RGPIO

Table of Contents

IP Facts

Introduction

This Trenz Electronic teCORE IP provides a RGPIO (Remote GPIO) Interface to talk to external RGPIO Devices. Convert parallel data bus into 3 wire signal and back to parallel data bus.

IP can be used as Master communicate with external RGPIO devices with Slave interface or as Slave to communicate with external RGPIO devices with Master interface.

Features

- Master RGPIO
- Slave RGPIO
- GPIO Expander

teCORE™ IP Facts Table	
Supported Device Family	UltrasScale+ Zynq, 7 Series Zynq,UltraScale+ Series, UltraScale Series, 7 Series
Supported User Interfaces	Custom
Resources	
Special Features	RGPIO Master and Slave Controller
Provided with Core	
Design Files	VHDL Source Code
Constraint Files	Not provided, depends on module PCB
Example Design	Not Provided
Test Bench	Not Provided
Simulation Model	Not Provided
Supported S/W Dirver	Not Provided
Tested Design Flows	
Design Entry	Vivado® Design Suite, IP Integrator
Simulation	Vivado Simulator
Synthesis	Vivado Synthesis
Tested Hardware Platforms	
Support	
Provided by Trenz Electronic GmbH	

Overview

This Trenz Electronic teCORE IP provides a RGPIO (Remote GPIO) Interface to talk to external RGPIO Devices over 3 wire

IP can be used as Master communicate with external RGPIO devices with Slave interface or as Slave communicate with external RGPIO devices with Master interface

Feature Summary

- Master RGPIO
- Slave RGPIO

Licensing and Ordering Information

This Trenz Electronic teCORE is licensed under MIT License. This IP is included in various Reference Designs or contact Trenz Electronic Support (support[at]trenz-electronic.de) with subject line "[TE-IP-Core Request]" to order this IP-Core.

Product Specification

Performance

Maximum RGPIO output CLK depends on Master and Slave device implementation. In the most cases maximum frequency of 25MHz is allowed.

Resource Utilization

Port Description

RGPIO_EXT External Interface

Communication channel between master and slave interface.

Port Name	IO	Description
RGPIO_M_CLK	out	RGPIO Master Clock
RGPIO_M_RX	in	RGPIO Master RXD
RGPIO_M_TX	out	RGPIO Master TXD

Port Name	IO	Description
RGPIO_S_CLK	out	RGPIO Slave Clock
RGPIO_S_RX	in	RGPIO Slave RXD
RGPIO_S_TX	out	RGPIO Slave TXD

RGPIO_M_USR Interface

Master user interface to communicate with slave device.

Port Name	IO	Description
RGPIO_M_OUT	out	23bit data output to slave device*
RGPIO_M_IN	in	23bit data input from slave device*
RGPIO_M_RESERVED_OUT	out	4bit reserved for future usage
RGPIO_M_RESERVED_IN	in	4bit reserved for future usage
RGPIO_M_SLAVE_ACTIVATION_CODE	out	4bit activation code from external slave for information only

RGPIO_M_ENABLE	in	Enable RGPIO communication. High active. Set RGPIO data as valid for Slave. Data will always transmitted, if CLK is available.
RGPIO_M_USRCLK	in	RGPIO transmission CLK for master and slave
RGPIO_M_RESET_N	in	RGPIO Reset. Low active.

*currently limited to 23 bit to use IP with CPLD implementations of TE Boards. For general usage, this restriction will be removed on future IP update.

RGPIO_S_USR Interface

Slave user interface to communicate with master device.

Port Name	IO	Description
RGPIO_S_OUT	out	23bit data output to master device*
RGPIO_S_IN	in	23bit data input from master device*
RGPIO_S_RESERVED_OUT	out	4bit reserved for future usage
RGPIO_S_RESERVED_IN	in	4bit reserved for future usage
RGPIO_S_MASTER_ACTIVATION_CODE	out	4bit activation code from external master for information only
RGPIO_S_ENABLED	out	Interface status. Indicates RGPIO data are valid.

*currently limited to 23 bit to use IP with CPLD implementations of TE Boards. For general usage, this restriction will be removed on future IP update.

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

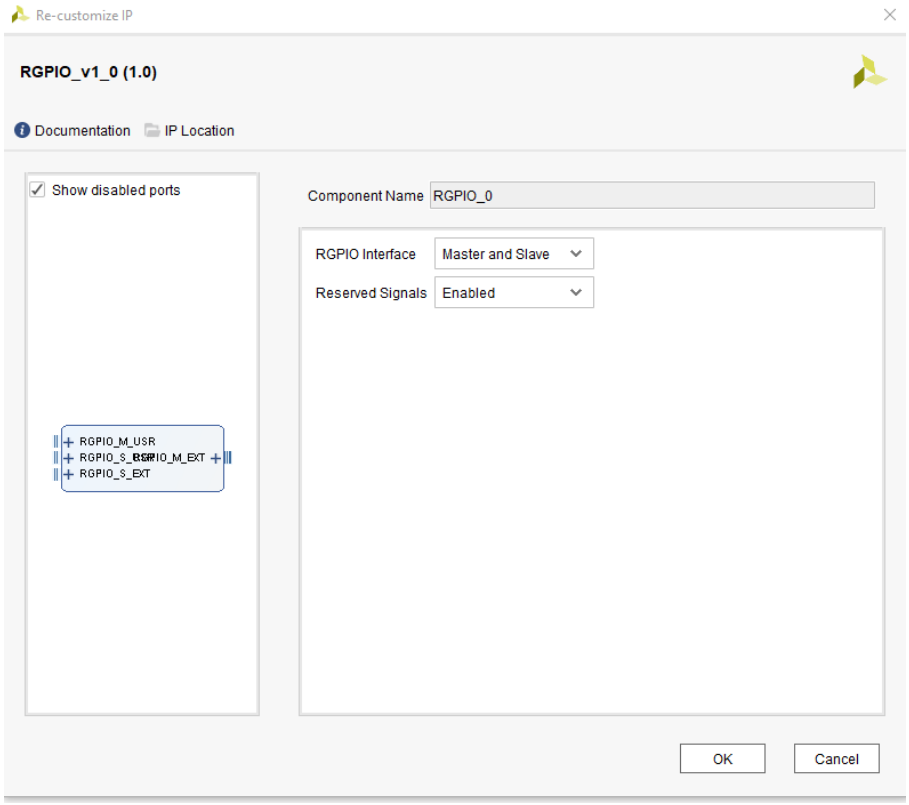
Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the Vivado IP integrator can be found in the following Vivado Design Suite user guides:

- Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator (UG994)
- Vivado Design Suite User Guide: Designing with IP (UG896)
- Vivado Design Suite User Guide: Getting Started (UG910)

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado Design Suite.



- RGPIO Interface: Master, Slave or both are possible
- Reserved Signals: Active/Deactivate reserved signals as interface

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

Loc constrains and IO Standard depends on module and usage.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

Maximum RGPIO output CLK depends on Master and Slave device implementation. In the most cases maximum frequency of 25MHz is always allowed.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

This core does not support simulation.

Synthesis and Implementation

This section contains information about synthesis and implementation in the Vivado Design Suite. For details about synthesis and implementation, see the Vivado Design Suite User Guide:

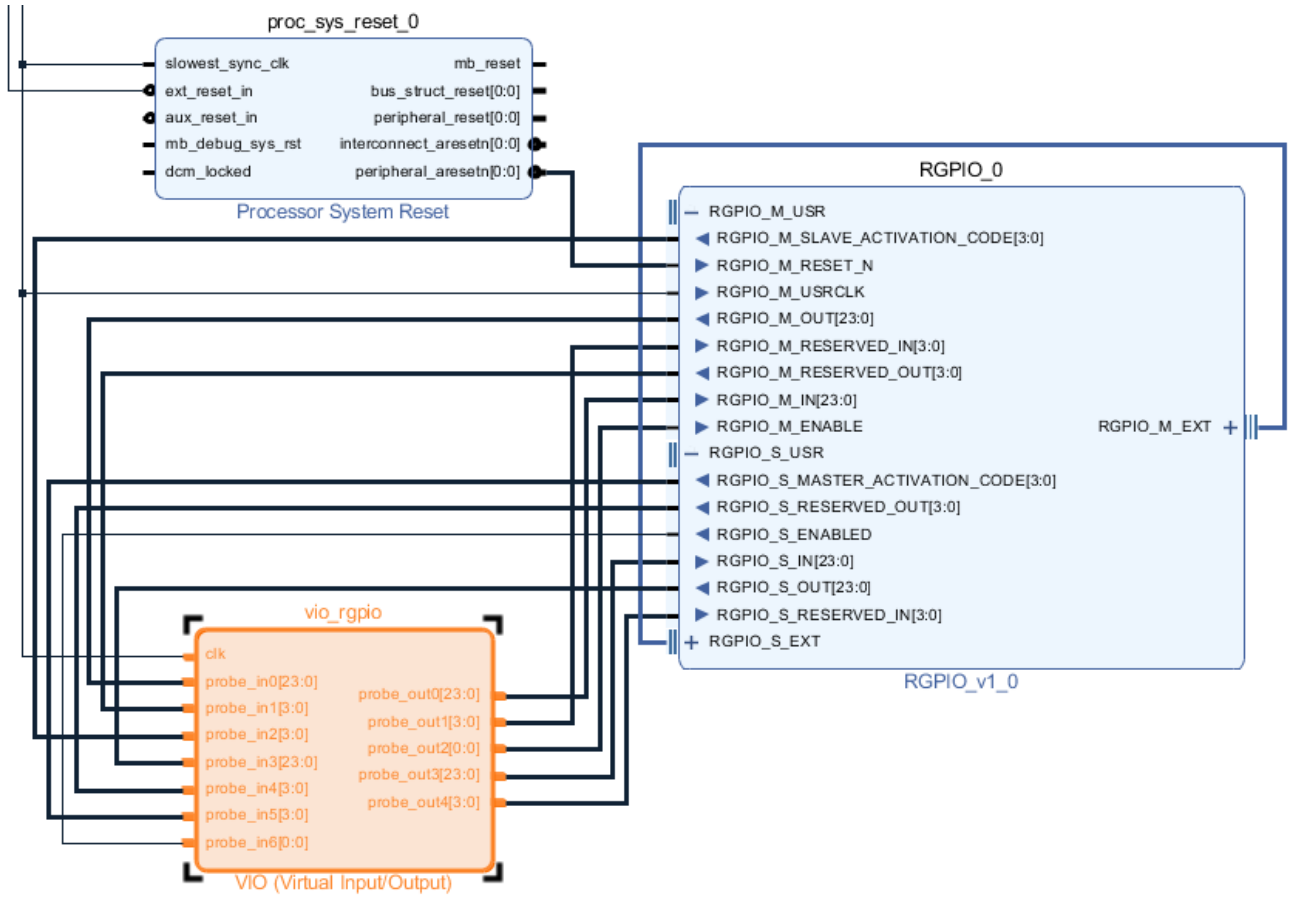
- Designing with IP (UG896)

Example Design

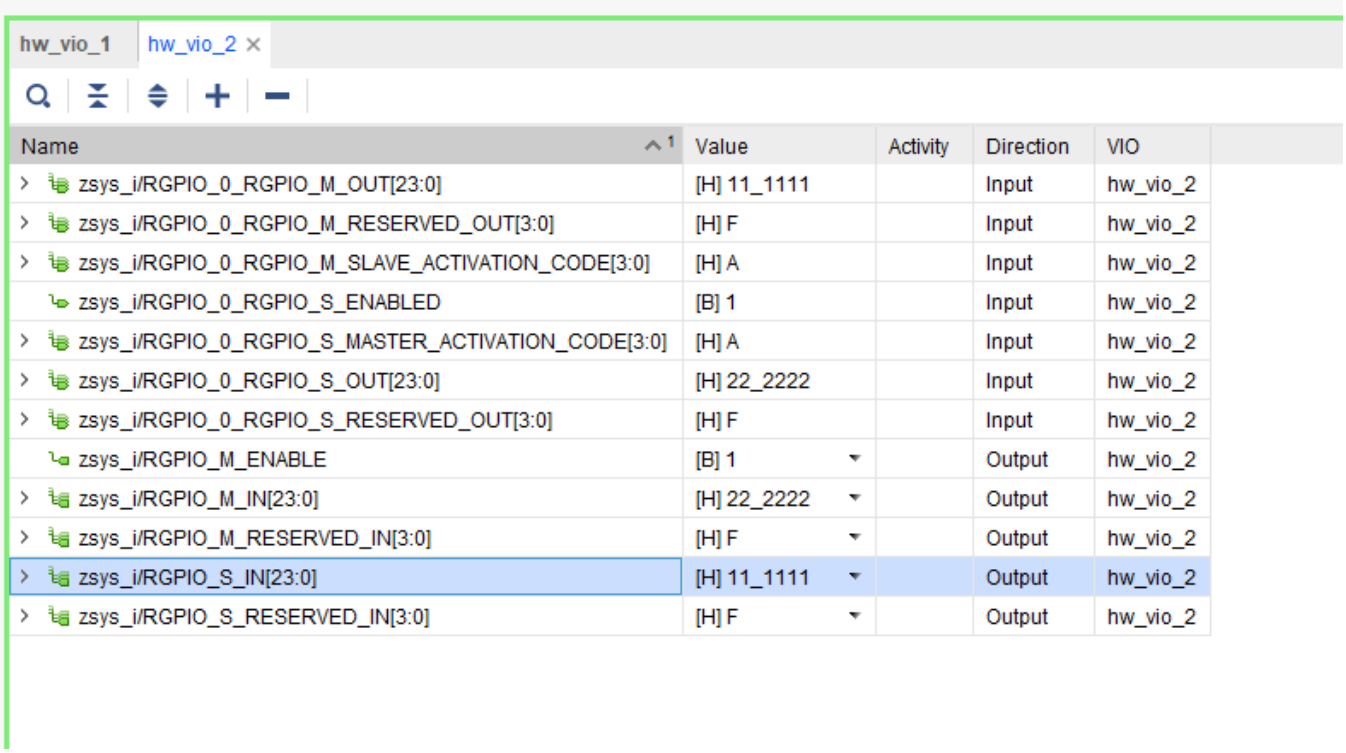
There is no example Design for this IP core release.

Use Master Slave loopback over RGPIO_EXT interface to test IP Master and Slave Interface together.

Vivado Block Design:



VIO HW Manager:



Test Bench

There is no test bench for this IP core release.

Appx. A: Change History and Legal Notices

Document Change History

Date	Document Revision	IP Revision	Authors	Description

Error rendering macro 'page-info' Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] Unknown macro: 'metadata'	v1.0	**Error rendering macro 'page-info'** Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]	• initial release with Vivado 2017.4

Legal Notices

Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

Copyright Notice

Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

Error rendering macro 'page-info'

Ambiguous method overloading for method jdk.proxy279.$Proxy4022#hasContentLevelPermission. Cannot resolve which method to invoke for [null, class java.lang.String, class com.atlassian.confluence.pages.Page] due to overlapping prototypes between: [interface com.atlassian.confluence.user.ConfluenceUser, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject] [interface com.atlassian.user.User, class java.lang.String, class com.atlassian.confluence.core.ContentEntityObject]