1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

Zmod Scope AXI Configuration 1.0 IP Core User Guide

Revised April 15, 2022; Author Eduard Niță

1 Introduction

This user guide describes the Digilent **Zmod Scope AXI Configuration** Intellectual Property. It is used in conjunction with the **Zmod Scope Controller IP** to configure the **calibration coefficients, gain, coupling and other settings** and inspect the Zmod Scope's **status flags** using a set of control & status registers that can be accessed via an **AXI-Lite interface**.

IP quick facts	
Supported device families	Zynq®-7000, 7 series
Supported user interfaces	Xilinx®: AXI4-Lite
Provided with core	
Design files	C++ (HLS), VHDL
Simulation model	HLS Cosimulation for AXI4-Lite adapter, Behavioral Simulation for Handshake Synchronizer
Constraints file	XDC
Software driver	HLS Generated
Tested design flows	
Design entry	Vitis™ HLS 2021.1
Synthesis	Vivado Synthesis 2021.1

2 Features

- Allows setting the Zmod Scope's **calibration coefficients, High/Low gain, and AC/DC coupling** of each channel
- Allows monitoring of the Zmod Scope's **status flags**
- Xilinx interfaces used: **AXI4-Lite**
- Signals are **synchronized** between clock domains
- Compatible with **all Zmod Scope** variants

3 Overview

The main functional block of the IP is a **Register File**. The register values are synchronized from the AXI4-Lite clock domain into receiving clock domains using a **handshake synchronizer**. The Register File and the handshake synchronizer modules have been created using Vitis **High-Level Synthesis**.

3.1 Register File (ZmodScopeConfig.vhd)

3.1.1 Register map

Table 1. Registers

Offset	Register Name	Description
0x00	HLS Control signals	bit 0 - ap_start (Read/Write/COH) bit 1 - ap_done (Read/COR) bit 2 - ap_idle (Read) bit 3 - ap_ready (Read) bit 7 - auto_restart (Read/Write) others - reserved
0x04	Global Interrupt Enable Register	bit 0 - Global Interrupt Enable (Read/Write)
0x08	IP Interrupt Enable Register (Read/Write)	bit 0 - Channel 0 (ap_done) bit 1 - Channel 1 (ap_ready)
0x0C	IP Interrupt Status Register (Read/TOW)	bit 0 - Channel 0 (ap_done) bit 1 - Channel 1 (ap_ready)
0x10	Channel 1 High Gain Multiplicative Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch1HgMultCoef data (Read/Write)
0x18	Channel 1 Low Gain Multiplicative Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch1LgMultCoef data (Read/Write)
0x20	Channel 1 High Gain Additive Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch1HgAddCoef data (Read/Write)
0x28	Channel 1 Low Gain Additive Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch1LgAddCoef data (Read/Write)
0x30	Channel 2 High Gain Multiplicative Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch2HgMultCoef data (Read/Write)
0x38	Channel 2 Low Gain Multiplicative Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch2LgMultCoef data (Read/Write)
0x40	Channel 2 High Gain Additive Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch2HgAddCoef data (Read/Write)
0x48	Channel 2 Low Gain Additive Coefficient Register	bit 31~18 – Unused bit 17~0 – Ch2LgAddCoef data (Read/Write)

Offset	Register Name	Description
0x50	Configuration Register	bit 31~4 – Unused bit 0 – Channel 1 Gain (Read/Write) bit 1 – Channel 2 Gain (Read/Write) bit 2 – Channel 1 Coupling (Read/Write) bit 3 – Channel 2 Coupling (Read/Write) bit 4 – Test Mode (Read/Write) bit 5 – Enable Acquisition (Read/Write) Bit 6 – Zmod Scope Controller Negative Polarity Reset (Read/Write)
0x58	Status Register	bit 31~5 – Unused bit 0 – RstBusy (Read) bit 1 – InitDoneADC (Read) bit 2 – ConfigError (Read) bit 3 – InitDoneRelay (Read) bit 4 – DataOverflow (Read)

(SC = Self Clear, COR = Clear on Read, TOW = Toggle on Write, COH = Clear on Handshake)

3.1.2 Controlling the Register File

Bit 0 of the HLS Control Register(0x00), **ap_start**, kicks off the core from software. Writing 1 to this bit will **read any inputs grouped into the AXI4-Lite** slave interface. This means that after **updating a register** value via the AXI4-Lite interface, ap_start will need to be **asserted** in order to send the values towards the Zmod Scope Controller.

Bit 7 of the HLS Control Register(0x00), **auto_restart**, will set the IP core into **free running mode** when asserted. This means that **ap_start will get re-asserted** after sending the register values towards the Zmod Scope Controller.

Bit 1 of the HLS Control Register(0x00), **ap_done**, is asserted when the AXI4-Lite **registers values have been sent** towards the Zmod Scope Controller.

Before being sent to the Zmod Scope Controller, the register values are first **synchronized** between clock domains.

The **ap_done** bit is used to **trigger an interrupt** that signals the **handshake synchronizer** to send the register data from the **AXI4-Lite clock domain to a receiving clock domain**. Thus, bit 0 of the Global Interrupt Enable Register (0x04) and IP Interrupt Enable Register(0x08) must be asserted in order for the core to function properly.

More details on the **0x00-0x0C registers** can be found in Vitis High-Level Synthesis User Guide (UG1399)[1].

Details on the Zmod Scope **calibration coefficients, gain, coupling and status bits** can be found in Zmod Scope Controller IP Core User Guide[2].

3.2 Handshake Synchronizer (req_gen.vhd and ack_gen.vhd)

The **Handshake Synchronizer** is used to synchronize **multiple signals across two clock domains**. It is split into **two HLS modules**, the Handshake Request Generator, which sends a **request to synchronize data** from the sending clock domain to the receiving domain, and the Handshake Acknowledge Generator, which takes the request from the sending clock domains and returns an **acknowledge that the data is safe to be sent back** to the receiving clock domains.

The Handshake Synchronizer module has been split into two (one for each clock domain) due Vitis HLS being unable to create **designs with multiple clock domains**.

The synchronizer design used is described in *Chapter 5.6.3 (Closed-loop – Multi Cycle Path formulation with acknowledge feedback)* of *Clock Domain Crossing (CDC) Design & Verification Techniques using System Verilog* by *Clifford E. Cummings*[3].

The Handshake Request Generator and Handshake Acknowledge Generator modules **do not create the registers** used to store the data in their respective clock domains. These registers are **created externally** and then **driven by the load data signals** generated by the Handshake Synchronizer modules.

Data sent from the AXI4-Lite domain towards the Zmod Scope Controller domains (calibration coefficients and configuration bits) is synchronized when the Register File triggers an interrupt (detailed in section 3.1.2).

Data sent from the Zmod Scope Controller towards the Register File (status flags) is synchronized whenever the Handshake Request Generator is ready.

3.3 Clocking

The IP is divided in three clock domains:

1. The **System Clock Domain** (SysClk100), which is used in synchronization of the coupling, gain, test mode settings and enable acquisition signal from the AXI4-Lite domain towards the Zmod Scope Controller. It is also used in synchronization of the status flags of the Zmod Scope Controller towards the AXI4-Lite domain.
2. The **Sampling Clock Domain** (ADC_SamplingClk), which is used in synchronization of the external calibration coefficients from the AXI4-Lite domain towards the Zmod Scope Controller.
3. The **AXI4-Lite Clock Domain** (s_axi_control_clk), which is used to clock the Register File.

3.4 Reset

The IP has a single n**egative polarity reset** (s_axi_control_rst_n) **synchronous with the AXI4-Lite** interface. To assure that recovery/removal time of sequential logic is respected, the reset input is distributed to the different clock domains throughout the IP by ResetBridge modules. The ResetBridge

modules are responsible with converting the asynchronous reset input in reset signals with **synchronous de-assertion** (RSD) for each clock domain.

4 IP Top-Level Port Description

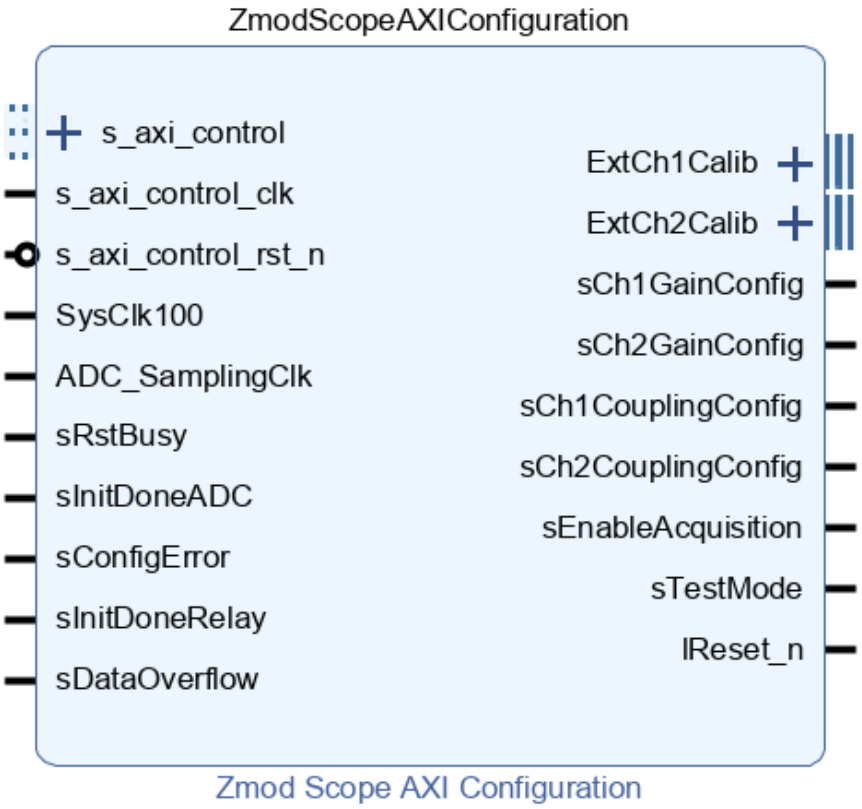

Table 2. IP port description

Signal Name	Interface	Signal Type	Init State	Description
SysClk100	-	I	N/A	100MHz input clock signal associated with the configuration and status signals
ADC_SamplingClk	-	I	N/A	Input clock associated with the calibration coefficients
sRstBusy	-	I	N/A	Zmod Scope Controller reset busy flag

Signal Name	Interface	Signal Type	Init State	Description
sInitDoneADC	-	I	N/A	Flag indicating when the Zmod Scope's ADC initialization is complete
sConfigError	-	I	N/A	This flag is asserted if the ADC initialization fails
sInitDoneRelay	-	I	N/A	Flag indicating when the Zmod Scope's relay initialization is complete
sDataOverflow	-	I	N/A	Flag indicating that the shallow synchronization FIFO in the Zmod Scope Controller is full.
cExtCh1LgMultCoef[17:0]	ExtCh1Calib	O	0	Channel1 low gain multiplicative coefficient
cExtCh1LgAddCoef[17:0]	ExtCh1Calib	O	0	Channel1 low gain additive coefficient
cExtCh1HgMultCoef[17:0]	ExtCh1Calib	O	0	Channel1 high gain multiplicative coefficient
cExtCh1HgAddCoef[17:0]	ExtCh1Calib	O	0	Channel1 high gain additive coefficient
cExtCh2LgMultCoef[17:0]	ExtCh2Calib	O	0	Channel2 low gain multiplicative coefficient
cExtCh2LgAddCoef[17:0]	ExtCh2Calib	O	0	Channel2 low gain additive coefficient
cExtCh2HgMultCoef[17:0]	ExtCh2Calib	O	0	Channel2 high gain multiplicative coefficient

Signal Name	Interface	Signal Type	Init State	Description
cExtCh2HgAddCoef[17:0]	ExtCh2Calib	O	0	Channel2 high gain additive coefficient
sCh1CouplingConfig	-	O	0	Channel1 AC DC coupling select • 1 = DC coupling. • 0 = AC coupling.
sCh2CouplingConfig	-	O	0	Channel2 AC DC coupling select • 1 = DC coupling. • 0 = AC coupling.
sCh1GainConfig	-	O	0	Channel1 gain select • 1 = High Gain. • 0 = Low Gain.
sCh2GainConfig	-	O	0	Channel2 gain select • 1 = High Gain. • 0 = Low Gain.
sTestMode	-	O	0	*sTestMode* is used to bypass the calibration block. When asserted, raw samples are provided on the output Data interface.
sEnableAcquisition	-	O	0	sEnableAcquisition when asserted enables data acquisition from the ADC.
lReset_n	--	O	0	Active low reset output synchronized in the AXI4-Lite clock domain, that can be used from software. Initially, the reset is asserted.
AXI4 Lite Interface Signals				
s_axi_control*		Input / Output		AXI4 Lite interface used to communicate with the control and status registers

5 Constraints

The IP uses the REG_ASYNC property to constrain the flip flops used to synchronize the request and acknowledge signals between clock domains.

set_false_path is used on signals that cross clock domains. These signals are marked with a CDC suffix.

6 Regenerating a HLS module

Opening one of the modules written in HLS is possible by executing the following command in the Vitis HLS Command Prompt:

```
cd <path_to_IP>/hls/<module_name>

vitis_hls -f run_hls_standalone.tcl
```

The **source files** of the project can be found in the **src** directory.

The **generated project** will be found inside the **ws** directory.

The top-level design uses the modules from the VHDL files generated by High Level Synthesis after synthesizing the C design.

7 References

The following documents provide additional information on the subjects discussed:

1. Xilinx Inc., “UG1399: Vitis High-Level Synthesis User Guide”, v2021.1, August 5
2. Digilent Inc., “Zmod Scope Controller IP Core User Guide”, v1.0, September 9, 2021
3. Clifford E. Cummings, “Clock Domain Crossing (CDC) Design & Verification Techniques using System Verilog”, SNUG Boston 2008