADC core.

Features

- AXI4-Lite interface 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(1)	Zynq-7000(2), Virtex®-7, Kintex™-7, Artix™-7
Supported User Interfaces	AXI4-Lite
Resources	See Table 2-1.
Provided with Core	
Design Files	VHDL
Example Design	VHDL
Test Bench	Not Provided
Constraints File	Not Applicable
Simulation Model	Not Provided
Supported S/W Drivers(3)	Standalone
Tested Design Flows(4)	
Design Entry	Xilinx Platform Studio (XPS) 14.3 Vivado™ Design Suite(5) 2012.3
Simulation	Mentor Graphics ModelSim
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

1. For a complete list of supported derivative devices, see Embedded Edition Derivative Device Support.
2. Supported in PlanAhead with XPS/ISE implementations only.
3. Standalone driver details can be found in the EDK or SDK directory (<*install_directory*>/doc/usenglish/xilinx_drivers.htm).
4. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.
5. Supports only 7 series devices.

Overview

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

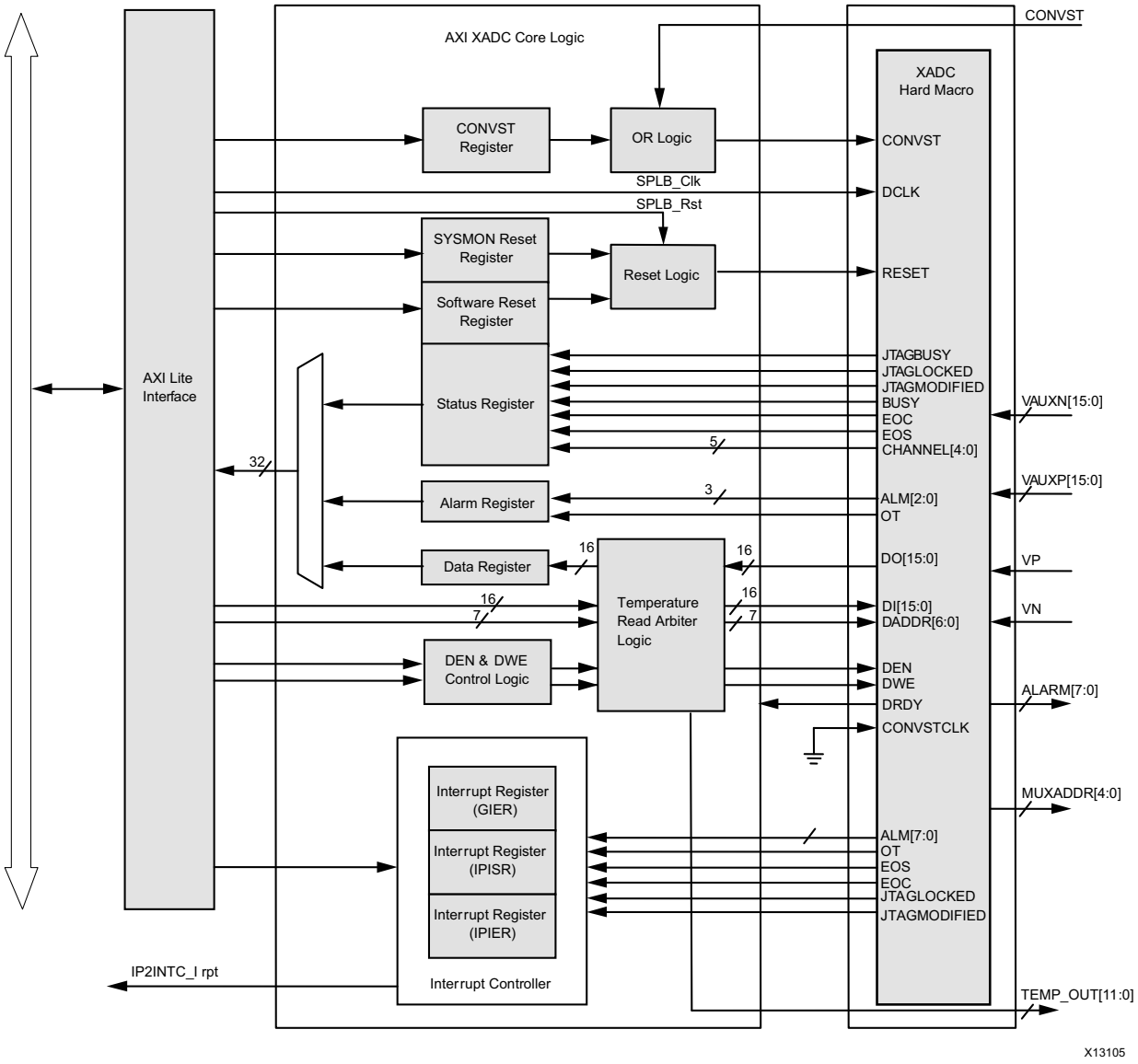

Figure 1-1: **Block Diagram of AXI XADC IP Core**

The AXI XADC IP Core consists of these 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. There is only one XADC primitive available in a 7 series device and Zynq-7000 AP SoC. If the AXI XADC core is used in an EDK system, the TEMP_OUT bus should be connected to the xadc_device_temp_i input port of the DDR3_SDRAM (MIG) block.

EDK will automatically disable inference of XADC hard block in DDR3_SDRAM if xadc_device_temp_i is connected. If AXI XADC is not used then xadc_device_temp_i bus is not connected and the XADC primitive is used by DDR3_SDRAM.

XADC Hard Macro

The XADC hard macro can be accessed via both Joint Test Action Group Test Access Port (JTAG TAP) and the AXI XADC 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 device logic. When JTAGLOCKED port is again deasserted through JTAG TAP, AXI XADC 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 device 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 device logic and thus ensures a non-destructive access through JTAG TAP.XADC Hard Macro.

The XADC hard macro is present in every 7 series device family and Zynq™-7000 AP SoC product family. The block diagram for the XADC hard macro is shown in Figure 1-2.

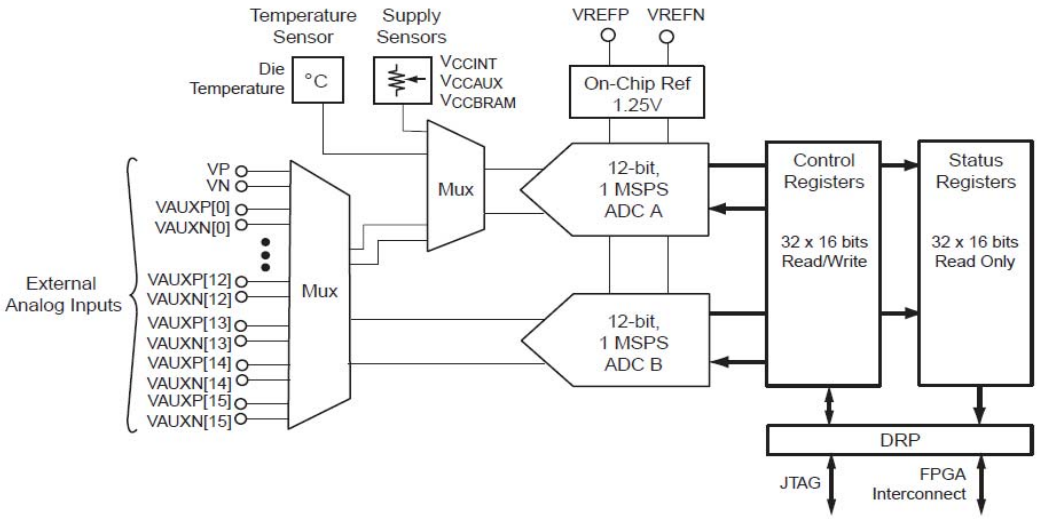

Figure 1-2: **Block Diagram of XADC Hard Macro**

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 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 device 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 Xilinx Design Tools: 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 XADC 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 and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado Design Suite and ISE Design Suite Embedded Edition tools under the terms of the Xilinx End User License.

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 IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards Compliance

The LogiCORE™ IP AXI XADC core contains an AXI4-Lite interface, which is based on the AMBA® AXI4 specification.

Resource Utilization

Because the AXI XADC IP core can be used with other design modules in the device, 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 device resources and timing of the AXI XADC IP core design varies from the results reported here.

Table 2-1 details the AXI XADC IP Core resource utilization for various parameter combinations measured with Xilinx® 7 series devices and Zynq™-7000 devices as the target device.

Table 2-1: **Performance and Resource Utilization on the 7 Series and Zynq-7000 Devices**

Parameter Values (Other parameters at default values)		Device Resources		Performance
C_INCLUDE_INTR	C_HAS_TEMP_BUS	Slice Flip-Flops	LUTs	F_{Max} (MHz)
0	0	140	144	100
1	0	238	242	100
0	1	336	383	100
1	1	428	497	100

Port Descriptions

The AXI XADC I/O signals are listed and described in Table 2-2.

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

Port	Signal Name	Interface	I/O	Initial State	Description
AXI Global System Signals					
P1	S_AXI_ACLK	AXI	I	-	AXI Clock
P2	S_AXI_ARESETN	AXI	I	-	AXI Reset, active LOW
AXI Write Address Channel Signals					
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.
AXI Write Channel Signals					
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.
AXI Write Response Channel Signals					
P10	S_AXI_BRESP[1 : 0]	AXI	O	0	Write response. This signal indicates the status of the write transaction “00” - OKAY (normal response) “10” - SLVERR (error response) “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.
AXI Read Address Channel Signals					
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.

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

Port	Signal Name	Interface	I/O	Initial State	Description
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.
AXI Read Data Channel Signals					
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. "00" - OKAY (normal response) "10" - SLVERR (error condition) "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.
AXI XADC IP Core Interface Signals					
P20	VAUXP[15:0]	XADC	I	-	Positive auxiliary differential analog inputs(1)(2)
P21	VAUXN[15:0]	XADC	I	-	Negative auxiliary differential analog inputs(1)(2)
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.
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
P25	TEMP_OUT[11:0]	XADC	O	-	12-bit temperature output bus for MIG. Should be connected to xadc_device_temp_i pin of MIG.

Notes:

1. Not all devices support 16 VAUX channels. See the device data sheet for more details.
2. The AXI XADC connects all 16 VAUX channels to the XADC. If you want to connect only a few channels, modifications are necessary to the MHS file to tie the rest of the channels to '0'.

AXI XADC Register Descriptions

Table 2-3 shows the AXI XADC IP Core registers and their corresponding addresses.

Table 2-3: **IP Core Registers**

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
AXI XADC Local Register Grouping				
C_BASEADDR + 0x00	Software Reset Register (SRR)	Write(1)	N/A	Software reset register
C_BASEADDR + 0x04	Status Register (SR)	Read(2)	N/A	Status register
C_BASEADDR + 0x08	Alarm Output Status Register (AOSR)	Read(2)	0x0	Alarm output status register
C_BASEADDR + 0x0C	CONVST Register (CONVSTR)	Write(1)	N/A	Bit [0]: ADC convert start register(3) Bit [1]: Enable temperature update logic Bit [17:2]: Wait cycle for temperature update
C_BASEADDR + 0x10	XADC Reset Register (SYSMONRR)	Write(1)	N/A	XADC hard macro reset register
AXI XADC Interrupt Controller Register Grouping				
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(4)	N/A	IP interrupt status register
C_BASEADDR + 0x68	IP Interrupt Enable Register (IPIER)	R/W	0x0	IP interrupt enable register
XADC Hard Macro Register Grouping(5)				
C_BASEADDR + 0x200	Temperature	Read(6)	N/A	The 12-bit Most Significant Bit (MSB) justified result of on-device temperature measurement is stored in this register.

Table 2-3: **IP Core Registers *(Cont'd)***

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x204	V_{CCINT}	Read[6]	N/A	The 12-bit MSB justified result of on-device V_{CCINT} supply monitor measurement is stored in this register.
C_BASEADDR + 0x208	V_{CCAUX}	Read[6]	N/A	The 12-bit MSB justified result of on-device V_{CCAUX} Data supply monitor measurement is stored in this register.
C_BASEADDR + 0x20C	V_P/V_N	R/W[7]	0x0	When read: The 12-bit MSB justified result of A/D conversion on the dedicated analog input channel (Vp/Vn) is stored in this register. When written: Write to this register resets the XADC hard macro. No specific data is required.
C_BASEADDR + 0x210	V_{REFP}	Read[6]	0x0	The 12-bit MSB justified result of A/D conversion on the reference input V_{REFP} is stored in this register.
C_BASEADDR + 0x214	V_{REFN}	Read[6]	0x0	The 12-bit MSB justified result of A/D conversion on the reference input V_{REFN} is stored in this register.
C_BASEADDR + 0x218	V_{BRAM}	Read[6]	0x0	The 12-bit MSB justified result of A/D conversion on the reference input V_{BRAM} 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[6]	N/A	The calibration coefficient for the supply sensor offset is stored in this register.
C_BASEADDR + 0x224	ADC Offset	Read[6]	N/A	The calibration coefficient for the ADC offset calibration is stored in this register.
C_BASEADDR + 0x228	Gain Error	Read[6]	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 + 0x234	Zynq-7000 Device Core Supply	N/A	Read	The VCCINT of PSS core supply. Present only on Zynq™-7000 devices.
C_BASEADDR + 0x238	Zynq-7000 Device Core Aux Supply	Read	N/A	The VCCAUX of PSS core supply. Present only on Zynq-7000 devices.
C_BASEADDR + 0x23C	Zynq-7000 Device Core Memory Supply	Read	N/A	The VCCMEM of PSS core supply. Present only on Zynq-7000 devices.

Table 2-3: **IP Core Registers *(Cont'd)***

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x240	$V_{AUXP}[0]$/ $V_{AUXN}[0]$	Read(6)	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	$V_{AUXP}[1]$/ $V_{AUXN}[1]$	Read(6)	0x0	The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 1 is stored in this register.
C_BASEADDR + 0x248	$V_{AUXP}[2]$/ $V_{AUXN}[2]$	Read(6)	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	$V_{AUXP}[3]$/ $V_{AUXN}[3]$	Read(6)	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	$V_{AUXP}[4]$/ $V_{AUXN}[4]$	Read(6)	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	$V_{AUXP}[5]$/ $V_{AUXN}[5]$	Read(6)	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	$V_{AUXP}[6]$/ $V_{AUXN}[6]$	Read(6)	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	$V_{AUXP}[7]$/ $V_{AUXN}[7]$	Read(6)	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	$V_{AUXP}[8]$/ $V_{AUXN}[8]$	Read(6)	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	$V_{AUXP}[9]$/ $V_{AUXN}[9]$	Read(6)	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	$V_{AUXP}[10]$/ $V_{AUXN}[10]$	Read(6)	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	$V_{AUXP}[11]$/ $V_{AUXN}[11]$	Read(6)	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	$V_{AUXP}[12]$/ $V_{AUXN}[12]$	Read(6)	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	$V_{AUXP}[13]$/ $V_{AUXN}[13]$	Read(6)	0x0	The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 13 is stored in this register.

Table 2-3: **IP Core Registers *(Cont'd)***

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x278	V_{AUXP}[14]/ V_{AUXN}[14]	Read(6)	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_{AUXP}[15]/ V_{AUXN}[15]	Read(6)	0x0	The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 15 is stored in this register.
C_BASEADDR + 0x280	Max Temp	Read(6)	N/A	The 12-bit MSB justified maximum temperature measurement.
C_BASEADDR + 0x284	Max V_{CCINT}	Read(6)	N/A	The 12-bit MSB justified maximum V_{CCINT} measurement.
C_BASEADDR + 0x288	Max V_{CCAUX}	Read(6)	N/A	The 12-bit MSB justified maximum V_{CCAUX} measurement.
C_BASEADDR + 0x28C	Max V_{BRAM}	Read(6)	N/A	The 12-bit MSB justified maximum V_{BRAM} measurement.
C_BASEADDR + 0x290	Min Temp	Read(6)	N/A	The 12-bit MSB justified minimum temperature measurement
C_BASEADDR + 0x294	Min V_{CCINT}	Read(6)	N/A	The 12-bit MSB justified minimum V_{CCINT} measurement
C_BASEADDR + 0x298	Min V_{CCAUX}	Read(6)	N/A	The 12-bit MSB justified minimum V_{CCAUX} measurement.
C_BASEADDR + 0x29C	Min V_{BRAM}	Read(6)	N/A	The 12-bit MSB justified minimum V_{BRAM} measurement.
C_BASEADDR + 0x2A0	Max V_{CCPINT}	Read	N/A	The 12-bit MSB justified maximum V_{CCPINT} measurement. This register is only available in Zynq devices.
C_BASEADDR + 0x2A4	Max V_{CCPAUX}	Read	N/A	The 12-bit MSB justified maximum V_{CCPAUX} measurement. This register is only available in Zynq devices.
C_BASEADDR + 0x2A8	Max V_{CCPDRO}	Read	N/A	The 12-bit MSB justified maximum V_{CCPDRO} measurement. This register is only available in Zynq devices.
C_BASEADDR + 0x2AC	Undefined	N/A	Undefined	These locations are unused and contain invalid data.
C_BASEADDR + 0x2B0	Min V_{CCPINT}	Read	N/A	The 12-bit MSB justified minimum V_{CCPINT} measurement. This register is only available in Zynq devices.
C_BASEADDR + 0x2B4	Min V_{CCPAUX}	Read	N/A	The 12-bit MSB justified minimum V_{CCPAUX} measurement. This register is only available in Zynq devices.
C_BASEADDR + 0x2B8	Min V_{CCPDRO}	Read	N/A	The 12-bit MSB justified minimum V_{CCPDRO} measurement. This register is only available in Zynq devices.

Table 2-3: **IP Core Registers *(Cont'd)***

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
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(6)	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(8)	0x0	XADC Configuration register 0
C_BASEADDR + 0x304	Configuration Register 1	R/W(8)	0x0	XADC Configuration register 1
C_BASEADDR + 0x308	Configuration Register 2	R/W(8)	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)
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_{CCINT} Upper)

Table 2-3: **IP Core Registers *(Cont'd)***

Base Address + Offset (hex)	Register Name	Access Type	Default Value (hex)	Description
C_BASEADDR + 0x348	Alarm Threshold Register 2	R/W	0x0	The 12-bit MSB justified alarm threshold register 2 (V_{CCAUX} Upper)
C_BASEADDR + 0x34C	Alarm Threshold Register 3	R/W(8)(9)	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_{CCINT} Lower)
C_BASEADDR + 0x358	Alarm Threshold Register 6	R/W	0x0	The 12-bit MSB justified alarm threshold register 6 (V_{CCAUX} 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 + 0x364	Alarm Threshold Register 9	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPint} Upper). This register is only on Zynq devices.
C_BASEADDR + 0x368	Alarm Threshold Register 10	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPaux} Upper). This register is only on Zynq devices.
C_BASEADDR + 0x36C	Alarm Threshold Register 11	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPdro} Upper). This register is only on Zynq devices.
C_BASEADDR + 0x370	Alarm Threshold Register 8	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (VBRAM Lower)
C_BASEADDR + 0x374	Alarm Threshold Register 13	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPint} Lower). This register is only on Zynq devices.
C_BASEADDR + 0x378	Alarm Threshold Register 14	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPaux} Lower). This register is only on Zynq devices.
C_BASEADDR + 0x37C	Alarm Threshold Register 15	R/W	0x0	The 12-bit MSB justified alarm threshold register 8 (V_{CCPdro} Lower). This register is only on Zynq devices.

Table 2-3: **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

Note:

1. Reading of this register returns an undefined value.
2. Writing into this register has no effect.
3. Used in event-driven sampling mode only.
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.
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.
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.
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.
8. Read the XADC User Guide, for setting the different bits available in configuration registers for 7 series devices.
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 125^0C 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.

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 2-4.

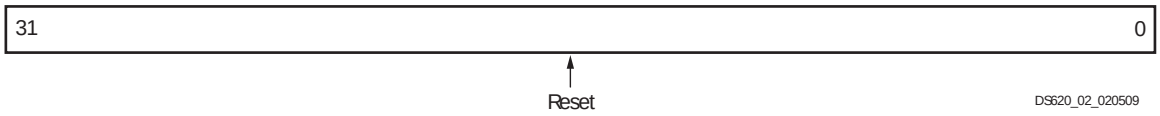

Figure 2-1: **Software Reset Register**

Table 2-4: **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 2-5.

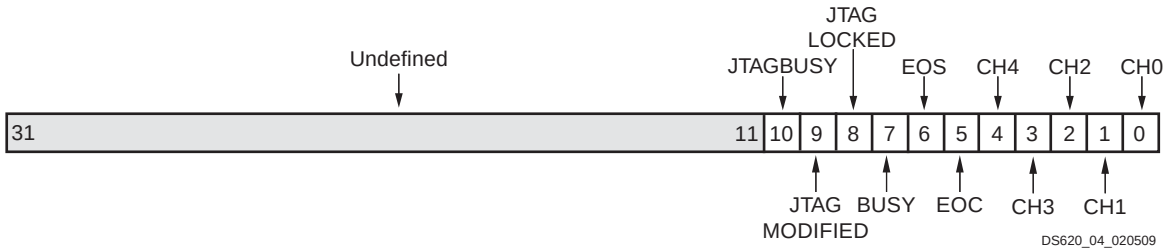

Figure 2-2: **Status Register**

Table 2-5: **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 device logic is performed. The DRP read/ write through the device 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.

Table 2-5: **Status Register (C_BASEADDR + 0x04) *(Cont'd)***

Bit(s)	Name	Core Access	Reset Value	Description
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 2-6.

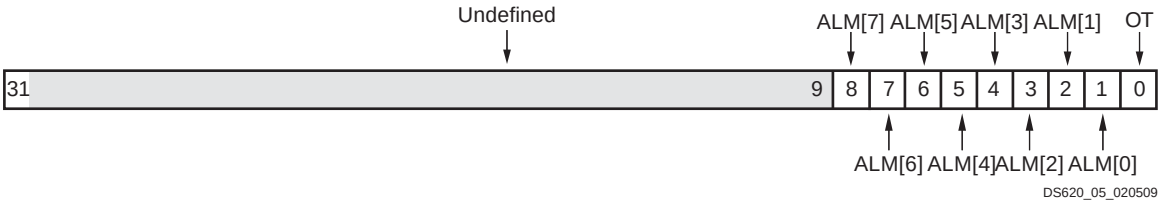

Figure 2-3: **Alarm Output Status Register**

Table 2-6: **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	ALM[6]	Read	'0'	**XADC VCCPdro-Sensor Status**. The XADCCCPdro-sensor alarm output interrupt occurs when V_{CCPdro} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices.
6	ALM[5]	Read	'0'	**XADC VCCPAUX-Sensor Status**. The XADCCCPAUX-sensor alarm output interrupt occurs when V_{CCPAUX} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices.
5	ALM[4]	Read	'0'	**XADC VCCPINT-Sensor Status**. The XADCCCPINT-sensor alarm output interrupt occurs when V_{CCPINT} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices.

Table 2-6: **Alarm Output Status Register (C_BASEADDR + 0x08) *(Cont'd)***

Bit(s)	Name	Core Access	Reset Value	Description
4	ALM[3]	Read	'0'	**XADC V_{BRAM}-Sensor Status.** $XADC_{VBRAM}$-sensor alarm output interrupt occurs when V_{BRAM} exceeds user-defined threshold.
3	ALM[2]	Read	'0'	**XADC V_{CCAUX}-Sensor Status.** XADC V_{CCAUX}-sensor alarm output interrupt occurs when V_{CCAUX} exceeds user-defined threshold.
2	ALM[1]	Read	'0'	**XADC V_{CCINT}-Sensor Status.** XADC V_{CCINT}-sensor alarm output interrupt occurs when V_{CCINT} exceeds user-defined threshold.
1	ALM[0]	Read	'0'	**XADC Temperature-Sensor Status.** XADC temperature-sensor alarm output interrupt occurs when device temperature exceeds user-defined threshold.
0	OT	Read	'0'	**XADC Over-Temperature Alarm Status.** Over-Temperature alarm output interrupt occurs when the die temperature exceeds a factory set limit of 125 °C.

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 2-7.

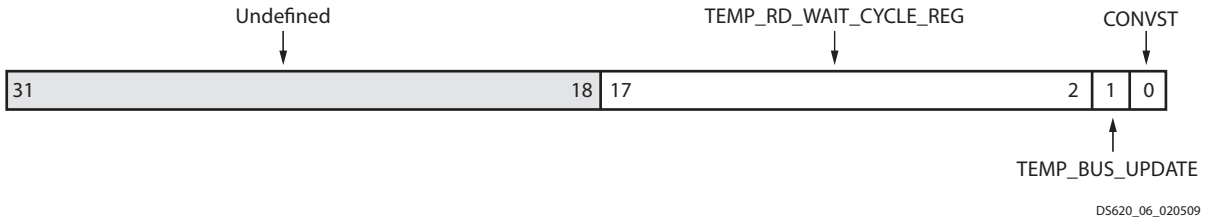

Figure 2-4: **CONVST Register**

Table 2-7: **CONVST Register (C_BASEADDR + 0x0C)**

Bit(s)	Name	Core Access	Reset Value	Description
31:18	Undefined	N/A	N/A	Undefined.
17:2	TEMP_RD_WAIT_CYCLE_REG	Write Only	0x03E8	Wait cycle for temperature update. Temperature update logic waits for this count of the S_AXI_ACLK.

Table 2-7: **CONVST Register (C_BASEADDR + 0x0C)**

Bit(s)	Name	Core Access	Reset Value	Description
1	TEMP_BUS_UPDATE	Write Only	0	Enable temperature update logic enables the temperature read from XADC and update of TEMP_OUT port.
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 device 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 2-8.

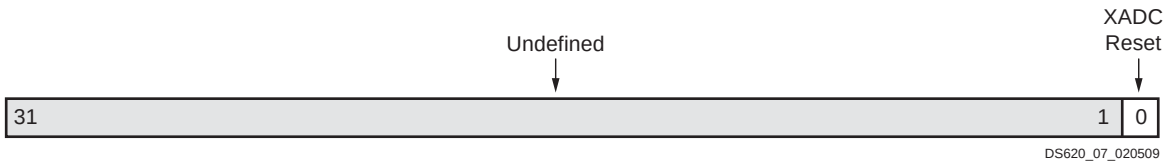

Figure 2-5: **XADC Reset Register**

Table 2-8: **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 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 2-9. This bit is a read/write bit and is cleared upon reset.

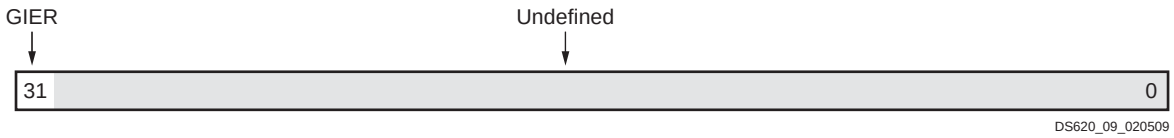

Figure 2-6: **Global Interrupt Enable Register (GIER)**

Table 2-9: **Global Interrupt Enable Register (GIER) Description (C_BASEADDR + 0x5C)**

Bit(s)	Name	Access	Reset Value	Description
31	GIER	R/W	'0'	**Global Interrupt Enable Register.** It enables all individually enabled interrupts to be passed to the interrupt controller. '0' = Disabled '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 2-10. 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.

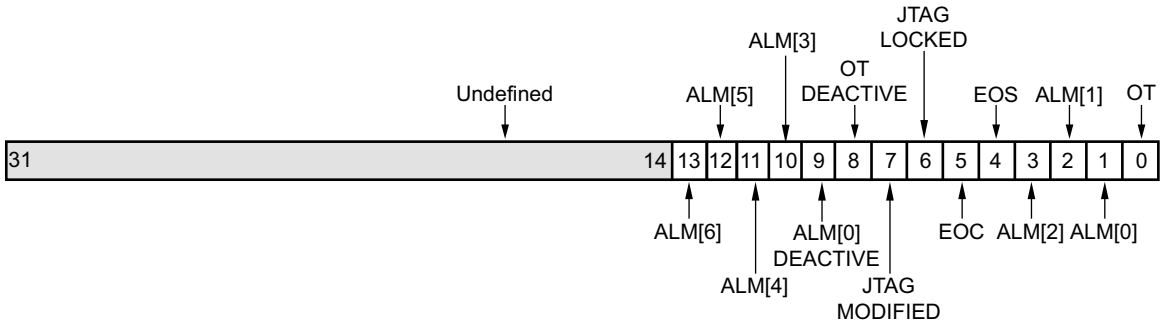

Figure 2-7: **IP Interrupt Status Register (IPISR)**

Table 2-10: **IP Interrupt Status Register (IPISR) Description (C_BASEADDR + 0x60)**

Bit(s)	Name	Access	Reset Value	Description
31:14	Undefined	N/A	N/A	Undefined
13	ALM[6]	R/TOW(1)(2)	'0'	**XADC $V_{CCPddro}$-Sensor Interrupt**. The XADC V_{CCPdro}-sensor alarm output interrupt occurs when V_{CCPdro} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices.
12	ALM[5]	R/TOW(1)(2)	'0'	**XADC V_{CCPAUX}-Sensor Interrupt**. The XADC V_{CCPAUX}-sensor alarm output interrupt occurs when V_{CCPAUX} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices.
11	ALM[4]	R/TOW(1)(2)	'0'	**XADC V_{CCPINT}-Sensor Interrupt**. The XADC V_{CCPINT}-sensor alarm output interrupt occurs when V_{CCPINT} exceeds the user-defined threshold. This bit is only valid for Zynq-7000 devices
10	ALM[3]	R/TOW(1)(2)	'0'	**XADC V_{BRAM}-Sensor Interrupt.** XADC V_{BRAM}-sensor alarm output interrupt occurs when V_{BRAM} exceeds user-defined threshold.
9	ALM[0] Deactive	R/TOW	'0'	**ALM[0] Deactive Interrupt.** This signal indicates that the falling edge of the Over Temperature signal is detected. It is cleared by writing '1' to this bit position. 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 Deactive	R/TOW(1)	'0'	**OT Deactive Interrupt.** This signal indicates that falling edge of the Over Temperature signal is detected. It is cleared by writing '1' to this bit position. 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 MODIFIED	R/TOW(1)(2)	'0'	**JTAGMODIFIED Interrupt.** This signal indicates that a write to DRP through the JTAG interface has occurred. It is cleared by writing '1' to this bit position.

Table 2-10: **IP Interrupt Status Register (IPISR) Description (C_BASEADDR + 0x60) *(Cont'd)***

Bit(s)	Name	Access	Reset Value	Description
6	JTAG LOCKED	R/TOW[1][2]	'0'	**JTAGLOCKED Interrupt.** 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[1][2]	N/A	**End of Conversion Signal Interrupt.** 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[1][2]	N/A	**End of Sequence Interrupt.** 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[1][2]	'0'	**XADC V_{CCAUX}-Sensor Interrupt.** XADC V_{CCAUX}-sensor alarm output interrupt occurs when V_{CCAUX} exceeds the user-defined threshold.
2	ALM[1]	R/TOW[1][2]	'0'	**XADC V_{CCINT}-Sensor Interrupt.** XADC V_{CCINT}-sensor alarm output interrupt occurs when V_{CCINT} exceeds the user-defined threshold.
1	ALM[0]	R/TOW[1][2]	'0'	**XADC Temperature-Sensor Interrupt.** XADC temperature-sensor alarm output interrupt occurs when device temperature exceeds the user-defined threshold.
0	OT	R/TOW[1][2]	'0'	**Over-Temperature Alarm Interrupt.** Over-Temperature alarm output interrupt occurs when the die temperature exceeds a factory set limit of 125 °C.

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 2-11. All bits are cleared upon reset.

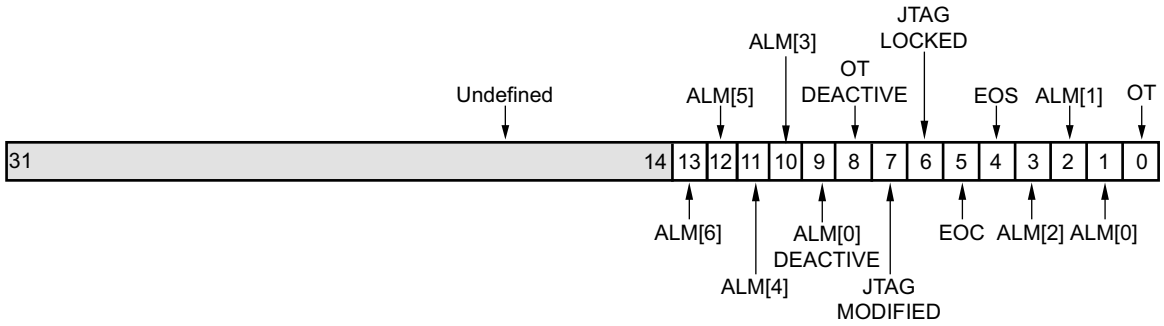

Figure 2-8: **IP Interrupt Enable Register (IPIER)**

Table 2-11: **IP Interrupt Enable Register (IPIER) Description (C_BASEADDR + 0x68)**

Bit(s)	Name	Access	Reset Value	Description
31:14	Undefined	N/A	N/A	Undefined
13	ALM[6]	R/W	'0'	XADC $V_{CCPddro}$-Sensor Interrupt. '0' = Disabled '1' = Enabled
12	ALM[5]	R/W	'0'	XADC V_{CCPAUX}-Sensor Interrupt. '0' = Disabled '1' = Enabled
11	ALM[4]	R/W	'0'	XADC V_{CCPINT}-Sensor Interrupt. '0' = Disabled '1' = Enabled
10	ALM[3]	R/W	'0'	XADC V_{BRAM}-Sensor Interrupt '0' = Disabled '1' = Enabled
9	ALM[0] Deactive	R/W	'0'	ALM[0] Deactive Interrupt '0' = Disabled '1' = Enabled
8	OT Deactive	R/W	'0'	OT Deactive Interrupt '0' = Disabled '1' = Enabled
7	JTAG MODIFIED	R/W	'0'	JTAGMODIFIED Interrupt '0' = Disabled '1' = Enabled
6	JTAG LOCKED	R/W	'0'	JTAGLOCKED Interrupt '0' = Disabled '1' = Enabled

Table 2-11: **IP Interrupt Enable Register (IPIER) Description (C_BASEADDR + 0x68) *(Cont'd)***

Bit(s)	Name	Access	Reset Value	Description
5	EOC	R/W	'0'	End of Conversion Signal Interrupt '0' = Disabled '1' = Enabled
4	EOS	R/W	'0'	End of Sequence Interrupt '0' = Disabled '1' = Enabled
3	ALM[2]	R/W	'0'	XADC V_{CCAUX}-Sensor Interrupt '0' = Disabled '1' = Enabled
2	ALM[1]	R/W	'0'	XADC V_{CCINT}-Sensor Interrupt '0' = Disabled '1' = Enabled
1	ALM[0]	R/W	'0'	XADC Temperature-Sensor Interrupt '0' = Disabled '1' = Enabled
0	OT	R/W	'0'	Over-Temperature Alarm Interrupt '0' = Disabled '1' = Enabled

More about Locally Generated Interrupt Bits in IPIER and IPISR

The interrupt bits ranging from 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 devices. The addresses of these registers are mentioned in Table 2-3. Because these registers are 16-bits wide but the processor data bus is 32 bits 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 device logic is successful. A DRP read/write through the device 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.

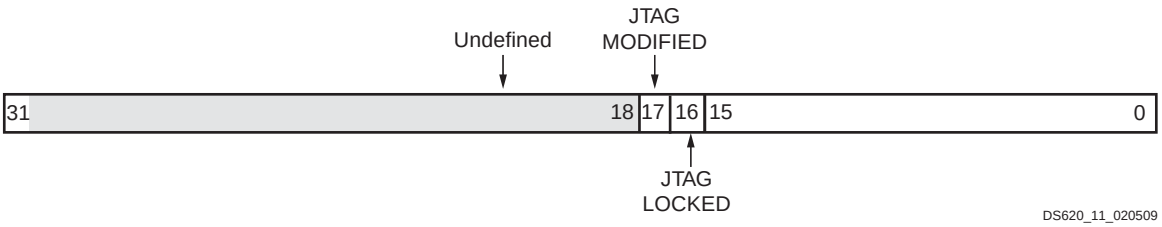

Figure 2-9: **XADC Hard Macro Register**

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.

Designing with the Core

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

AXI XADC Design Parameters

To allow you to obtain an AXI XADC IP core that is uniquely tailored for your system, certain features can be parameterized in the AXI XADC design. This allows you 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 3-1.

Table 3-1: **AXI XADC Design Parameters**

Generic	Feature/ Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameters					
G1	Target device family	C_FAMILY	"Kintex7", "Virtex7", Artix7", "Zynq"	"Kintex7"	string
AXI4 Parameters					
G2	AXI Address Bus Width	C_S_AXI_ADDR_WIDTH	11	11	integer
G3	AXI Data Bus Width	C_S_AXI_DATA_WIDTH	32	32	integer
AXI XADC Parameters					
G4	Include/Exclude Interrupt Support	C_INCLUDE_INTR	0 = Exclude interrupt support 1 = Include interrupt support	1	integer
G5	File name for Analog input stimuli	C_SIM_MONITOR_FILE	string	Design.txt	string
G6	Include 12-bit temperature output bus for MIG	C_HAS_TEMP_BUS	0 = Exclude TEMP_OUT port and its update logic 1 = Include TEMP_OUT port and its update logic	0	integer

Parameter - Port Dependencies

The dependencies between the AXI XADC IP Core design parameters and I/O signals are described in Table 3-2.

Table 3-2: **Parameter-Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G2	C_S_AXI_ADDR_WIDTH	P3, P13	-	Affects the number of bits in address bus
G3	C_S_AXI_DATA_WIDTH	P6, P7, P16	-	Affects the number of bits in data bus
I/O Signals				
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, *LogiCORE IP AXI Interconnect 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.

Customizing and Generating the Core

GUI

The AXI XADC core is located at `Embedded_Processing\AXI_Peripheral\Analog` in the Vivado™ IP Catalog.

To access the AXI XADC core:

1. Open an existing project or create a new project in the Vivado tool.
2. Open the IP catalog and navigate to any of the above taxonomies.
3. Double-click **AXI XADC** to bring up the AXI XADC GUI.

System Parameter Screen (Vivado Tool)

The AXI XADC GUI contains one screen (Figure 4-1), which provides information about the core, allows core configuration, and provides the option to generate the core.

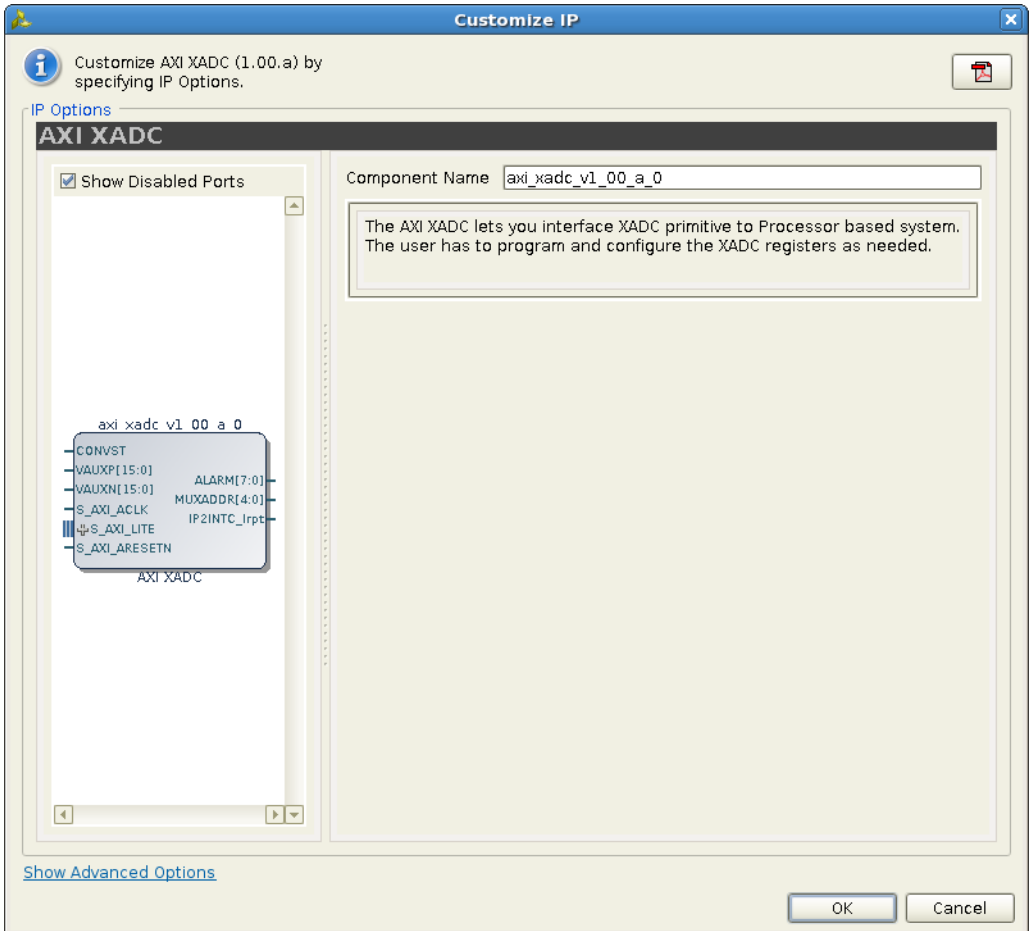

Figure 4-1: **Customize IP Screen in the Vivado Tool**

Component Name

The base name of the output files generated for the core must begin with a letter. It can consist of any of these characters: a to z, 0 to 9, and "_".

There are no customizable options available in the AXI XADC GUI. The XADC registers must be programmed as per user needs.

The AXI XADC, by default, connects all 16 VAUX channels. The unwanted VAUX channels should be tied to 0.

XPS Tool

This core can be customized in the Xilinx® Platform Studio (XPS) tool to suit user needs. Based on the selections, the parameters in the MHS file are updated. Figure 4-2 shows the configuration screen for the AXI XADC core in the XPS tool.

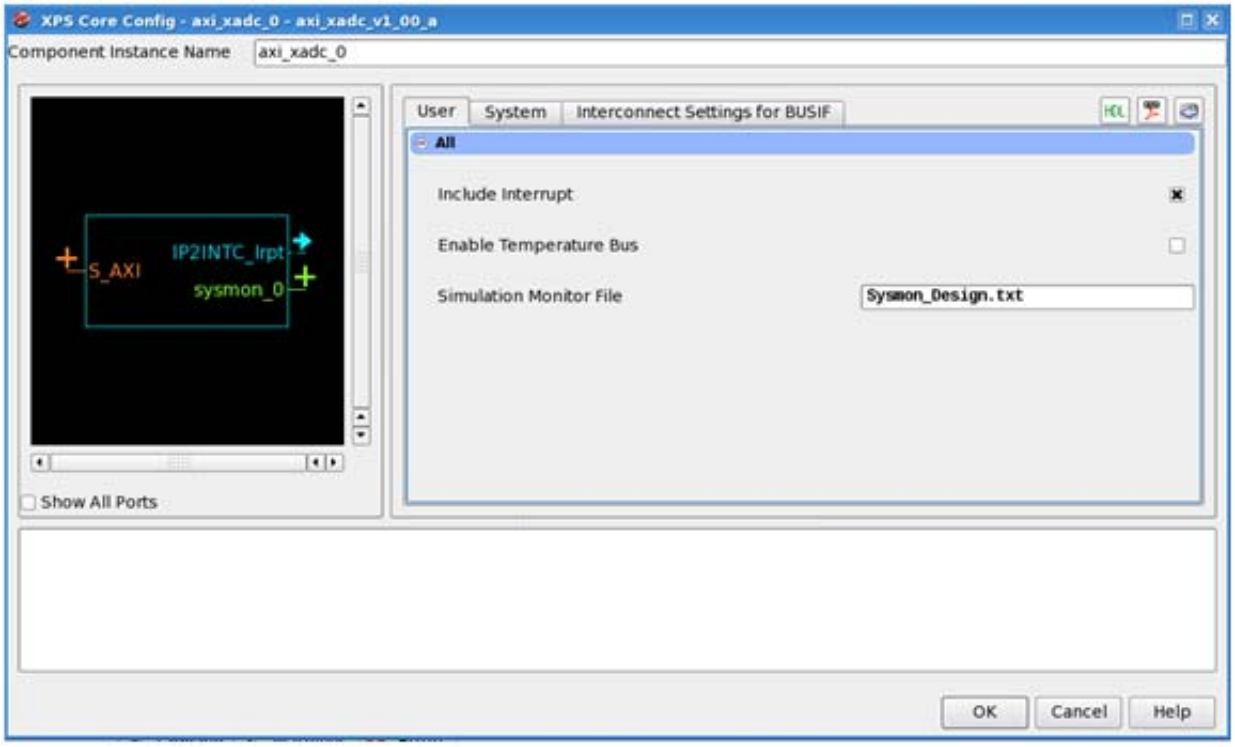

Figure 4-2: **XPS Customization Screen**

Include Interrupt

Check this box if interrupt generation is required. This box is checked by default.

Enable Temperature Bus

Check this box if TEMP_OUT bus is required. This box is unchecked by default.

Simulation Monitor File

Enter the name of the text file that has the information for simulating the XADC. See the *7 Series FPGAs XADC User Guide* [Ref 1] for more information.

Output Generation

The output hierarchy when the core is generated from the Vivado tool is shown in Figure 4-3. Here the component name of the IP generated is `axi_xadc_v1_00_a_0`.

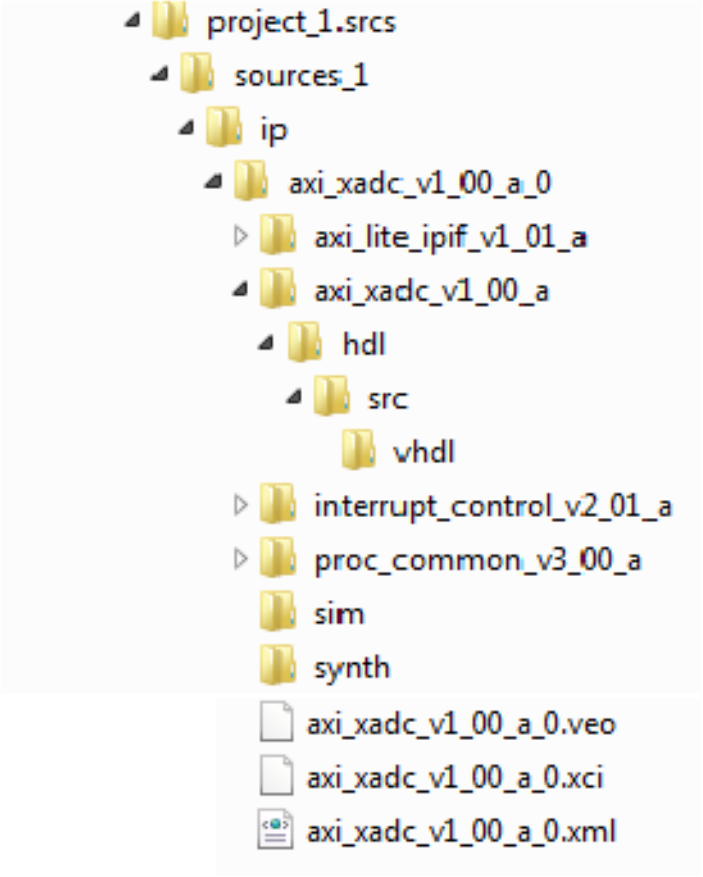

Figure 4-3: **Output Hierarchy**

The Vivado tool delivers the RTL files of the `axi_xadc` core as well as all its helper cores. The helper cores used by the `axi_xadc` core are `axi_lite_ipif_v1_01_a`, `interrupt_control_v2_01_a`, and `proc_common_v3_00_a`.

The RTL files of the `axi_xadc` core are delivered under `/ip/axi_xadc_v1_00_a/hdl/src/vhdl`.

The `sim` and `synth` folders contain the wrappers for simulation and synthesis, respectively.

The tool also delivers the instantiation template file `.veo/.vho`.

Constraining the Core

There are no constraints generated for the AXI XADC core when using the Vivado™ Design Suite.

Debugging

See Solution Centers in Appendix B for information helpful to the debugging progress.

Appendix B

Additional Resources

Xilinx Resources

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

www.xilinx.com/support.

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

www.xilinx.com/company/terms.htm.

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 //www.xilinx.com/support.

1. UG480, *7 Series FPGAs XADC User Guide*
2. DS765, *LogiCORE IP AXI Lite IPIF (axi_lite_ipif) Data Sheet*
3. *AXI4 Advanced Microcontroller Bus Architecture (AMBA®) AXI Protocol Version: 2.0 Specification*
4. Vivado™ Design Suite documentation can be found at www.xilinx.com/cgi-bin/docs/rdoc?v=2012.3;t=vivado+userguides

Technical Support

Xilinx provides technical support at 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.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/11	1.0	Initial Xilinx release.
04/24/12	1.1	• Added support for Artix™-7 and Zynq™-7000 devices. • Added Zynq-7000 device registers to Table 2-3 and Zynq-7000 device specific bits to Table 2-6. • Added ALM[6], ALM[5}, and ALM[4] descriptions to Table 2-10 and Table 2-11 (also Zynq-7000 device specific).
07/25/12	2.0	• Added support for Vivado Design Suite v2012.2. • Added footnotes to Table 2-2. • Removed AXI BASE ADDRESS and AXI HIGH ADDRESS parameters from Table 3-1. • Added GUI and updated Output Generation.
10/16/12	3.0	Updated support for Vivado Design Suite 2012.3 and XPS v14.3. Added details about temperature control logic for MIG.

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 2011–2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, 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.