clock_measure
Data Sheet

Content

Table of Contents

Figures

1 Introduction

This component implements a frequency measurement for one or more clocks. The AXI clock is used as reference.

1.1 Purpose

The document here describes this very generic firmware developed for all kinds of projects.

1.2 Scope

This document provides a detailed overview of the firmware interface and specifies the user interface.

1.3 Definitions, acronyms, and abbreviations

This document is based on the “IEEE Recommended Practice for Software Requirements Specifications” [1].

BSP	Board Support Package. Collection of software drivers adding functionality to easily accessing the components used in a System On Chip.
FPGA	Field Programmable Gate Array. Programmable logic device.

1.4 References

[1] IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications.

1.5 Overview

Chapter 2 provides an overview and how the firmware is related to the other firmware used.
Chapter 3 contains all the detail information on the interfaces.
Chapter 4 is meant for developers working on the core.

2 Overall description

The data_rec feature is loaded into a System On Chip (SOC) as a AXI slave component. The number of clocks to measure (*Clocks*) is configurable.

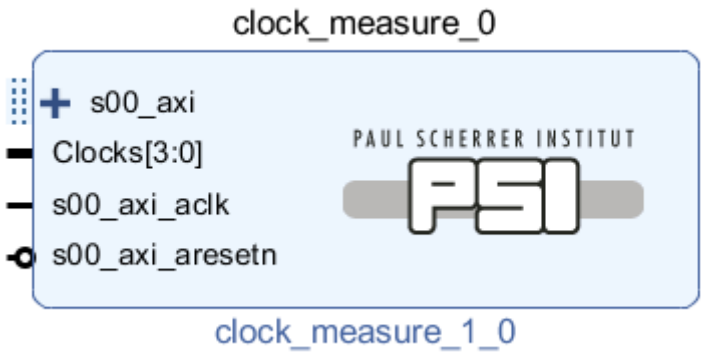

Figure 1: Component Overview

The number of clocks to measure as well as the frequency of the AXI clock (used as reference) and the maximum frequency that can be measured are configurable over the GUI.

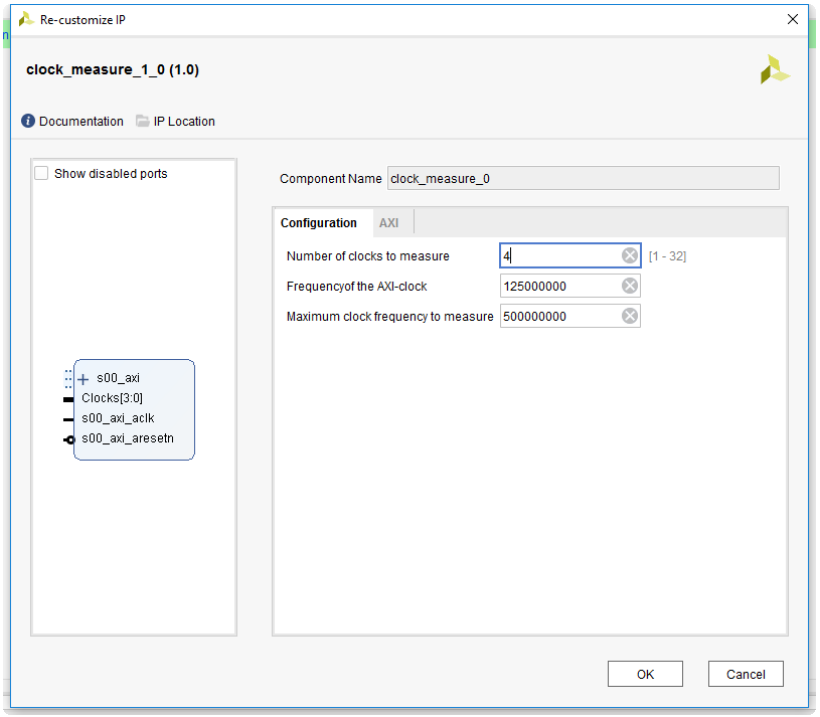

Figure 2: Component Configuration GUI

3 Request composition

3.1 Configuration

The following short notations for component parameters are used:

Address offset	R/W	Bit	Name	Description
0x00+4*n	R	31:0	FreqN	Frequency of the clock *n* in Hz

Table 1: Registers

4 Developer Information

4.1 Tools

To work on this core, the following tools are required:

- Vivado
- Modelsim PE

4.2 Simulation

A selfchecking testbench including a regression test script exists for this core. To run the regression test, follow the steps below:

1. Open ModelSim
2. Navigate to the “sim” directory
3. Execute “source ./run_tcl.tcl”

The regression script automatically compiles all VHDL files, runs all testbenches and checks if any errors occurred.

For interactive work during development, execute the steps below:

1. Execute a regression test as described above
2. For compiling all sources execute “psi::sim::compile –all”
3. Simulations can be ran either manually or by “psi::sim::run_tb –all”

4.3 Packaging

To simplify re-packaging of the IP-Core after changes and avoid trouble with the Vivado GUI, a packaging script was written. To re-package the IP-Core, follow the steps below:

1. Open Vivado
2. In the TCL console, navigate to the “scripts” directory (using “cd <path>”)
3. Execute “source ./package.tcl”