Transceiver FPGA Reference Design Guide

797 North Grove Rd, Suite 101
Richardson, TX 75081
Phone: (972) 671-9570
www.redrapids.com

Red Rapids reserves the right to alter product specifications or discontinue any product without notice. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment. This product is not designed, authorized, or warranted for use in a life-support system or other critical application.

All trademark and registered trademarks are the property of their respective owners.

Table of Contents

List of Figures

1.0 Introduction

Products equipped with a customer programmable FPGA include a VHDL reference design that demonstrates key functions of the product. This manual is targeted to the FPGA developer that will use the reference design as the template for a customer specific application. A separate manual describes each of the VHDL models used by the reference design.

The latest product documentation and software is available for download from the Red Rapids website (www.redrapids.com).

1.1 Conventions

This manual uses the following conventions:

- Hexadecimal numbers are prefixed by "0x" (e.g. 0x00058C).
- *Italic* font is used for names of registers.
- Blue font is used for names of directories, files and OS commands.
- Green font is used to designate source code.
- Active low signals are followed by '#', For example, TRST#.

☞	Text in this format highlights useful or important information.

!	Text shown in this format is a warning. It describes a situation that could potentially damage your equipment. Please read each warning carefully.

The following are some of the acronyms used in this manual.

- **ADC** Analog to Digital Converter
- **DAC** Digital to Analog Converter
- **DCM** Digital Clock Manager
- **DMA** Direct Memory Access
- **IDELAY** Virtex-5 Input Delay Element
- **IDELAYCTRL** Virtex-5 Input Delay Control Element
- **IOB** Virtex-5 Input/Output Block
- **MSPS** Mega Samples per Second
- **PCIe** Peripheral Component Interconnect Express
- **QDR** Quad Data Rate

1.2 Reference Design FPGA Code Distribution

Source code to the FPGA reference design is distributed in an archive file that can be downloaded from the Red Rapids website. The archive contains all of the design files necessary to create a Vivado project and generate a bitstream.

The archive will create a \RedRapids root directory and the following subdirectories:

\RedRapids\vhdl_models

The vhdl_models subdirectory contains VHDL source code that is not associated with any specific product. These models perform common functions that are used in multiple reference designs distributed by Red Rapids.

\RedRapids\<product name>

The <product name> subdirectory is unique to each specific product family. This directory contains subdirectories for source code and project files associated with the FPGA reference design.

\RedRapids\<product name>\constraints

The constraints subdirectory contains files that assign signal names to FPGA pins and timing objectives to clock signals.

\RedRapids\<product name>\ip_sources

The ip_sources subdirectory contains intellectual property cores supplied by Xilinx and third parties.

\RedRapids\<product name>\ip_sources \plda

PCI Express DMA core supplied by PLDA and licensed through Red Rapids.

\RedRapids\<product name>\ip_sources \vivado

Cores produced by the Xilinx Vivado design tool for use in a Vivado project.

\RedRapids\<product name>\project

The project subdirectory contains a Tcl file that will build a Vivado project specific to a unique Red Rapids part number. Consult the ReadMe file within this folder for instructions on executing the Tcl script.

\RedRapids\<product name>\top_source

The top_sources subdirectory contains source code and constraint files specific to each product model number.

\RedRapids\<product name>\top_source\ Mxxx

The Mxxx subdirectory notation is an abbreviation for a specific model number. For example, the M377 folder contains files unique to the Model 377.

\RedRapids\<product name>\top_source\ Mxxx\NO_SRAM

The NO_SRAM subdirectory contains source code specific to build options without the QDR II+ SRAM.

\RedRapids\<product name>\top_source\ Mxxx\FF_SRAM

The FF_SRAM subdirectory contains source code specific to build options that include the QDR II+ SRAM directly clocked by a fixed frequency oscillator.

\RedRapids\<product name>\top_source\ Mxxx\SS_SRAM

The SS_SRAM subdirectory contains source code specific to build options that include the QDR II+ SRAM clocked by a spread spectrum MMCM.

1.3 Reference Design Binary Code Distribution

Binary files containing the FPGA reference design are distributed in an archive file that can be downloaded from the Red Rapids website.

The archive contains at least one subdirectory named \Model_XXX-YYY, where XXX-YYY matches the specific part number of the product you ordered from Red Rapids (e.g. Model 377-403).

Each product directory contains a bitstream (.bit) file that can be used to program the FPGA and a hex (.mcs) file to program the configuration flash. These files will restore the reference design that is loaded in flash when the product ships.

1.4 Revision History

Version	Date	Description
R01	8/4/2016	Updated code distribution description.
R00	3/9/2014	Initial release.

2.0 Transceiver Architecture

The Transceiver VHDL model, shown in Figure 2-1, can be scaled to any number of channels of arbitrary bus width. This model forms the basis of the Transceiver reference design that is used to demonstrate the functionality of several Red Rapids products. The Transceiver model is actually a hierarchical grouping of interconnected lower level models. Further details about all of the models shown in Figure 2-1 can be found in the VHDL Model Reference Manual.

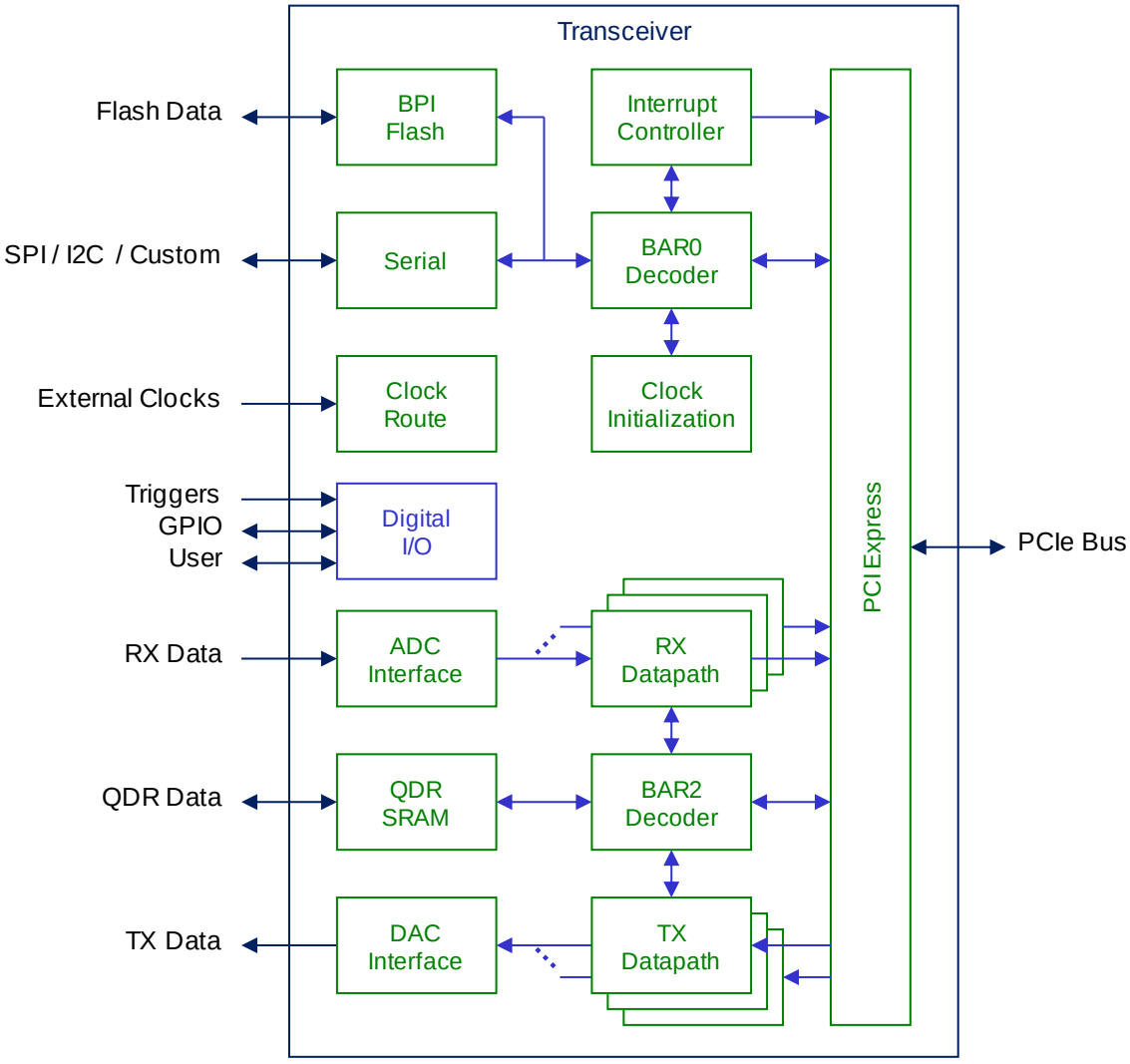

Figure 2-1 Transceiver Model Functional Block Diagram

> ☞ Keep your first FPGA design simple so that you can focus on the operation of the transceiver without the overhead of complex application logic to consider.

The Transceiver reference design consists of the Transceiver model surrounded by I/O buffers as shown in Figure 2-2. A unique top level VHDL model, package and constraints file are supplied for each product in a family. These files define the design specific characteristics of that product. Variables that are defined by constants in the package include the number of channels, bus widths, buffer types, etc. Even products that have only receiver channels or

transmitter channels use this design. The missing channels are never synthesized, so there are no wasted resources.

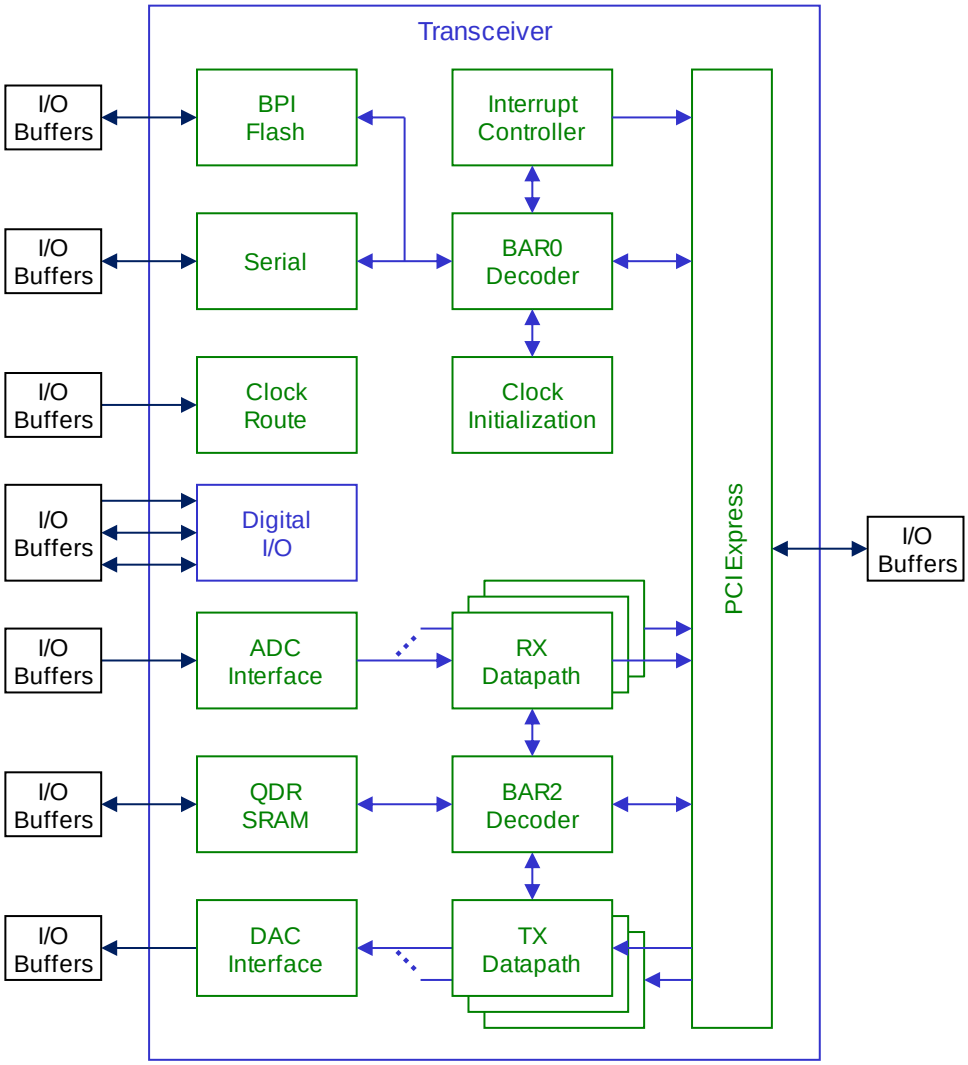

Figure 2-2 Transceiver Reference Design Block Diagram

A default Transceiver configuration assigns one receiver channel to each physical analog input port and one transmitter channel to each physical analog output port. Each channel includes a dedicated DMA controller, so the number of DMA data channels will always match the total number of analog ports available on a product.

Channels are sequentially numbered in the reference design starting with one for the first receiver channel. Transmitter channel numbering follows the last receiver channel if there are any present in the product. For instance, the first transmitter channel will be assigned the number one if there are no receiver channels available. However, the first transmitter channel will be assigned the number three if there are two receiver channels available.

2.1 Top Level VHDL Model

The top level VHDL model defines all of the I/O buffers associated with a specific product and calls the scalable Transceiver model that is instantiated by all products. All of the ports assigned to the top level entity correspond to physical pins on the FPGA device.

The port names are referenced in the constraint files to assign physical location to each of these pins on the package.

!	The port names of the top level entity should never be changed. The buffer assignments in the top level architecture should not be changed unless the GPIO or User bus needs to be configured for differential signaling.

2.2 Constraint Files

Constraint files are used to assign physical placement information and timing goals for the design. A different file is supplied for each major interface to keep the information organized and isolate design specific features that may not apply to all products. The majority of constraints apply to physical placement or timing goals for the design.

Physical placement constraints are used to assign ports in the top level entity to specific pins on the FPGA package. Placement constraints are also used to lock down the location of timing critical elements of a design.