LogiCORE IP XADC Wizard v1.2

User Guide

UG772 March 1, 2011

XILINX®

Xilinx is providing this product documentation, hereinafter “Information,” to you “AS IS” with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice.

XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.

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

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/11	1.0	Initial Xilinx release.

Table of Contents

Chapter 4: Detailed Example Design

About This Guide

The *LogiCORE IP XADC Wizard User Guide* describes the function and operation of the Xilinx® LogiCORE™ IP XADC Wizard in Xilinx Kintex™-7, and Virtex®-7 FPGA devices.

Guide Contents

This guide contains the following chapters:

- Preface, "About this Guide" introduces the organization and purpose of this guide, a list of additional resources, and the conventions used in this document.
- Chapter 1, Introduction describes the Wizard and related information, including recommended design experience, additional resources, technical support, and submitting feedback to Xilinx.
- Chapter 2, Installing the Wizard provides instructions for installing and licensing the XADC Wizard in the Xilinx® CORE Generator™ tool.
- Chapter 3, Running the Wizard provides a step-by-step procedure for generating the XADC Wizard, implementing the Wizard in hardware using the accompanying example design, and simulating the Wizard with the provided example test bench.
- Chapter 4, Detailed Example Design provides detailed information about the example design, including a description of files and the directory structure generated by the Xilinx® CORE Generator™ tool, the purpose and contents of the provided scripts, the contents of the example design, and the operation of the demonstration test bench.

Additional Resources

To find additional documentation, see the Xilinx website at:

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

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

http://www.xilinx.com/support/mysupport.htm.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *User Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Dark Shading	Items that are not supported or reserved	This feature is not supported
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*option_name*] *design_name*
Braces { }	A list of items from which you must choose one or more	**`lowpwr =`**{**`on`**|**`off`**}
Vertical bar |	Separates items in a list of choices	**`lowpwr =`**{**`on`**|**`off`**}
Angle brackets < >	User-defined variable or in code samples	<directory name>

Convention	Meaning or Use	Example

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis ...	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address 0x00112975 returned 45524943h.
	An '_n' means the signal is active low	**`usr_teof_n`** is active low.

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

XILINX®

Chapter 1

Introduction

This chapter introduces and describes the LogiCORE™ IP XADC Wizard and provides related information, including recommended design experience, additional resources, technical support, and submitting feedback to Xilinx.

About the Wizard

The XADC Wizard is a Xilinx® CORE Generator™ tool that generates source Register Transfer Level (RTL code) and is designed to support both Verilog and VHDL design environments. In addition, the example design and simulation test bench delivered with the Wizard is provided in both Verilog and VHDL.

The Wizard supports Xilinx Kintex™-7, and Virtex-7 FPGA devices and instantiates the XADC primitive to implement the design.

The XADC Wizard is an ISE CORE Generator™ IP tool, included with the ISE Design Suite software. For information about system requirements and installation, see Chapter 2, Installing the Wizard.

Features

- Simple user interface
- Easy configuration of various modes and parameters
- Simple interface for channel selection and configuration
- Ability to select/deselect alarm outputs
- Ability to set alarm limits
- Calculates all the parameters and register values

Supported Devices

The Wizard supports the following FPGAs:

- Virtex®-7
- Kintex™-7

For a complete listing of supported devices, see the Release Notes for this Wizard. For more information on the 7 series FPGAs, see the *7 Series Overview*.

Provided with the Wizard

The following are provided with the Wizard:

- Documentation: This user guide
- Design Files: Verilog and VHDL
- Example Design: Verilog and VHDL
- Test Bench: Verilog and VHDL
- Constraints File: Not provided
- Simulation Model: Not provided

Recommended Design Experience

For those with less experience, Xilinx offers various training classes to help you with various aspects of designing with Xilinx FPGAs. These include classes on such topics as designing for performance and designing with multi-gigabit serial I/O. For more information, see http://www.xilinx.com/training.

Your local Xilinx sales representative can provide a closer review and estimation for your specific requirements.

Related Xilinx Documents

For detailed information and updates about the XADC Wizard, see the following documents, located on the Architecture Wizards page:

- XADC Wizard Release Notes
- *LogiCORE IP XADC Wizard User Guide*

Prior to generating the XADC Wizard, users should be familiar with the following:

- DS180: *7 Series FPGAs Overview*
- UG480: *7 Series FPGAs XADC User Guide*
- ISE software documentation: www.xilinx.com/ise

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team with expertise using the XADC Wizard.

Xilinx will provide technical support for use of this product as described in this guide. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Xilinx provides technical 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

The LogiCORE IP XADC Wizard is provided free of charge under the terms of the Xilinx End User License Agreement. The Wizard can be generated by the Xilinx ISE CORE Generator software, which is a standard component of the Xilinx ISE Design Suite. This version of the core can be generated using the ISE CORE Generator system v13.1. For more information, please visit the Architecture Wizards web page.

Information about additional Xilinx LogiCORE modules is available at the Xilinx IP Center. For pricing and availability of other Xilinx LogiCORE modules and software, please contact your local Xilinx sales representative.

Feedback

Xilinx welcomes comments and suggestions about the XADC Wizard core and the accompanying documentation.

XADC Wizard

For comments or suggestions about the XADC Wizard, please submit a WebCase from www.xilinx.com/support/clearexpress/websupport.htm. Be sure to include the following information:

- Product name
- Core version number
- Explanation of your comments, including whether you are requesting an *enhancement* (you believe something could be improved) or reporting a *defect* (you believe something is not working correctly).

Document

For comments or suggestions about the XADC Wizard, please submit a WebCase from www.xilinx.com/support/clearexpress/websupport.htm. Be sure to include the following information:

- Document title
- Document number
- Page number(s) to which your comments refer
- Explanation of your comments, including whether the case is requesting an *enhancement* (you believe something could be improved) or reporting a *defect* (you believe something isn't working correctly).

XILINX®

Chapter 2

Installing the Wizard

This chapter provides instructions for installing the LogiCORE™ IP XADC Wizard.

Tools and System Requirements

Operating Systems

Windows

- Windows XP Professional 32-bit/64-bit
- Windows Vista Business 32-bit/64-bit

Linux

- Red Hat Enterprise Linux WS v4.0 32-bit/64-bit
- Red Hat Enterprise Desktop v5.0 32-bit/64-bit ((with Workstation Option)
- SUSE Linux Enterprise (SLE) desktop and server v10.1 32-bit/64-bit

Tested Design Tools

Design Entry

- CORE Generator software 13.1
- ISE® Design Suite software 13.1

Simulation

- ISE software 13.1
- ISim 13.1
- Mentor Graphics ModelSim 6.6d
- Cadence Incisive Enterprise Simulator (IES) 10.2
- Synopsys VCS and VCS MX 2010.06

Synthesis

- XST 13.1

Check the release notes for the required Service Pack; ISE Service Packs can be downloaded from www.xilinx.com/xlnx/xil_sw_updates_home.jsp?update=sp.

Before you Begin

Before installing the Wizard, you must have a MySupport account. If you already have an account and have the software installed, go to Installing the Wizard, otherwise do the following:

1. Click **Login** at the top of the Xilinx home page then follow the onscreen instructions to create a MySupport account.

Installing the Wizard

The XADC Wizard is included with the ISE 13.1 Design suite software, and is accessed from the ISE CORE Generator tool. ISE software may be downloaded from the Xilinx Download Center, www.xilinx.com/support/download/index.htm.

For details, see the ISE Design Suite Release Notes and Installation Guide available under the Design Tools tab in the Xilinx Documentation Center.

Verifying Your Installation

Use the following procedure to verify that you have successfully installed the XADC Wizard in the CORE Generator tool.

1. Start the CORE Generator tool.
2. After creating a new Virtex -7, or Kintex-7 family project or opening an existing one, the IP core functional categories appear at the left side of the window, as shown in Figure 2-1.

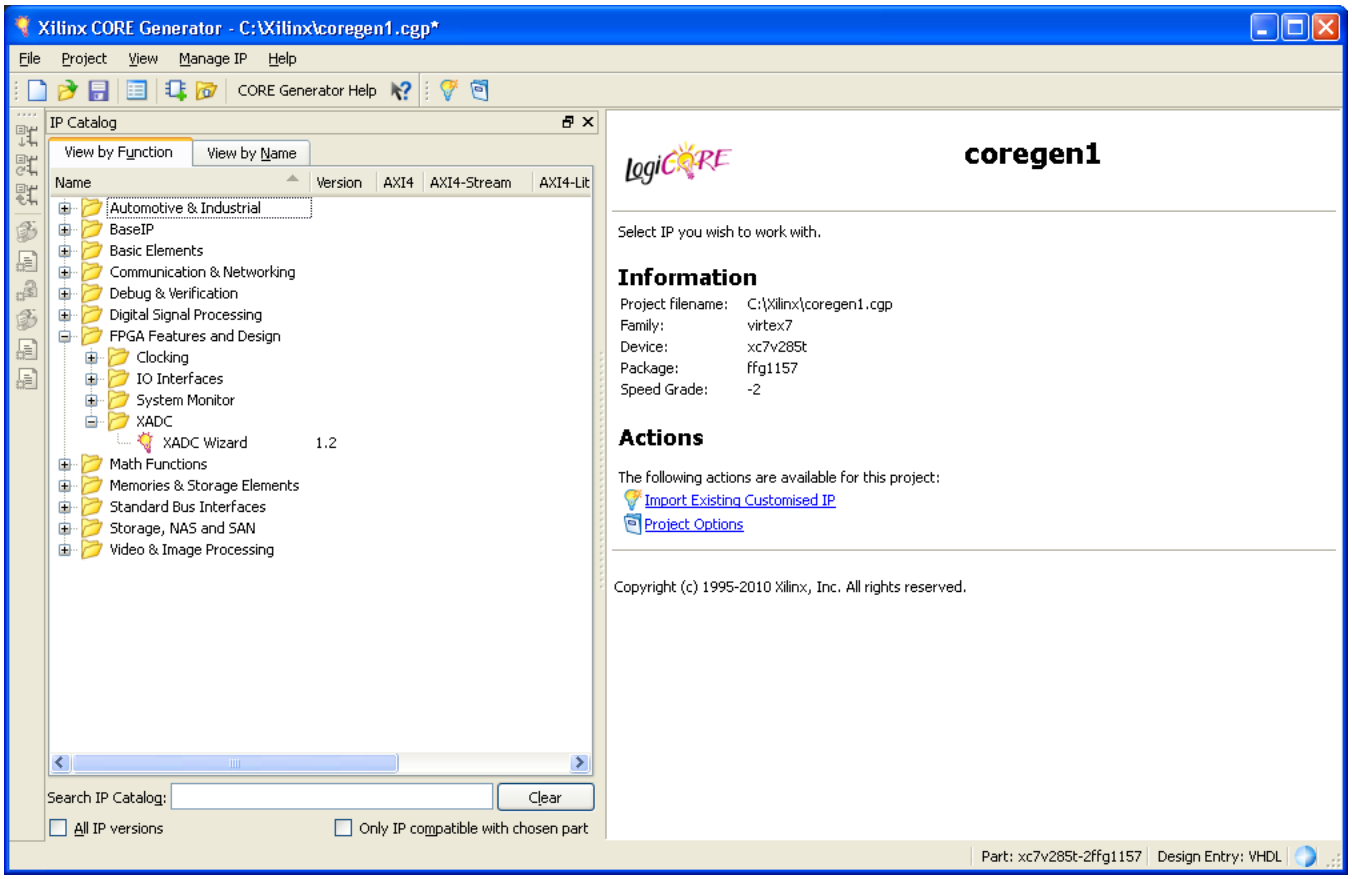

Figure 2-1: **CORE Generator Window**

3. Determine if the installation was successful by verifying that XADC Wizard appears at the following location in the Functional Categories list: `/FPGA Features and Design/XADC`.

XILINX®

Chapter 3

Running the Wizard

Overview

This chapter describes the GUI and follows the same flow required to set up the XADC primitive. Tool tips are available in the GUI for most features; simply place your mouse over the relevant text, and additional information is provided in a pop-up dialog.

Functional Overview

The XADC Wizard is an interactive graphical user interface (GUI) that instantiates a XADC block configured to your requirements. Using the wizard, users can explicitly configure the XADC to operate in the desired mode. The GUI allows you to select the channels, enable alarms, and set the alarm limits.

XADC Functional Features

Major functional XADC features can be used to determine an appropriate mode of operation. These features include:

- Analog to digital conversion
- FPGA temperature and voltage monitoring
- Generate alarms based on user set parameters

I/O Signals

Table 3-1 describes the input and output ports provided from the XADC Wizard. Availability of ports is controlled by user-selected parameters. For example, when Dynamic Reconfiguration is selected, only ports associated with Dynamic Reconfiguration are exposed. Any port that is not exposed is tied off or connected to a signal labeled as "unused" in the delivered source code.

Table 3-1: **XADC I/O Signals**

Port	Direction	Description
DI_IN[15:0]	Input	Input data bus for the dynamic reconfiguration port (DRP).
DO_OUT[15:0]	Output	Output data bus for the dynamic reconfiguration port.
DADDR_IN[6:0]	Input	Address bus for the dynamic reconfiguration port.

Table 3-1: **XADC I/O Signals *(Cont'd)***

Port	Direction	Description
DEN_IN	Input	Enable signal for the dynamic reconfiguration port.
DWE_IN	Input	Write enable for the dynamic reconfiguration port.
DCLK_IN	Input	Clock input for the dynamic reconfiguration port.
DRDY_OUT	Output	Data ready signal for the dynamic reconfiguration port.
RESET_IN	Input	Reset signal for theXADC control logic and max / min registers.
CONVST_IN	Input	Convert start input. This input is used to control the sampling instant on the ADC input and is only used in Event Mode Timing (see Event-Driven Sampling in the *7 Series FPGAs XADC User Guide*).
CONVSTCLK_IN	Input	Convert start input. This input is connected to a global clock input on the interconnect. Like CONVST, this input is used to control the sampling instant on the ADC inputs and is only used in Event Mode Timing.
VP_IN VN_IN	Input	One dedicated analog-input pair. The XADC has one pair of dedicated analog-input pins that provide a differential analog input.
VAUXP15[15:0] VAUXN15[15:0]	Inputs	16 auxiliary analog-input pairs. In addition to the dedicated differential analog-input, the XADC uses 16 differential digital-input pairs as low-bandwidth differential analog inputs. These inputs are configured as analog during FPGA configuration.
USER_TEMP_ALARM_OUT	Output	XADC temperature-sensor alarm output.
VCCINT_ALARM_OUT	Output	XADC VCCINT-sensor alarm output.
VCCAUX_ALARM_OUT	Output	XADC VCCAUX-sensor alarm output.
OT_OUT	Output	Over-Temperature alarm output.
CHANNEL_OUT[4:0]	Outputs	Channel selection outputs. The ADC input MUX channel selection for the current ADC conversion is placed on these outputs at the end of an ADC conversion.
EOC_OUT	Output	End of Conversion signal. This signal transitions to an active-High at the end of an ADC conversion when the measurement result is written to the status registers. For detailed information, see the XADC Timing section in the *7 Series FPGAs XADC User Guide*.

Table 3-1: **XADC I/O Signals *(Cont'd)***

Port	Direction	Description
EOS_OUT	Output	End of Sequence. This signal transitions to an active-High when the measurement data from the last channel in the Channel Sequencer is written to the status registers. For detailed information, see the XADC Timing section in the *7 Series FPGAs XADC User Guide*.
BUSY_OUT	Output	ADC busy signal. This signal transitions High during an ADC conversion. This signal transitions High for an extended period during calibration.
JTAGLOCKED_OUT	Output	Used to indicated that DRP port has been locked by the JTAG interface.
JTAGMODIFIED_OUT	Output	Used to indicate that a JTAG write to the DRP has occurred.
JTAGBUSY_OUT	Output	Used to indicate that a JTAG DRP transaction is in progress.
VBRAM_ALARM_OUT	Output	XADC VBRAM-sensor alarm output.
MUXADDR_OUT[4:0]	Output	Use in external multiplexer mode to decode external MUX channel.
ALARM_OUT	Output	Logic OR of alarms. Can be used to flag occurrence of any alarm.

User Attributes

The XADC functionality is configured through control registers (See the Register File Interface sections in the *7 Series FPGAs XADC User Guide*). Table 3-2 lists the attributes associated with these control registers. These control registers can be initialized via HDL by configuring attaching HDL attributes to the XADC primitive instance and configuring them according to Table 3-2. The control registers can also be initialized through the DRP at run time. The XADC Wizard simplifies the initialization of these control registers in the HDL instantiation by automatically configuring them to implement the operating behavior you specify via the IP core GUI.

Table 3-2: **XADC Attributes**

Attribute	Name	Control Reg Address	Description
INIT_40	Configuration register 0	40h	XADC configuration registers. For detailed information, see the *7 Series FPGAs XADC User Guide*.
INIT_41	Configuration register 1	41h	
INIT_42	Configuration register 2	42h	

Table 3-2: **XADC Attributes *(Cont'd)***

Attribute	Name	Control Reg Address	Description
INIT_48 to INIT_4F	Sequence registers	48h to 4Fh	Sequence registers used to program the Channel Sequencer function in the XADC. For detailed information, see the *7 Series FPGAs XADC User Guide*.
INIT_50 to INIT_5F	Alarm Limits registers	50h to 5Fh	Alarm threshold registers for the XADC alarm function. For detailed information, see the *7 Series FPGAs XADC User Guide*.
SIM_MONITOR_FILE	Simulation Analog Entry File	-	This is the text file that contains the analog input stimulus. This is used for simulation.

Setting Up the Project

Before generating the example design, set up the project as described in Creating a Directory and Setting the Project Options of this guide.

Creating a Directory

To set up the example project, first create a directory using the following steps:

1. Change directory to the desired location. This example uses the following location and directory name:

   `/Projects/xadc_example`

2. Start the Xilinx CORE Generator ™ software.

   For help starting and using the CORE Generator software, see *CORE Generator Help*, available in ISE software documentation.

3. Choose **File > New Project** (Figure 3-1).
4. Change the name of the `.cgp` file (optional).

5. Click **Save**.

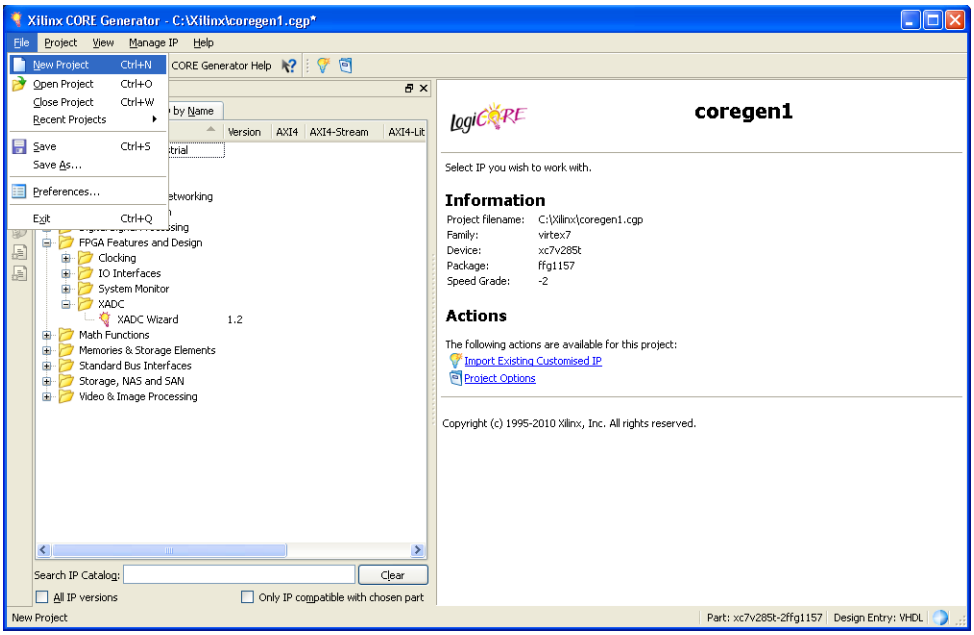

Figure 3-1: **New Project**

Setting the Project Options

Set the project options using the following steps:

1. Click **Part** in the option tree.
2. Select **Virtex-7 or Kintex™-7** from the Family list.
3. Select a device from the Device list that support XADC primitive.
4. Select an appropriate package from the Package list. This example uses the XC7V500T device (see Figure 3-2).

   Note: If an unsupported silicon family is selected, the XADC Wizard remains light grey in the taxonomy tree and cannot be customized. Only devices containing the XADC Wizard are supported by the Wizard. See the *7 Series FPGAs Overview* for a list of devices containing XADC.
5. Click **Generation** in the option tree and select either Verilog or VHDL as the output language.
6. Click **OK**.

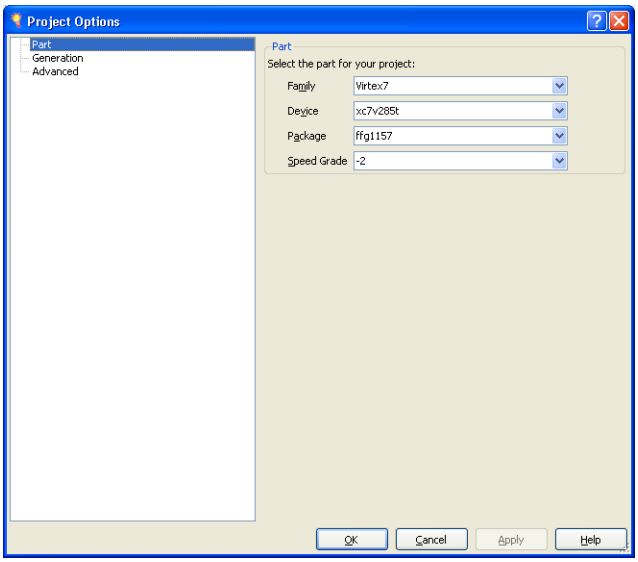

Figure 3-2: **Target Architecture Setting**

Generating the Core for Virtex-7 and Kintex-7 FPGA Devices

This section provides instructions for generating an example XADC design using the default values. The wrapper and its supporting files, including the example design, are generated in the project directory. For additional details about the example design files and directories provided with the XADC Wizard, see Chapter 4, Detailed Example Design.

1. Locate XADC Wizard in the taxonomy tree under:

   `/FPGA Features and Design/XADC`. (See Figure 3-3)

2. Double-click XADC Wizard to launch the Wizard.

After the wizard is launched, the CORE Generator tool displays a series of screens that allow you to configure the XADC Wizard.

Figure 3-3: **Locating the XADC Wizard**

XADC Setup

The XADC Wizard screen (Page 1) of the Wizard (Figure 3-4) allows you to select the component name, analog stimulus filename, startup channel mode, timing mode, and DRP timing options.

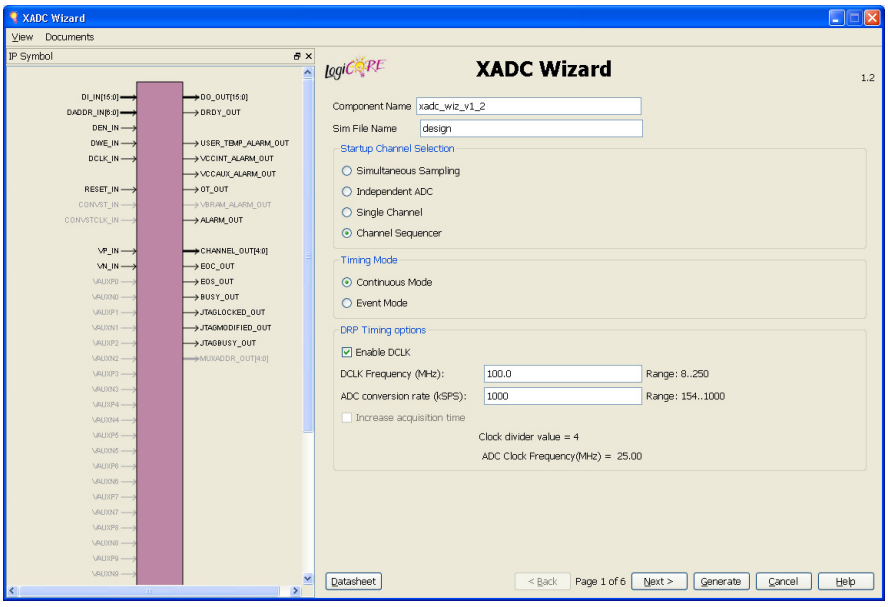

Figure 3-4: **XADC Setup - Page 1**

Component Name

User selectable component name is available. Component names must not contain any reserved words in Verilog or VHDL.

SIM File Name

Use this field to customize the name of the XADC analog stimulus file.

Startup Channel Selection

The XADC can be configured in one of the four modes listed below:

- Single Channel: In this mode, you can select only one channel to monitor.
- Channel Sequencer: Choosing this mode, allows you to select any number of channels to monitor. The channels to be used for this mode can be selected on Page 4 (Figure 3-7) and Page 5 (Figure 3-8) of the Wizard.
- Simultaneous Sampling Mode: This mode allows you to monitor two external channels simultaneously. For more information about this mode refer to the *7 Series FPGAs XADC User Guide*.
- Independent ADC Mode: This mode allows you to run the XADC in independent mode. Here, the XADC independently monitors the externals channels and at the same time monitors the FPGA Voltages and temperature.

Timing Mode

The XADC can operate in two timing modes:

- Continuous Mode: In this mode, the XADC continues to sample and convert the selected channel/channels.
- Event Mode: This mode requires an external trigger event, CONVST or CONVSTCLK, to start a conversion on the selected channel. Event Mode should only be used with external channels.

DRP Timing Options

The XADC clock (ADCCLK) is derived from the dynamic reconfiguration port (DRP) clock DCLK. The XADC supports a DRP clock frequency of up to 250MHz. The XADC can also operate in absence of DCLK. For more information on the DRP see the *7 Series FPGAs XADC User Guide*.

The ADCCLK clock, should be in the range of 4-26 MHz. To support this lower frequency clock the XADC has an internal clock divider. The GUI allows an external DCLK frequency and required ADC conversion rate (maximum 1 Msps) to be specified. Based on the value of DCLK clock, the wizard then calculates the appropriate clock divider value based on the values of DCLK clock and ADC conversion rate, the wizard calculates the appropriate clock divider value.

The wizard also displays the ADC Clock frequency value and the actual conversion rate of the ADC.

I/O Ports

The I/O Port Selection screen (Page 2) of the Wizard (Figure 3-5) allows you to select the I/O ports on the XADC primitive.

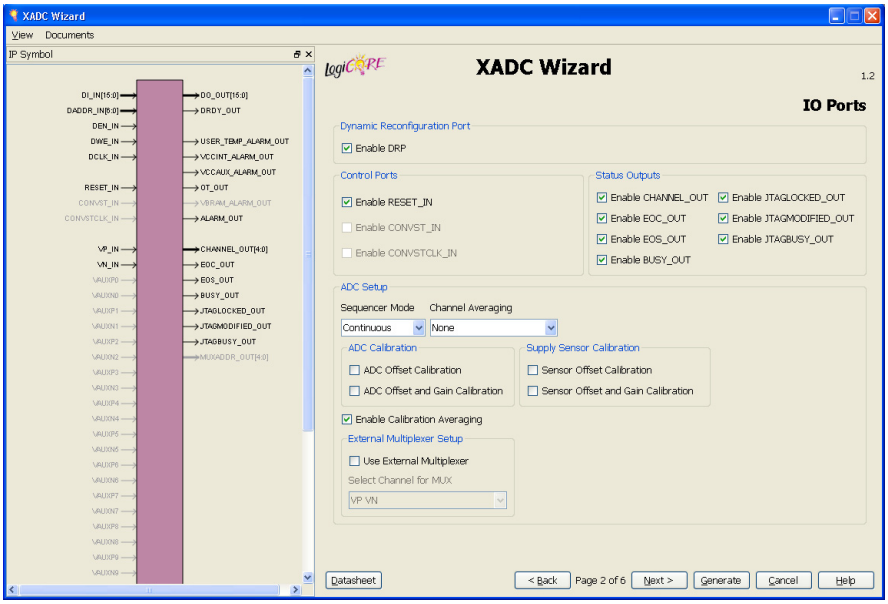

Figure 3-5: **I/O Ports - Page 2**

Dynamic Reconfiguration Port

This port is the FPGA fabric interface for XADC. It facilitates access to the register file interface of the XADC. The XADC control registers can be read or written using this port. This port can be enabled only when DCLK clock is present.

Control Ports

This section allows you to select control input ports:

- RESET_IN allows an external input reset signal to be connected to the XADC
- CONVST_IN and/or CONVSTCLK_IN as trigger sources for Event Mode Timing

Status Outputs

A number of output status signals are also provided to facilitate interfacing of the XADC to a user design. See UG480: *7 Series FPGAs XADC User Guide* for more information.

ADC Setup

If the XADC is configured for Channel Sequencer, Simultaneous Sampling or Independent ADC mode, you can choose the required sequencer mode. The available options are Continuous, One-pass or Default mode.

The Channel Averaging drop-down menu allows you to select the required averaging value. The available options are None, 16, 64 and 256.

You may select the type of ADC Calibration and/or Supply Sensor Calibration by checking the respective checkboxes. Calibration Averaging is enabled by default in XADC. You can disable this by deselecting the box.

External Multiplexer Setup

The XADC supports a new timing mode that allows users to use an external analog multiplexer in situations where FPGA IO resources may be limited or auxiliary analog IO are more valuable when used to implement another interface.

You can opt to use this feature by checking the box against Use External Mux. If checked, it is necessary to specify the external channel to which the Mux will connect. Select this channel using the drop-down menu.

Alarm Setup

The Alarm Setup screen (Page 3) of the Wizard (Figure 3-6) allows alarm outputs to be enabled for the on-chip sensors. If a measurement of an on-chip sensor lies outside the specified limits, then a logic output will go active if enabled. For a detailed description of the alarm functionality see the *7 Series XADC User Guide*.

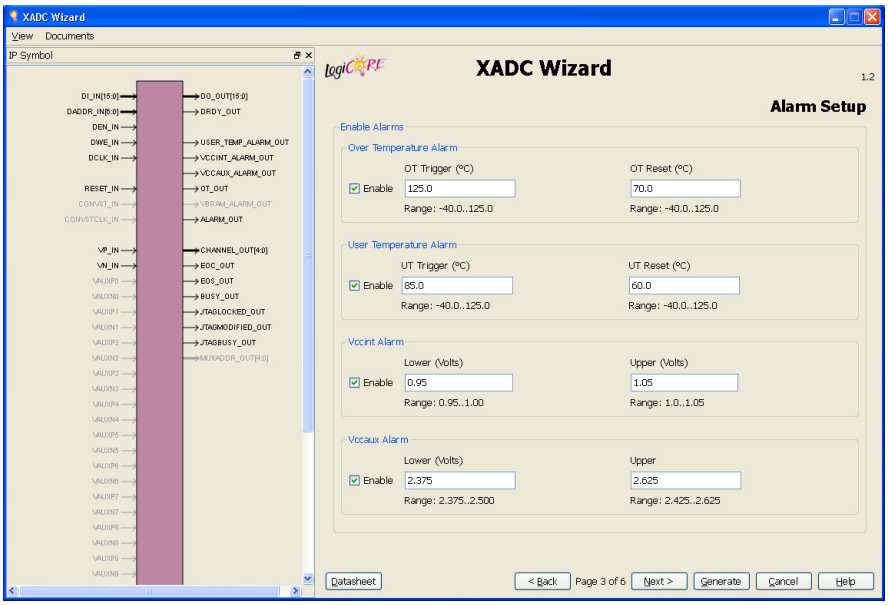

Figure 3-6: **Alarm Setup - Page 3**

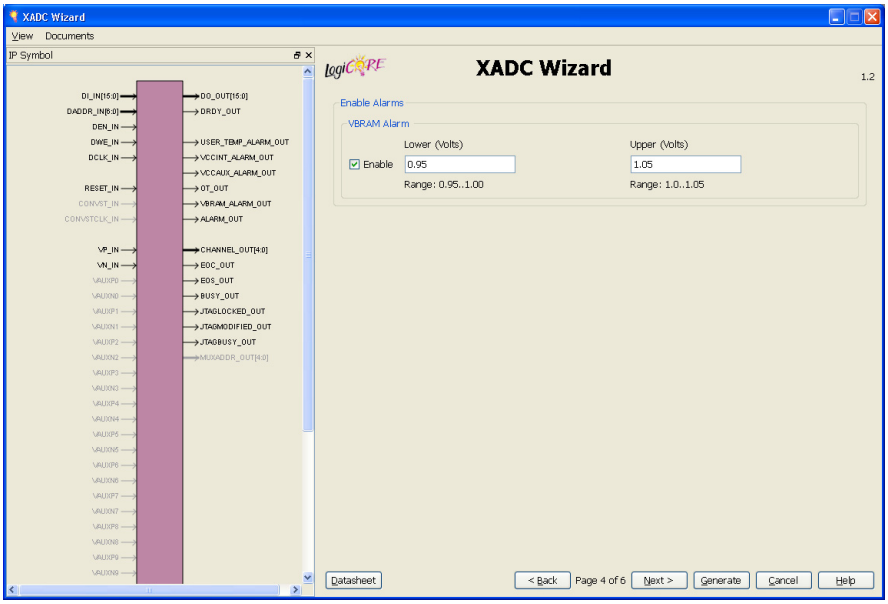

Figure 3-7: **Alarm Setup - Page 4**

Enable Alarms

Use the checkboxes to enable alarm logic outputs. The five options are:

- Over temperature alarm
- User temperature alarm
- V_{CCNT} alarm
- V_{CCAUX} alarm
- V_{BRAM} alarm

Temperature Alarm Limits

Trigger and Reset levels for temperature alarm output can be entered using these fields. You can set both; the trigger as well as reset levels for the OT alarm.

VCCINT, VCCAUX, and, VBRAM Limits

Both upper and lower alarm thresholds can be specified for the on-chip power supplies. If the measured value moves outside these limits the alarm logic output will go active. The alarm output is reset once a measurement inside these limits is generated. The default limits in the GUI represent ± 5% on the nominal supply value.

Channel Sequencer Setup P1 and Setup P2

Channel Sequencer Setup P1 (Figure 3-7) and Setup P2 (Figure 3-8) screens of the Wizard are used to configure the XADC sequence registers when the XADC is configured in Channel Sequencer, Simultaneous sampling or Independent ADC mode. All the possible channels that can be included in the sequence are listed in the table spread across screens 5 and 6 (Figure 3-7 and Figure 3-8) of the Wizard:

- Use the Channel Sequencer Setup P1 and P2 screen to select Channels for monitoring, enable Averaging for selected channels, enable Bipolar mode for external channels and increase the Acquisition time for the selected channels.
- In the case of Simultaneous sampling mode, selecting channel Vauxp[0]/Vauxn[0] would automatically select channel Vauxp[8]/Vauxn[8]. Similarly selecting channel Vauxp[1]/Vauxn[1] would select channel Vauxp[9]/Vauxn[9] and so on.
- In case of Independent ADC mode, only external channels are listed and can be user-selected.

For more information about the simultaneous sampling mode and Independent ADC mode, see the *7 Series XADC User Guide*.

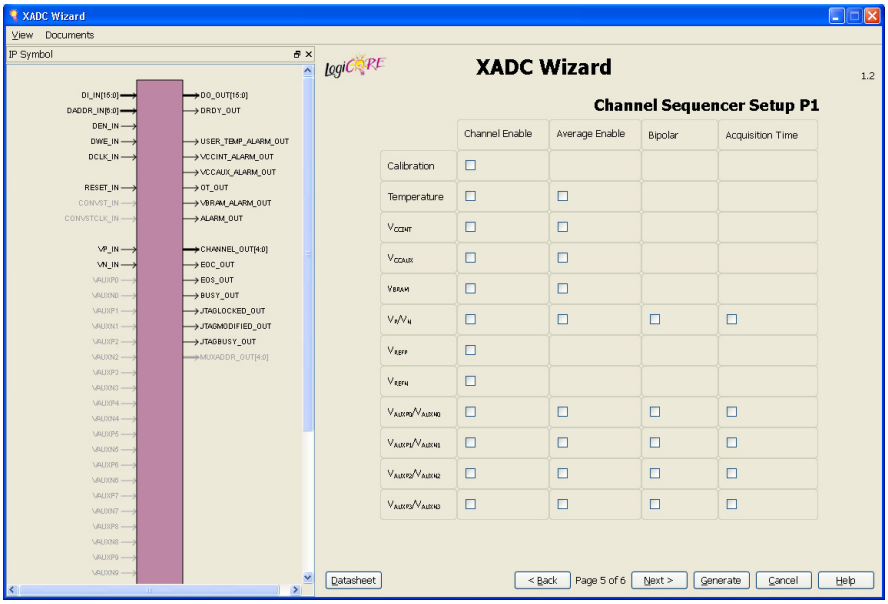

Figure 3-8: **Channel Sequencer Setup P1 - Page 5**

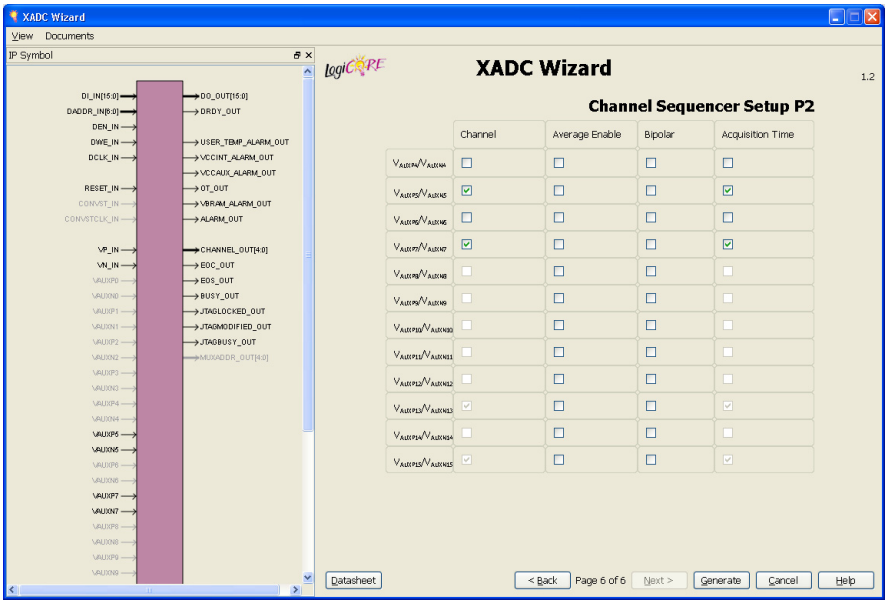

Figure 3-9: **Channel Sequencer Setup P2 - Page 6**

Single Channel Mode

If the Single Channel Mode operation (see Startup Channel Selection, page 24) is selected on Page 1 of the Wizard (Figure 3-4, page 23), the Single Channel Setup is displayed on Page 4 of the Wizard (Figure 3-9). Page 4 allows you to select the channel for measurement and the analog input mode if the channel is an external analog input (that is, unipolar or bipolar).

The columns Channel Enable, Average Enable and Increase Acquisition Time are disabled and are shown only information and ease.

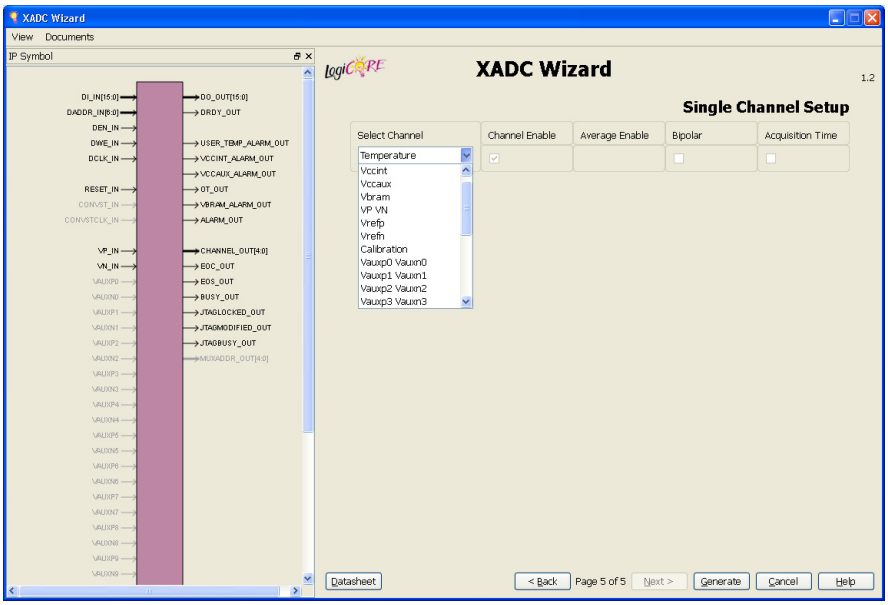

Figure 3-10: **Single Channel Setup - Page 5**

Generating the HDL Wrapper

After selecting the configuration options, click Generate on the final Wizard screen to generate the HDL wrapper and other Wizard outputs.

The output files are placed in the project directory you selected or created when setting up a new CORE Generator project.

XILINX®

Chapter 4

Detailed Example Design

This chapter provides detailed information about the example design, including a description of files and the directory structure generated by the ISE CORE Generator™ software, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

Directory and File Structure

- **<project directory>**
  Top-level project directory; name is user-defined
  - <project directory>/<component name>
    Core release notes file
    - <component name>/doc
      Product documentation
    - <component name>/example design
      Verilog or VHDL design files
    - <component name>/implement
      Implementation script files
      - implement/results
        Results directory, created after implementation scripts are run, and contains implement script results
    - <component name>/simulation
      Simulation scripts
      - simulation/functional
        Functional simulation files

Directory and File Contents

The XADC Wizard directories and their associated files are defined in the following sections.

<project directory>

The <project directory> contains all the CORE Generator software project files.

Table 4-1: **Project Directory**

Name	Description
<project_dir>	
<component_name>.v[hd]	Verilog or VHDL simulation model.
<component_name>.xco	CORE Generator tool project-specific option file; can be used as an input to the CORE Generator tool.
<component_name>_flist.txt	List of files delivered with the core.
<component_name>.{veo|vho}	VHDL or Verilog instantiation template.

Back to Top

<project directory>/<component name>

The <component name> directory contains the readme file provided with the core, which may include last-minute changes and updates.

Table 4-2: **Component Name Directory**

Name	Description
<project_dir>/<component_name>	
xadc_wiz_readme.txt	Core readme file.

Back to Top

<component name>/doc

The doc directory contains the PDF documentation provided with the core.

Table 4-3: **Doc Directory**

Name	Description
<project_dir>/<component_name>/doc	
ug772_xadc_wizard.pdf	*LogiCORE IP XADC Wizard User Guide*

Back to Top

<component name>/example design

The example design directory contains the example design files provided with the core.

Table 4-4: **Example Design Directory**

Name	Description
<project_dir>/<component_name>/example_design	
`<component_name>_exdes.v(hd)`	Verilog and VHDL top-level example design file.

Back to Top

<component name>/implement

The implement directory contains the core implementation script files.

Table 4-5: **Implement Directory**

Name	Description
<project_dir>/<component_name>/implement	
`implement.bat` `implement.sh`	Windows and Linux based implementation scripts.
`xst.prj`	The XST project file for the example design; it lists all of the source files to be synthesized.
`xst.scr`	The XST script file for the example design that is used to synthesize the core, called from the implement script described above.

Back to Top

implement/results

The results directory is created by the implement script, after which the implement script results are placed in the results directory.

Table 4-6: **Results Directory**

Name	Description
<project_dir>/<component_name>/implement/results	
`Implement script result files.`	

Back to Top

<component name>/simulation

The simulation directory contains the simulation scripts provided with the core.

Table 4-7: **Simulation Directory**

Name	Description
<project_dir>/<component_name>/simulation	
`<component_name>_tb.v[hd]`	Demonstration test bench.

Back to Top

simulation/functional

The functional directory contains functional simulation scripts provided with the core.

Table 4-8: **Functional Directory**

Name	Description
<project_dir>/<component_name>/simulation/functional	
`simulate_isim.sh` `simulate_isim.bat`	Linux and Windows simulation scripts for ISIM simulator.
`simulate_mti.do`	Modelsim simulation script.
`simulate_ncsim.sh`	Linux script for running simulation using Cadence Incisive Enterprise Simulator (IES).
`simulate_vcs.sh`	Linux script for running simulation using VCS MX.

Back to Top

Implementation Scripts

The implementation script is either a shell script or batch file that processes the example design through the Xilinx tool flow. It is located at:

LINUX and UNIX

```
<project_dir>/<component_name>/implement/implement.sh
```

Windows

```
<project_dir>/<component_name>/implement/implement.bat
```

The implement script performs the following steps:

- Synthesizes the HDL example design files using XST
- Runs Ngdbuild to consolidate the core netlist and the example design netlist into the NGD file containing the entire design
- Maps the design to the target technology
- Place-and-routes the design on the target device
- Performs static timing analysis on the routed design using Timing Analyzer (TRCE)
- Generates a bitstream
- Enables Netgen to run on the routed design to generate a VHDL or Verilog netlist (as appropriate for the Design Entry project setting) and timing information in the form of SDF files

The Xilinx tool flow generates several output and report files. These are saved in the following directory which is created by the implement script:

```
<project_dir>/<component_name>/implement/results
```

Simulation Scripts

Functional Simulation

The test scripts are a ModelSim, Cadence IES, VCS, VCS MX, or ISim macro that automate the simulation of the test bench. They are available from the following location:

```
<project_dir>/<component_name>/simulation/functional/
```

The test script performs the following tasks:

- Compiles the structural UniSim simulation model
- Compiles HDL Example Design source code
- Compiles the demonstration test bench
- Starts a simulation of the test bench
- Opens a Wave window and adds signals of interest
- Runs the simulation to completion

Example Design

Top Level Example Design

The following files describe the top-level example design for the XADC Wizard core.

VHDL

```
project_dir>/<component_name>/example_design/<component_name>_exdes.vhd
```

Verilog

```
project_dir>/<component_name>/example_design/<component_name>_exdes.v
```

The example design, instantiates the XADC core that is generated by the wizard.

Demonstration Test Bench

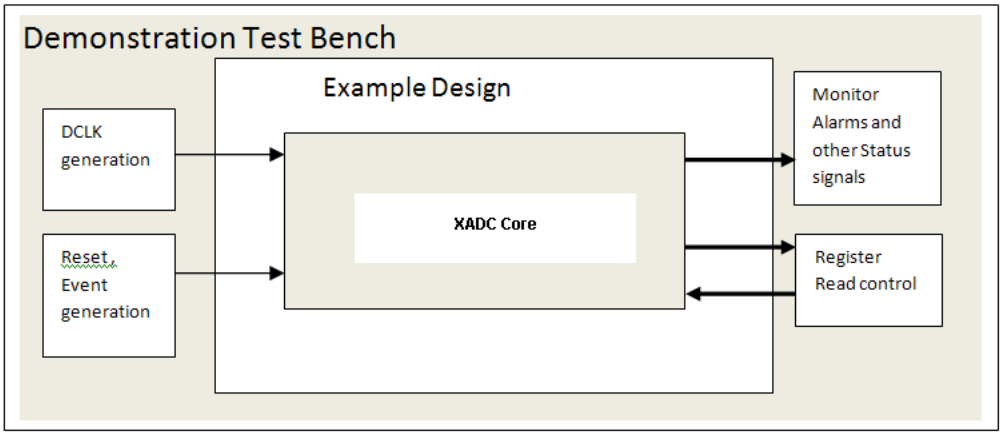

Figure 4-1: **Demonstration Test Bench for the XADC Wizard and Example Design**

The following files describe the demonstration test bench.

VHDL

```
project_dir>/<component_name>/simulation/<component_name>_tb.vhd
```

Verilog

```
project_dir>/<component_name>/simulation/<component_name>_tb.v
```

The demonstration test bench is a simple VHDL or Verilog program to exercise the example design and the core.

The demonstration test bench performs the following tasks:

- Generates the input DCLK clock signal
- Applies a reset to the example design
- Monitors the alarms and other status outputs
- Reads the respective registers when a conversion is complete