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

AXI4-Stream Decimator 1.0 IP Core User Guide

Revised February 8, 2022; Author Eduard Nițǎ

1 Introduction

This user guide describes the Digilent **AXI4-Stream Decimator** Intellectual Property. It takes an input streaming signal consisting of 32-bit samples over a slave AXI4-Stream interface, **decimates the signal by a factor** configured by the user and outputs on a master AXI4-Stream interface. It has an AXI4-Lite interface for control.

IP quick facts	
Supported device families	Zynq®-7000, 7 series
Supported user interfaces	Xilinx®: AXI4-Lite, AXI-Stream
Provided with core	
Design files	C++ VHDL/Verilog (generated)
Simulation model	CSim
Constraints file	XDC
Software driver	HLS Generated
Tested design flows	
Design entry	Vitis™ HLS 2021.1
Synthesis	Vivado Synthesis 2021.1

2 Features

- Configurable **packet length** (for DMA integration)
- Configurable **decimation factor**
- Xilinx interfaces used: AXI4-Lite, AXI-Stream

3 Designing with the core

The IP has been initially designed for a xc7z020clg400-1 target device with a target clock frequency of 125MHz (8.00 ns).

Decimation by a factor of N is done by **keeping only every Nth sample**.

The AXI4-Stream interfaces have their signals **registered**.

A **TLAST** signal is generated whenever the **number of samples sent** is equal to the **packet length**.

The **latency** of the IP is of 2 clock cycles.

3.1 Customization

Changes to the target device and target clock frequency can be done from the project GUI after the project was generated or by modifying the **SOLUTION_PART/SOLUTION_CLKP** variables found inside the *run_hls_standalone.tcl* file and then generating the project, according to the steps found in Generating the HLS Project.

4 Register map

Offset	Register Name	Description
0x00	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	Data signal of axilDecimationFactor	bit 31~0 - axilDecimationFactor [31:0] (Read/Write)
0x18	Data signal of axilPacketLength	bit 31~0 - axilPacketLength [31:0] (Read/Write)

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

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

5 Generating the HLS Project

Opening the IP in HLS is possible by executing the following command in the Vitis HLS Command Prompt:

```
cd <path_to_IP>/hls_proj

vitis_hls -f run_hls_standalone.tcl
```

Besides creating the project, the script will also **synthesize** the design and **export** the IP as an archive.

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

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

6 References

6.1 https://www.xilinx.com/support/documentation/sw_manuals/xilinx2021_1/ug1399-vitis-hls.pdf