LogiCORE IP XADC Wizard v2.2

User Guide

UG772 July 25, 2012

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/11	1.0	Initial Xilinx release.
06/22/11	1.1	Updated for 13.2 release.
08/17/11	1.2	Updated for text edits.
01/18/12	1.3	Updated screen captures to core version 1.4 and ISE version to 13.4.
04/24/12	1.4	Core version 2.1. ISE version 14.1.
07/25/12	1.5	Updated with information about Core version 2.2, released in Vivado version 2012.2 only.

Table of Contents

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 generates Verilog or VHDL Register Transfer Level (RTL) source code to configure the XADC primitive in Xilinx 7 series FPGAs. An example design and simulation test bench demonstrate how to integrate the core into user designs.

The XADC Wizard is included with the ISE and Vivado Design Suite software. For information about system requirements and installation, see Chapter 2, Installing the Wizard. Version 2.2 of the XADC Wizard is a Vivado-only update. The information in this version of the user guide covers use of the wizard in Vivado Design Suite only.

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:

- 7 Series FPGAs

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 FPGAs Overview* [Ref 1].

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: 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 www.xilinx.com/training.

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

Technical Support

For technical support, go to www.xilinx.com/support to file a WebCase. Questions are routed to a team with expertise using the XADC Wizard. Additional support resources available at this site include Answers, Documentation, Downloads, and Forums.

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 Vivado IP catalog, which is a standard component of the Xilinx Vivado Design Suite. This version of the core can be generated using the Vivado IP catalog 2012.2. For more information, 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, 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 core, 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).

Documentation

For comments or suggestions about the XADC Wizard documentation, 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 is working incorrectly).

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

- Vivado™ Design Suite 2012.2

Simulation

- Vivado simulator 2012.2
- Mentor Graphics ModelSim
- Cadence Incisive Enterprise Simulator (IES)
- Synopsys VCS and VCS MX

For the supported versions of the tools, see the ISE Design Suite 14: Release Notes Guide.

Synthesis

- Vivado synthesis 2012.2
- Synopsys Synplify PRO

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,

- Click **Login** at the top of the Xilinx home page then follow the on screen instructions to create a MySupport account.

Installing the Wizard

Version 2.2 of the XADC Wizard is included with the Vivado Design Suite 2012.2, and is accessed from the Vivado IP catalog within that toolset. v2012.2 of the Vivado Design Suite can be downloaded from the Xilinx Download Center, www.xilinx.com/support/download/index.htm.

For details, see the Vivado Design Suite Release Notes and Installation Guide.

Verifying Your Installation

Use the following procedure to verify that you have successfully installed the XADC Wizard in the Vivado tools.

1. Start Vivado.
2. After creating a new 7 series family project or opening an existing one, the IP catalog appears at the right side of the window, as shown in Figure 2-1.

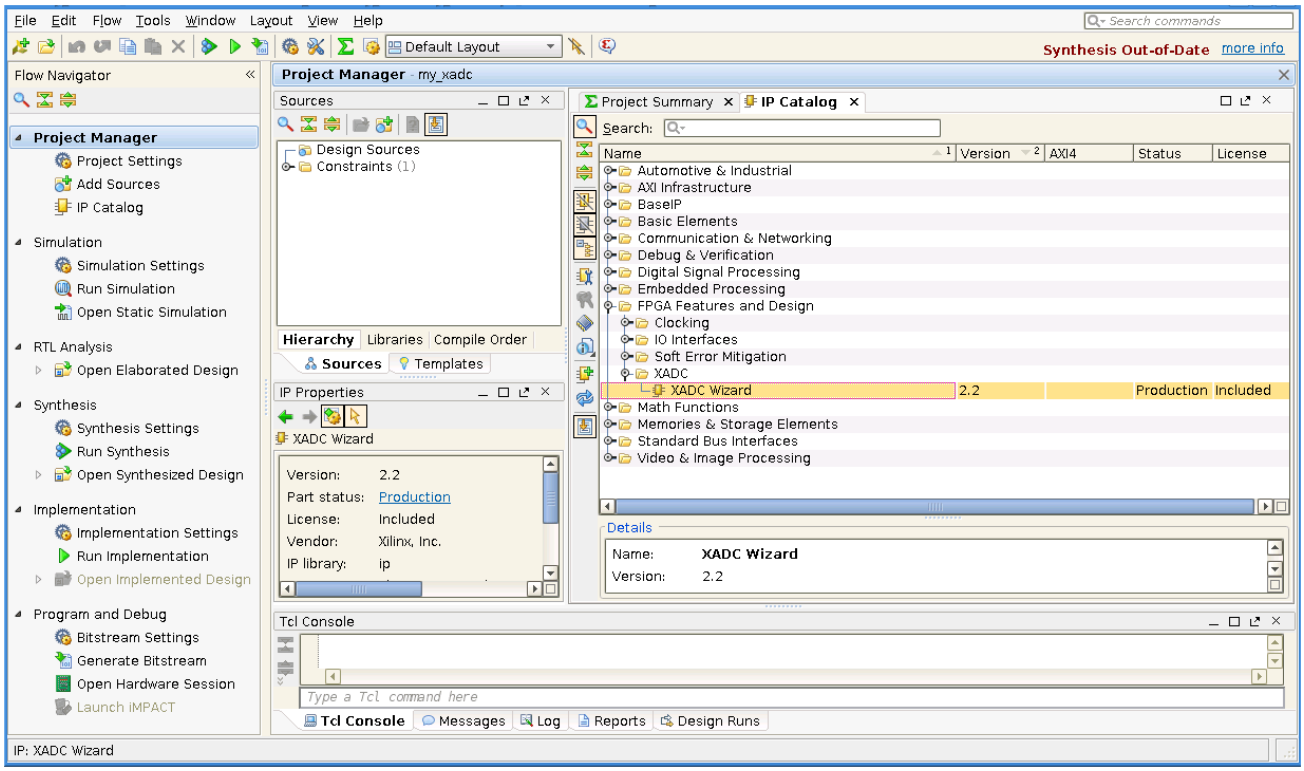

Figure 2-1: **Vivado IP Catalog**

3. Determine if the installation was successful by verifying that XADC Wizard appears at the following location in the catalog 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 using v2.2 of the wizard. Tool tips are available in the GUI for most features; 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 the XADC 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* [Ref 2].
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* [Ref 2].

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* [Ref 2].
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 indicate 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.
VCCPINT_ALARM_OUT	Output	XADC VCCPINT-sensor alarm output
VCCPAUX_ALARM_OUT	Output	XADC VCCPAUX-sensor alarm output
VCCDDRO_ALARM_OUT	Output	XADC VCCDDRO-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* [Ref 2]). Table 3-2 lists the attributes associated with these control registers. These control registers can be initialized using 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 using 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* [Ref 2].
INIT_41	Configuration register 1	41h	
INIT_42	Configuration register 2	42h	
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* [Ref 2].
INIT_50 to INIT_5F	Alarm Limits registers	50h to 5Fh	Alarm threshold registers for the XADC alarm function. For detailed information, seethe *7 Series FPGAs XADC User Guide* [Ref 2].
SIM_MONITOR_FILE	Simulation Analog Entry File	-	This is the text file that contains the analog input stimulus. This is used for simulation.
SIM_DEVICE	Device family information	-	Specifies the device family. For 7 Series devices, this value is "7Series".

Setting Up the Project

This section describes how to set up a project in the Vivado Design Suite flow. For information on setting up a project in the ISE Design Suite flow, please consult the user guide for XADC Wizard at www.xilinx.com/support/documentation/ip_documentation/xadc_wiz/v2_1/ug772_xadc_wiz.pdf.

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 Vivado™.

   For help starting and using Vivado, see the *Vivado Help*, available in the *Vivado documentation* [Ref 3].

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

5. Create project with default settings.

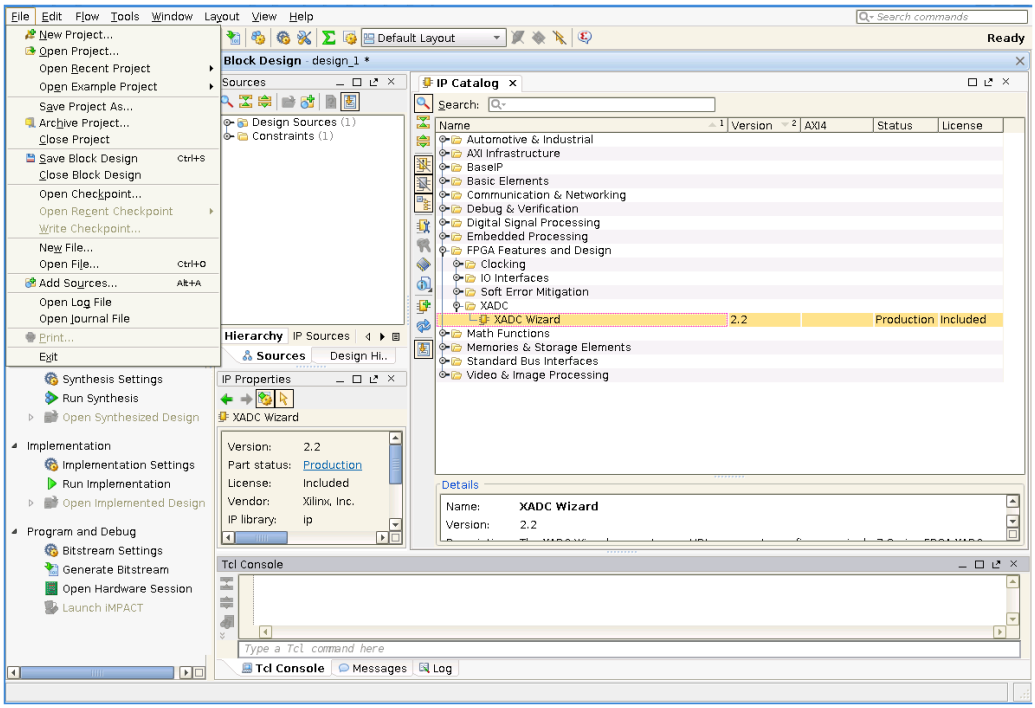

Figure 3-1: **New Project**

Setting the Project Options

Set the project options using the following steps:

1. Click **Project Part** in the option tree.
2. Select a **7 series** FPGA 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 XC7K235T device (see Figure 3-2).

   Note: If an unsupported silicon family is selected, the XADC Wizard remains light gray in the taxonomy tree and cannot be customized. Only devices containing the XADC are supported by the Wizard. See the *7 Series FPGAs Overview* [Ref 1] for a list of devices containing XADC.
5. Select either Verilog or VHDL as the target language.
6. Click **OK**.

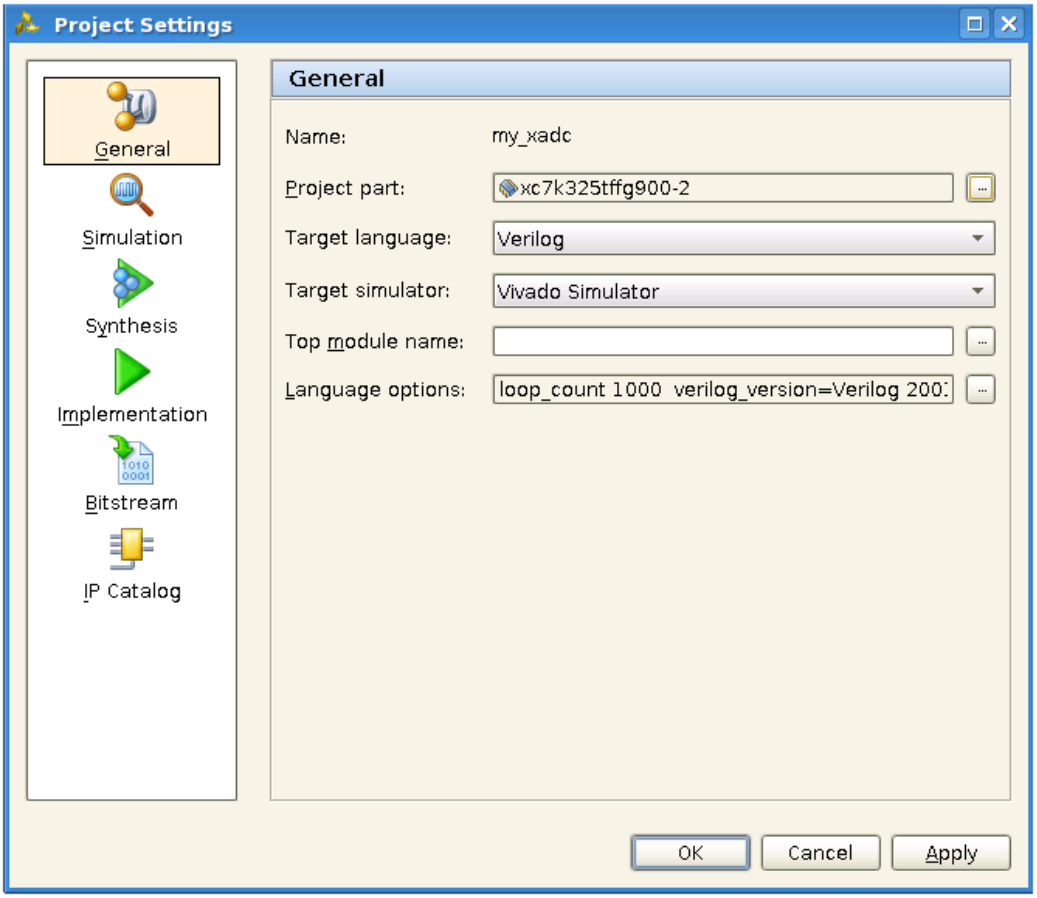

Figure 3-2: **Target Architecture Setting**

Generating the Core for 7 Series 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 the XADC Wizard in the taxonomy tree under:

   `/FPGA Features and Design/XADC`. (See Figure 3-1)

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

After the wizard is launched, the IP catalog displays a series of screens that allow you to configure the XADC Wizard.

XADC Setup

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

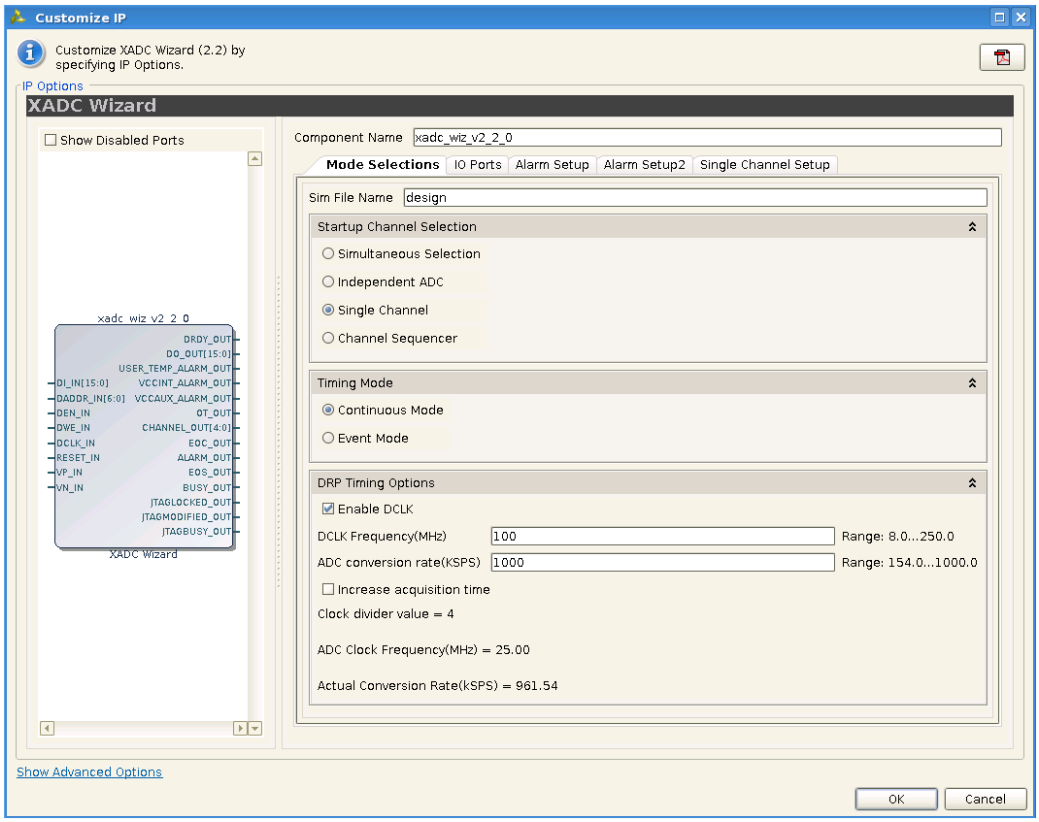

Figure 3-3: **XADC Setup - Page 1**

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.

Start-up Channel Selection

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

- **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 5 (Figure 3-7, page 25) and Page 6 (Figure 3-8, page 26) of the Wizard.
- **Simultaneous Sampling Mode**: This mode allows you to monitor two external channels simultaneously. For more information about this mode see the *7 Series*

FPGAs XADC User Guide [Ref 2].

- **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* [Ref 2].

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 3-4) allows you to select the I/O ports on the XADC primitive.

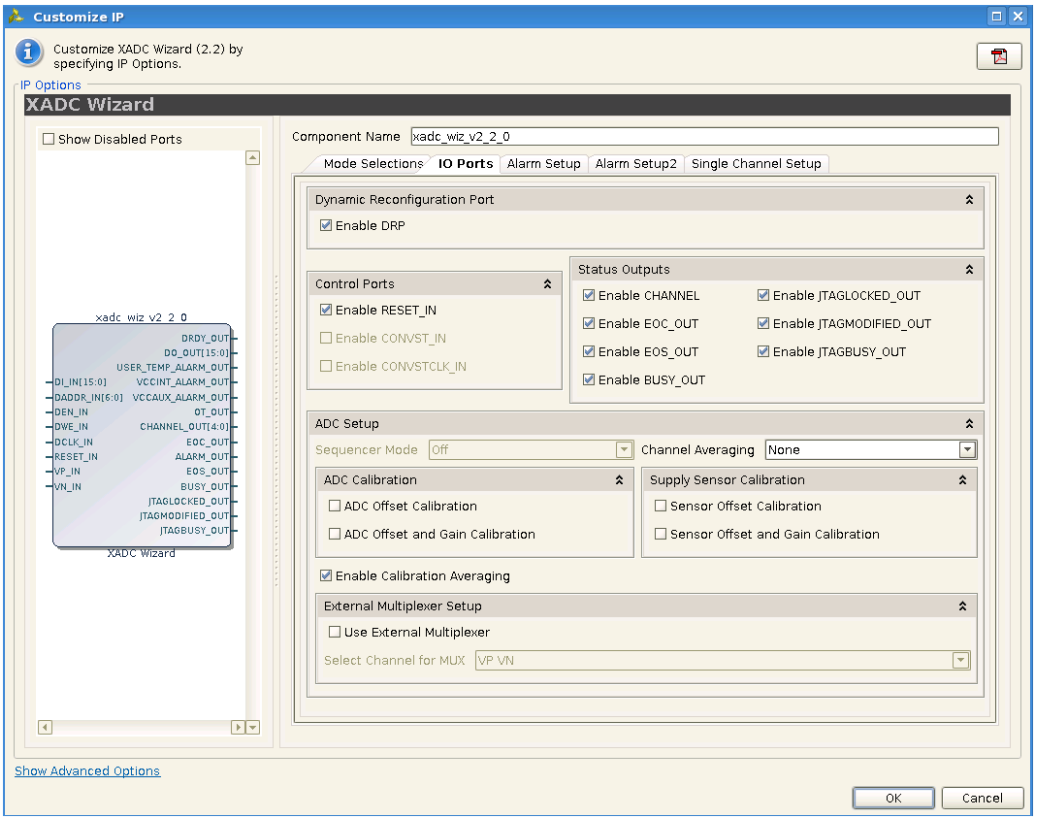

Figure 3-4: **I/O Ports - Page 2**

Dynamic Reconfiguration Port

This port is the FPGA logic 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

Output status signals are also provided to facilitate interfacing of the XADC to a user design. See the *7 Series FPGAs XADC User Guide* [Ref 2] 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 can 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 I/O resources might be limited or auxiliary analog I/O 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 connects. Select this channel using the drop-down menu.

Alarm Setup

The Alarm Setup screens of the Wizard (Figure 3-5 and Figure 3-6) allow the 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 goes active if enabled. For a detailed description of the alarm functionality see the *7 Series FPGAs XADC User Guide* [Ref 2].

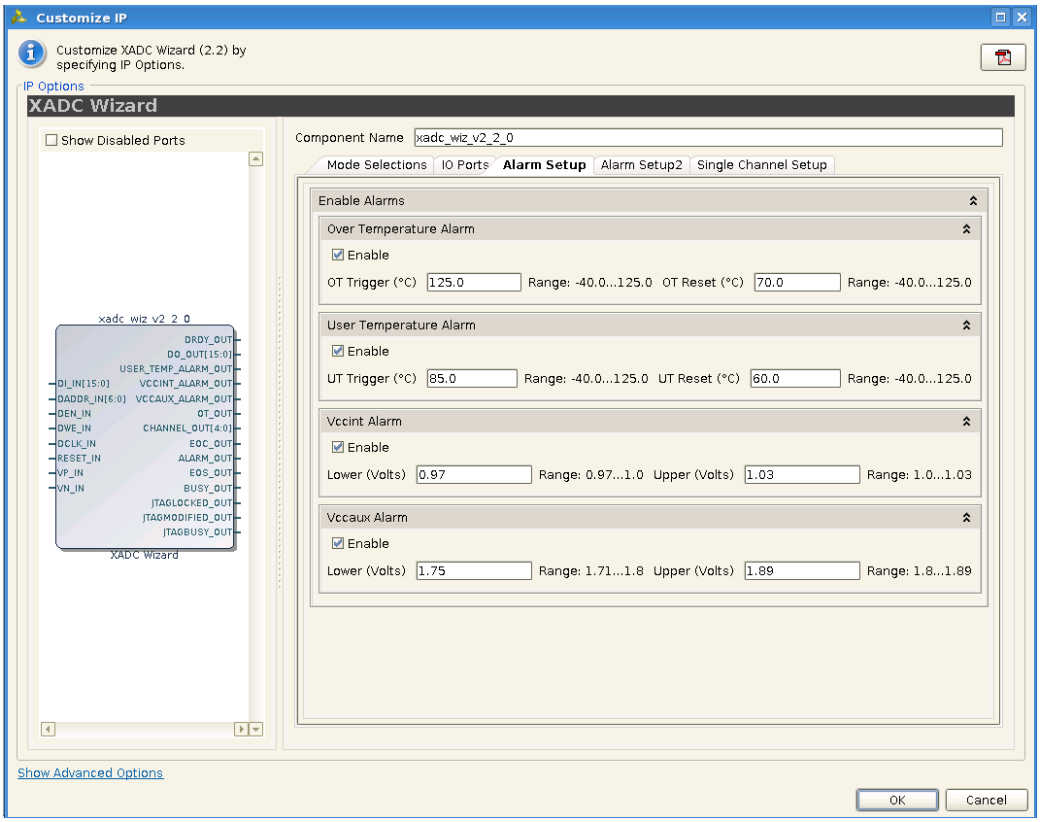

Figure 3-5: **Alarm Setup P1 - Page 3**

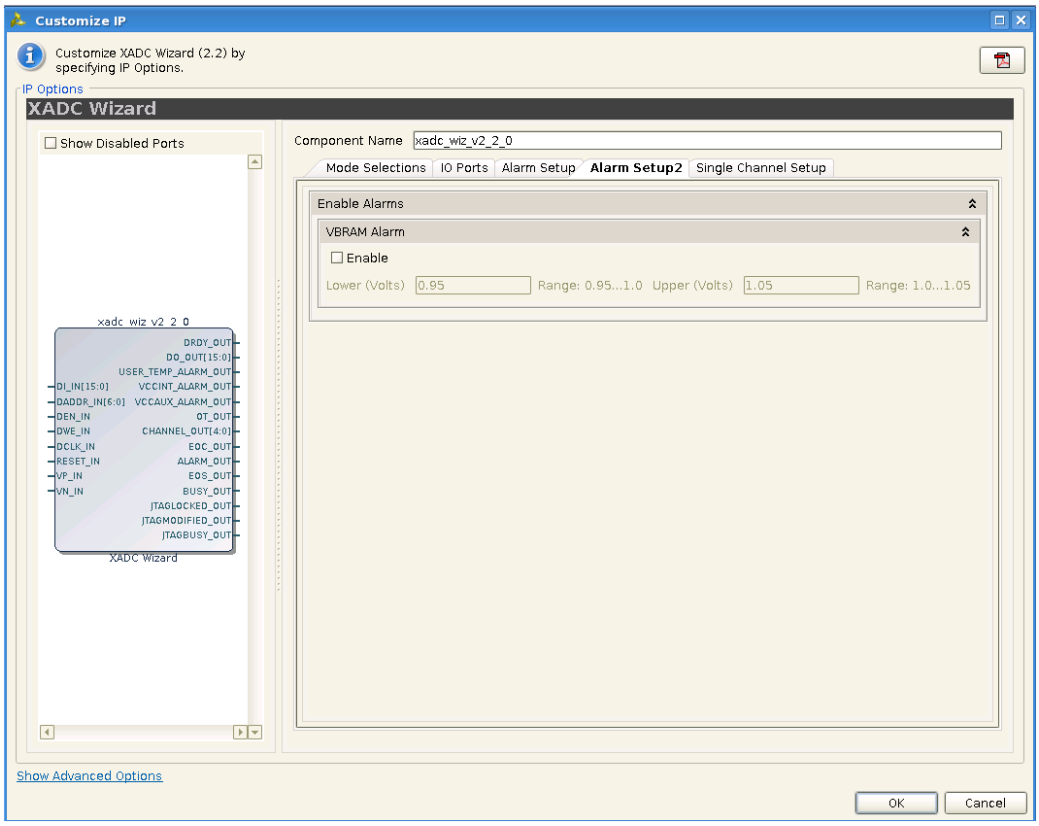

Figure 3-6: **Alarm Setup P2 - Page 4**

Enable Alarms

Use the checkboxes to enable alarm logic outputs. The eight 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 goes active. The alarm output is reset when 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 FPGAs XADC User Guide* [Ref 2].

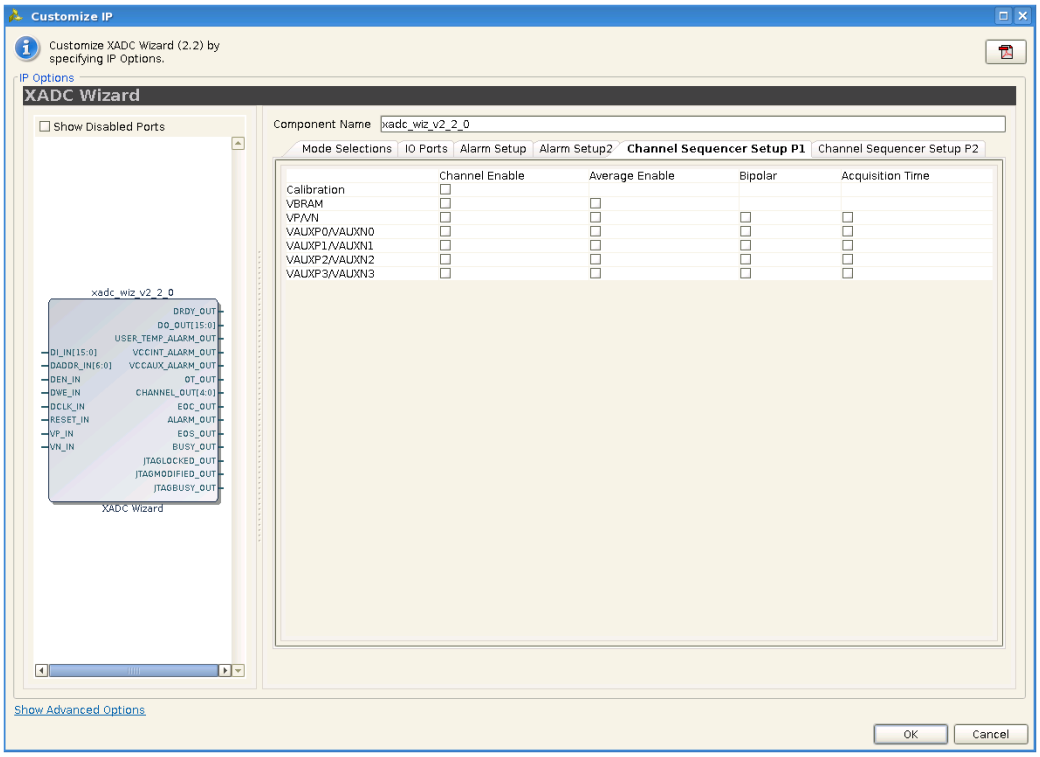

Figure 3-7: **Channel Sequencer Setup P1 - Page 5**

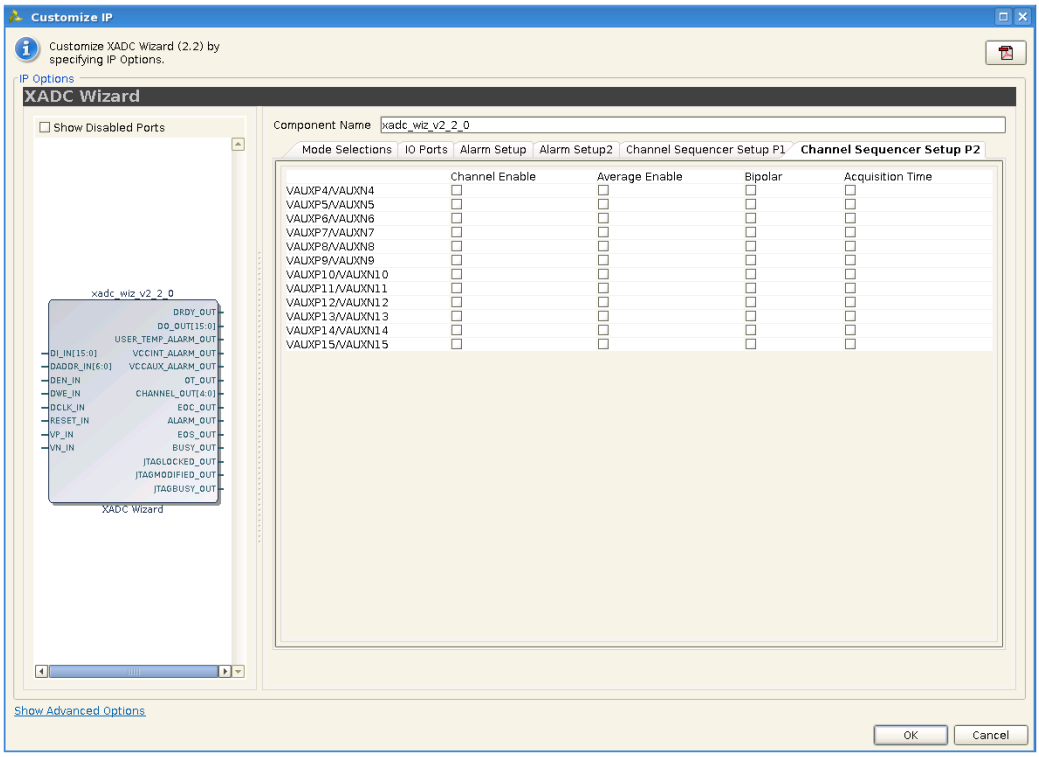

Figure 3-8: **Channel Sequencer Setup P2 - Page 6**

Single Channel Mode

If the Single Channel Mode operation (see Start-up Channel Selection, page 19) is selected on Page 1 of the Wizard (Figure 3-3, page 19), the Single Channel Setup is displayed on Page 5 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.

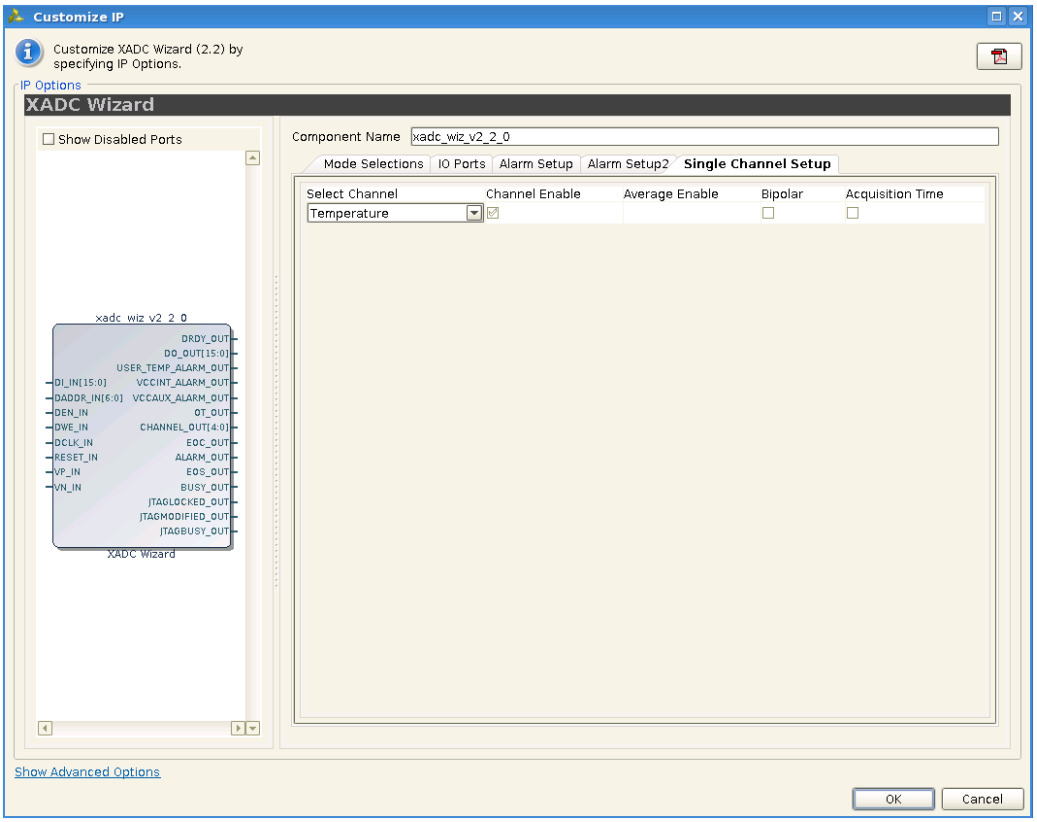

Figure 3-9: **Single Channel Setup - Page 5**

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_name>/<project_name>.srcs/sources_1/ip/<component_name>/` directory you selected or created when setting up a new Vivado 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 Xilinx® Vivado™ tools, 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_name>/<project_name>.srcs/sources_1/ip/**
  Top-level project directory; name is user-defined
  - <project_name>/<project_name>.srcs/sources_1/ip/<component name>
    Core release notes file
    - <component name>/doc
      Product documentation
    - <component name>/example design
      Verilog or VHDL design files
    - <component name>/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
      - simulation/timing
        Timing simulation files

Directory and File Contents

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

<project_name>/<project_name>.srcs/sources_1/ip/

The <project_name>/<project_name>.srcs/sources_1/ip/ contains all the Vivado project files.

Table 4-1: **Project Directory**

Name	Description
<project_name>/<project_name>.srcs/sources_1/ip/	
<component_name>.v[hd]	Verilog or VHDL simulation model.
<component_name>.xci	Vivado project-specific option file; can be used as an input to the Vivado tools.
<component_name>_flist.txt	List of files delivered with the core.
<component_name>.{veo|vho}	VHDL or Verilog instantiation template.
<component_name>.xdc	Constraint file for core.

Back to Top

<project_name>/<project_name>.srcs/sources_1/ip/<component name>

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

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

Name	Description
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>	
xadc_wiz_v2_2_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_name>/<project_name>.srcs/sources_1/ip/<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_name>/<project_name>.srcs/sources_1/ip/<component_name>/example_design	
<component_name>_exdes.v(hd)	Verilog and VHDL top-level example design file.
<component_name>_exdes.xdc	Constraint file for example design.

Back to Top

<component name>/implement/results

The results directory should be created by the user and implementation files should be copied to the results directory before running timing simulations.

Table 4-5: **Results Directory**

Name	Description
<project_name>/<project_name>.srcs/sources_1/ip/<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 4-6: **Simulation Directory**

Name	Description
<project_name>/<project_name>.srcs/sources_1/ip/<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 4-7: **Functional Directory**

Name	Description
`<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/simulation/functional`	
`simulate_xsim.sh` `simulate_xsim.bat`	Linux and Windows simulation scripts for Vivado 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

simulation/timing

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

Table 4-8: **Functional Directory**

Name	Description
`<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/simulation/timing`	
`simulate_xsim.sh` `simulate_xsim.bat`	Linux and Windows simulation scripts for Vivado simulator.
`simulate_mti.do`	ModelSim simulation script.
`simulate_ncsim.sh`	Linux script for running simulation using Cadence Incisive Enterprise Simulator (IES).

Back to Top

Simulation Scripts

Functional Simulation

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

```
<project_name>/<project_name>.srcs/sources_1/ip/<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

Timing Simulation

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

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/
simulation/timing/
```

The test script performs the following tasks:

- Compiles the routed example design
- 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_name>/<project_name>.srcs/sources_1/ip/<component_name>/
example_design/<component_name>_exdes.vhd
```

Verilog

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/
example_design/<component_name>_exdes.v
```

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

Demonstration Test Bench

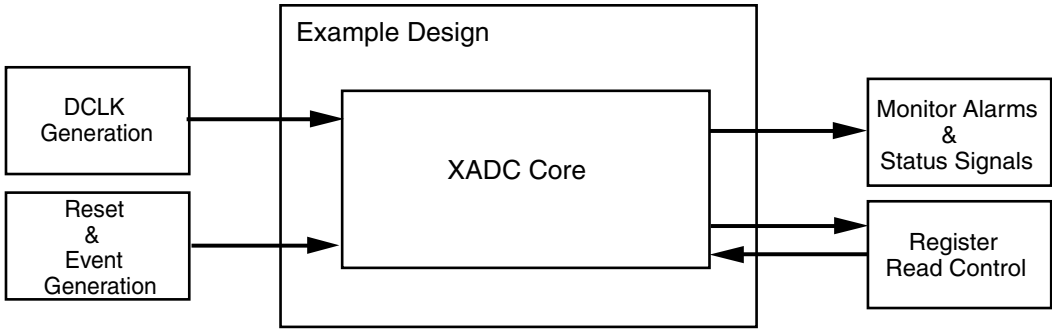

Figure 4-1: **Demonstration Test Bench for the XADC Wizard and Example Design**

The following files describe the demonstration test bench.

VHDL

```
<project_name>/<project_name>.srcs/sources_1/ip/<component_name>/
simulation/<component_name>_tb.vhd
```

Verilog

```
<project_name>/<project_name>.srcs/sources_1/ip/<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

Open Example Project Flow

In the Vivado tools, the command

```
open_example_project [get_ips <component_name>]
```

in the tcl console invokes a separate example design project where it creates `<component_name>_exdes` as the top module for synthesis and `<component_name>_tb` as the top module for simulation. Implementation or simulation of the example design can be run from the example project.

XILINX®

Appendix A

Additional Resources

Xilinx Resources

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

www.xilinx.com/company/terms.htm

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*

References

1. 7 Series FPGAs Overview (DS180)
2. 7 Series FPGAs XADC User Guide (UG480)
3. Vivado™ documentation: www.xilinx.com/support/documentation/dt_vivado.htm