RTI FPGA Programming Blockset

FPGA Interface Reference

For RTI FPGA Programming Blockset 3.11

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

Contents

About This Reference

Content

This RTI Reference is a description of the Real-Time Interface (RTI) blocks and their settings provided by the **FPGA Interface** sublibrary of the RTI FPGA Programming Blockset. The common block settings and the framework-specific block settings are described in separate sections.

You can use this blockset to integrate an FPGA model in your dSPACE system. For the supported dSPACE hardware, refer to Overview of the RTI FPGA Programming Blockset on page 15.

Audience profile

It is assumed that you have good knowledge in:

- Applying generally accepted FPGA design rules to ensure a stable and reliable FPGA application.
- The architectural structure of FPGAs (CLB architecture, slice flip-flops, memory resources, DSP resources, clocking resources) with a verifiable experience on digital designs (structural mapping, tool-flow knowledge, synthesis options, timing analysis).
- Modeling with Simulink®.
- Modeling with the Xilinx® System Generator Blockset.
- Using the Xilinx® design tools for simulation and debugging.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
⚠ DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.

Symbol	Description
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
⍰	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
📖	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Examples:

- Where you find terms such as `rti<XXXX>` replace them by the RTI platform support you are using, for example, `rti1007`.
- Where you find terms such as `<model>` or `<submodel>` in this document, replace them by the actual name of your model or submodel. For example, if the name of your Simulink model is `smd_1007_sl.slx` and you are asked to edit the `<model>_usr.c` file, you actually have to edit the `smd_1007_sl_usr.c` file.

RTI block name conventions All I/O blocks have default names based on dSPACE's board naming conventions:

- Most RTI block names start with the board name.
- A short description of functionality is added.
- Most RTI block names also have a suffix.

Suffix	Meaning
B	Board number (for PHS-bus-based systems)
M	Module number (for MicroAutoBox II)
C	Channel number
G	Group number
CON	Converter number
BL	Block number
P	Port number
I	Interrupt number

A suffix is followed by the appropriate number. For example, DS2201IN_B2_C14 represents a digital input block located on a DS2201 board. The suffix indicates board number 2 and channel number 14 of the block. For more general block naming, the numbers are replaced by variables (for example, DS2201IN_Bx_Cy).

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the 📄 icon in dSPACE Help. The PDF opens on the first page.

General Information on the FPGA Interface of the RTI FPGA Programming Blockset

Where to go from here

Information in this section

Overview of the RTI FPGA Programming Blockset

Introduction

To provide a short description of the blockset's components and how to access them.

RTI FPGA Programming Blockset

The RTI FPGA Programming Blockset is a Simulink® blockset for using an FPGA model with a dSPACE system.

The blockset provides RTI blocks for implementing and simulating the interface between the FPGA mounted on a dSPACE I/O board and the board's I/O, and the interface between the dSPACE I/O board and its processor board. The following table shows the supported FPGA hardware.

Platform	Supported Hardware	Framework	Notes
MicroLabBox	MicroLabBox	*DS1202 FPGA I/O Type 1*	Remaining I/O channels cannot be used by RTI/RTLib.
		DS1202 FPGA I/O Type 1 (Flexible I/O)	Can be used together with the RTI1202 and the RTI Electric Motor (EMC) blocksets.
MicroAutoBox II	▪ MicroAutoBox II 1401/1511/1514 ▪ MicroAutoBox II 1401/1513/1514	*FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*	Supports the DS1552 Multi-I/O Module.
		FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)	Supports the DS1552B1 Multi-I/O Module.
		FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)	Supports the DS1554 Engine Control I/O Module.
MicroAutoBox III	DS1514 FPGA Base Board	FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)	Supports the DS1552 Multi-I/O Module.
		FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)	Supports the DS1552B1 Multi-I/O Module.
		FPGA1403Tp1 (7K325) with Engine Control Module (DS1554)	Supports the DS1554 Engine Control I/O Module.
SCALEXIO	DS2655 FPGA Base Board (7K160)	*DS2655 (7K160) FPGA Base Board*	The SCALEXIO FPGA base boards provide 5 slots to extend the I/O capability with DS2655M1 Multi-I/O Modules, DS2655M2 Digital I/O Modules, and DS6551 Multi-I/O Modules. The assembly has no slot dependencies. With the *Inter-FPGA Interface* framework, you can use I/O module slots of the SCALEXIO FPGA base boards as inter-FPGA interfaces.
	DS2655 FPGA Base Board (7K410)	*DS2655 (7K410) FPGA Base Board*	
	DS6601 FPGA Base Board	*DS6601 (KU035) FPGA Base Board*	
	DS6602 FPGA Base Board	*DS6602 (KU15P) FPGA Base Board*	
	DS2655M1 Multi-I/O Module	*DS2655M1 I/O Module*	Can be used after you load one of the SCALEXIO FPGA base board frameworks.
	DS2655M2 Digital I/O Module	*DS2655M2 I/O Module*	
	DS6651 Multi-I/O Module	*DS6651 Multi-I/O Module*	
	Inter-FPGA connection	*Inter-FPGA Interface*	
	MGT communication bus	*DS660X_MGT*	An MGT module can be plugged into the DS6601 and DS6602 FPGA base boards.

Platform	Supported Hardware	Framework	Notes
PHS-bus-based system	DS5203 FPGA Board (7K325)	*DS5203 (7K325) with onboard I/O*	Supports the FPGA board, but no I/O modules.
		DS5203 (7K325) with Multi-I/O Module (DS5203M1)	Supports the FPGA board and the DS5203M1 Multi-I/O Module.
	DS5203 FPGA Board (7K410) without I/O Module	*DS5203 (7K410) with onboard I/O*	Supports the FPGA board, but no I/O modules.
		DS5203 (7K410) with Multi-I/O Module (DS5203M1)	Supports the FPGA board and the DS5203M1 Multi-I/O Module.

Library access

To open the library, execute one of the following methods:

- In the MATLAB Command Window, enter `rtifpga`.
- To access the RTI blocks of the library separately:

  In the Simulink Library Browser, navigate to the **dSPACE RTI FPGA Programming Blockset** folder.
- For MicroAutoBox II:

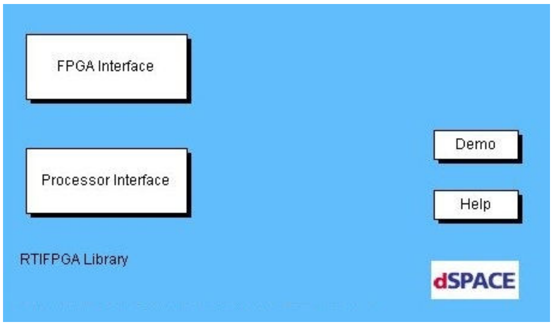
  In the RTI1401 Blockset, click one of the following blocksets:
  - **MicroAutoBox II DS1511/DS1514**
  - **MicroAutoBox II DS1513/DS1514**

  Then click **FPGA Type 1**.
- For MicroLabBox:

  In the **DS1202 MicroLabBox FPGA I/O Type 1** blockset, Click **FPGA Class 1**.
- For PHS-bus-based systems:

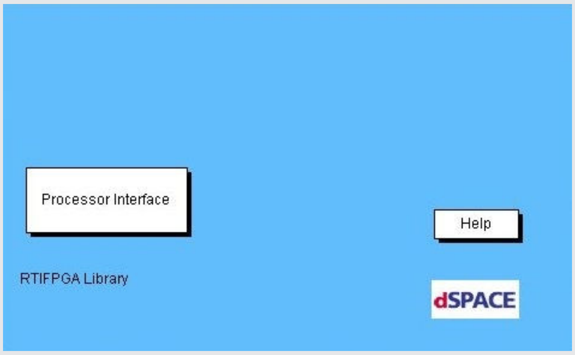
  In the DS1006 or DS1007 Blockset, click **Blocksets - RTI FPGA Pr. Blockset**.

If you open the block library, the blockset is displayed.



Note

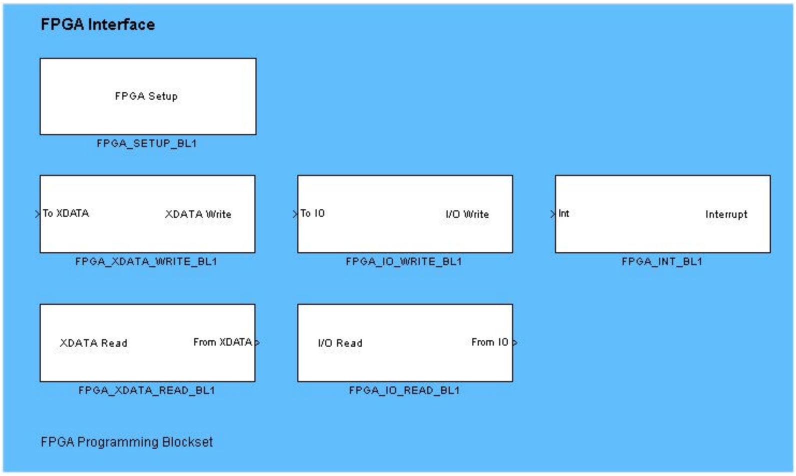
The **FPGA Interface** library and the **Demo** model are only available, if you have the full license for the RTI FPGA Programming Blockset.
If you have the standard RTI license only, the blockset looks like this:



Library components

The following components are available in the RTI FPGA Programming Blockset:

FPGA Interface The RTI blocks of the FPGA INTERFACE sublibrary are used on the dSPACE I/O board that provides an FPGA, for example, the DS5203 FPGA Board. They let you configure the interface to external I/O and to a processor board, for example, a DS1007.



The **FPGA INTERFACE** sublibrary provides the following RTI blocks:

- Setup
  - FPGA_SETUP_BL on page 27
- Processor board access
  - FPGA_XDATA_READ_BL on page 40
  - FPGA_XDATA_WRITE_BL on page 45
  - FPGA_INT_BL on page 62
- External I/O access
  - FPGA_IO_READ_BL on page 50
  - FPGA_IO_WRITE_BL on page 56

Processor Interface The RTI blocks of the **Processor Interface** library are used on the dSPACE processor board to implement the communication with the I/O board. The model on the I/O board that you want to access, for example, the DS5203 FPGA Board, must contain the related FPGA interface RTI blocks.

For further information, refer to RTI FPGA Programming Blockset - Processor Interface Reference .

> **Note**
>
> If you use a SCALEXIO system or MicroAutoBox III, the processor interface is implemented within the behavior model using blocks of the **Model Port Block** library.

Demo model

If you have the full license for the RTI FPGA Programming Blockset, Simulink models are available that show how to use the RTI blocks of the RTI FPGA Programming Blockset. Double-click the **Demos** button in the blockset to open the library containing the demo models. In the next step you have to choose the demo model for the framework which you have in use. A demo model prepared for a different framework will not work. You can also find the model files at `<RCP_HIL_InstallationPath>\Demos\RTIFPGA`.

Related topics

References

Features of the RTI FPGA Programming Blockset.. 19

Features of the RTI FPGA Programming Blockset

Introduction

The RTI FPGA Programming Blockset allows you to program an FPGA within a dSPACE system.

Main features

Specific dSPACE hardware provides a Xilinx® FPGA for which you can implement an application. The RTI FPGA Programming Blockset allows you to integrate an FPGA model created with the Xilinx® System Generator Blockset in a Simulink model that can be built to run on dSPACE hardware.

With the full license for the RTI FPGA Programming Blockset, you have access to the **Processor Interface** sublibrary and the **FPGA Interface** sublibrary.

The main features of the blockset are:

- I/O access

  You can connect the FPGA model with external I/O using the **FPGA_IO_READ_BL** and **FPGA_IO_WRITE_BL** blocks. The basic functionality for accessing analog and digital input and output signals is provided by the FPGA framework coming with the FPGA board.

  For example, if your PHS-bus-based system contains a DS5203 FPGA Board (7K325), you can select the *DS5203 (7K325) with onboard I/O* framework in the block dialog of the **FPGA_SETUP_BL** block.

  For an overview of the FPGA board's I/O capability, refer to Hardware Supported by the RTI FPGA Programming Blockset (RTI FPGA Programming Blockset Guide 🕮).

- Communication between processor board and I/O board

  You can connect the FPGA model with the processor model running on the computation node. Data exchange between the I/O board and the processor board runs via the board-specific bus.

  - DS1006 or DS1007 processor board: **PHS bus**
  - MicroAutoBox II 1401/1511/1514 and 1401/1513/1514: **Intermodule bus**
  - MicroAutoBox III with DS1514 FPGA Base Board: **Intermodule bus**
  - MicroLabBox: **Local bus**
  - SCALEXIO system: **IOCNET bus**

  If you are using a PHS-bus-based system, MicroAutoBox II, or MicroLabBox, you must have an **FPGA_XDATA_READ_BL** or **FPGA_XDATA_WRITE_BL** block in the FPGA model and a corresponding **PROC_XDATA_WRITE_BL** or **PROC_XDATA_READ_BL** block in the processor model to implement a communication line.

  Note

  If you use a SCALEXIO system or a MicroAutoBox III, the **Model Port Block** library is used instead of the **Processor Interface** sublibrary.

  For an overview of the processor communication capability, refer to Exchanging Data With the Processor Model (RTI FPGA Programming Blockset Guide 🕮)

- Asynchronous tasks

  With the interrupt blocks from the FPGA interface (**FPGA_INT_BL**) and the processor interface (**PROC_INT_BL**), you can implement interrupt-driven tasks in the processor model triggered from the FPGA model.

Note

If you use a SCALEXIO system, the **Runnable Function** block from the **Model Port Block** library is used instead of the **PROC_INT_BL** block.

Managing FPGA and processor application

The setup blocks of the RTI FPGA Programming Blockset (**FPGA_SETUP_BL** and **PROC_SETUP_BL**) provide commands to manage the FPGA application and the processor application:

- Starting the build process for the FPGA model.
- Executing a timing analysis for the FPGA model using a feature of the Xilinx System Generator.
- Executing an HDL simulation for the FPGA model using a feature of the Xilinx System Generator.
- Programming the generated FPGA code to the flash of the FPGA board or the RAM of the FPGA.
- Starting the build process for the processor application.
- Creating a separate burn application to explicitly program the FPGA.
- Integrating the FPGA application into the processor application to automatically program the FPGA at startup.

Note

If you use a SCALXIO system, you have to use ConfigurationDesk for handling the processor application.

Dynamic block settings

Unlike the settings in other RTI blocksets, numerous dialog settings are generated dynamically according to some basic settings done in the generic part of the block dialogs. In the initial state of the FPGA_XDATA_xxx, FPGA_IO_xxx and FPGA_INT blocks, the Parameters and Description pages are empty. The settings are defined separately in the framework INI file, which is loaded the first time you request information from it.

INI files used with the RTI FPGA Programming Blockset

There are two kinds of initialization files:

- Framework INI file

  A framework INI file contains the interface definitions for the FPGA, the FPGA board's I/O and the processor. It also contains the function-specific settings that are displayed in the dialogs of the FPGA Interface RTI blocks according to the specified function. It is therefore mainly used for configuring the FPGA Interface RTI blocks.
- FPGA model INI file

  An FPGA model INI file is created when you build an FPGA application. It allows you to include built FPGA applications in your processor application without specifying the corresponding FPGA model. For example, in the setup block for the processor model, you can specify either an FPGA subsystem or an FPGA model INI file for further actions.

Basic information and instructions

For further information on the RTI FPGA Programming Blockset, refer to RTI FPGA Programming Blockset Guide 📖.

Details on the access types

The FPGA framework contains the definition of the data storage areas. It specifies one data storage type as register (implemented as Flip-Flop) and one data storage type as buffer (implemented in the FPGA RAM). With the access type, you can choose the data storage that you want to use for the data exchange.

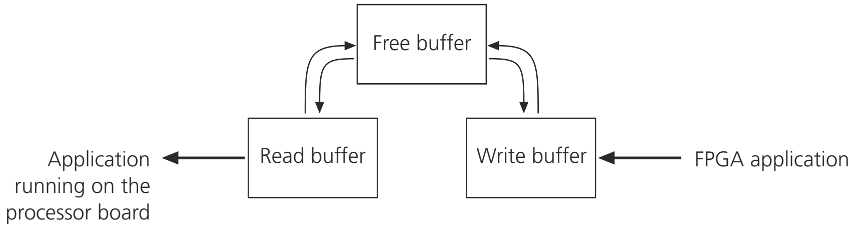
Register access Register access lets you access a scalar value in the register. The data is identified by the specified channel number. The values are transmitted element by element.

Register group access You can group registers to a register group via a common **Register Group ID**. All the values that belong to the same **Register Group ID** are synchronously updated in the FPGA subsystem.

For read access, the registers of a register group are read from the board-specific bus sequentially and then provided to the FPGA application simultaneously. For write access, the registers of a register group are sampled simultaneously in the FPGA application. These values form a consistent data group that is written to the board-specific bus.

Buffer access Buffer access lets you access a vector value in the data buffer. One specific value of the data is identified by the specified channel number and the position within the buffer.

Data exchange is implemented via a FIFO buffer that works as a swinging buffer. This means that there are two separate buffers for reading and writing, and one buffer that switches between reading and writing. Only the pointer has to be changed to switch the buffer so that no buffer has to be copied from one position to another.



Related topics

Basics

Exchanging Data With the Processor Model (RTI FPGA Programming Blockset Guide 🕮)
Hardware Supported by the RTI FPGA Programming Blockset (RTI FPGA Programming Blockset Guide 🕮)

References

Overview of the RTI FPGA Programming Blockset.. 15

FPGA Interface RTI Blocks

Introduction

The FPGA Interface library provides RTI blocks that you use in the FPGA model to implement access to the processor model and to external I/O.

Where to go from here

Information in this section

Common Settings of the FPGA Interface RTI Blocks....................................27
The common settings of a block dialog are available independently of a hardware-specific framework.

RTI Block Settings for the DS1202 FPGA I/O Type 1....................................67
The block dialogs provide hardware-specific settings after you load one of the DS1202 FPGA I/O Type 1 frameworks.

RTI Block Settings for the DS2655 FPGA Base Board Framework............121
The block dialogs provide hardware-specific settings after you load one of the DS2655 FPGA Base Board frameworks.

RTI Block Settings for the DS6601 FPGA Base Board Framework............169
The block dialogs provide hardware-specific settings after you load the DS6601 (KU035) FPGA Base Board framework.

RTI Block Settings for the DS6602 FPGA Base Board Framework............219
The block dialogs provide hardware-specific settings after you load the DS6602 (KU15P) FPGA Base Board framework.

RTI Block Settings for the DS2655M1 I/O Module Framework................279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

RTI Block Settings for the DS2655M2 I/O Module Framework................307
The block dialogs provide hardware-specific settings after you load the DS2655M2 I/O Module framework.

RTI Block Settings for the DS6651 Multi-I/O Module Framework............345
The block dialogs provide hardware-specific settings after you load the DS6651 Multi-I/O Module framework.

Common Settings of the FPGA Interface RTI Blocks

Introduction

The common settings of a block dialog are available independently of a hardware-specific framework.

Where to go from here

Information in this section

FPGA_SETUP_BL

Purpose

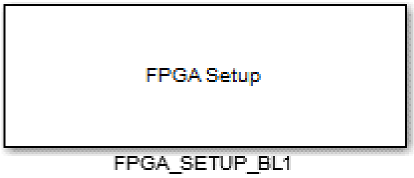
To specify general settings of the FPGA model and to perform specific model actions.

Where to go from here

Information in this section

Block Description (FPGA_SETUP_BL)

Block overview



Purpose

To specify general settings for the FPGA model and to perform FPGA model actions.

Description

The FPGA_SETUP_BL block is used to specify the hardware configuration of the installed FPGA board. It also provides settings, for example, the interface down sample factor, that are relevant for all blocks from the FPGA Interface library. If the FPGA model is complete, you can start several model actions from this block. If you apply the modifications or close the dialog, the settings are announced to all FPGA Interface blocks in the FPGA model.

An FPGA model that must always be encapsulated in a subsystem, must contain one FPGA_SETUP_BL block at its top level. More FPGA_SETUP_BL blocks in the FPGA model will lead to an error when you try to configure interface blocks or to build the model code.

For each FPGA board connected to your processor board, you must provide a separate subsystem with its own FPGA_SETUP_BL block.

The settings in the FPGA_SETUP_BLx block overwrite the settings in an existing Xilinx System Generator Setup block.

Related topics

References

Unit Page (FPGA_SETUP_BL)

Purpose

To specify the FPGA framework.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Note

- If you use a PHS-bus-based system, this setting can be specified to the number of the available DS5203 FPGA Boards in the range 1 … 16. The board number then specifies one specific board.
  The board numbers are ordered by the PHS-bus addresses of the boards. The board with the lowest PHS-bus address gets number 1.
- If you use MicroAutoBox II/III or MicroLabBox, only one FPGA unit is available. The board number has to be set to 1.
- If you use a SCALEXIO system, the number of the available SCALEXIO FPGA base boards is in the range 1 … 16. The board number is used to distinguish between multiple FPGA boards during modeling, but you assign specific FPGA boards in ConfigurationDesk via the hardware resource assignment.

Framework / piggyback Lets you select a framework that is delivered with your hardware. The framework provides the definitions for the implemented on-board I/O features and processor-bus access features. If you want to use other boards or a piggyback plugged to the board, you must select the related framework.

The framework of a piggyback board provides the I/O features of the base board and additionally the I/O extensions of the piggyback board.

Note

The framework consists of a framework INI file. The definitions in this file control the settings that are displayed in the hardware-specific dialog pages.

Framework author Displays the author of the framework INI file that you selected by the **Framework / piggyback** setting.

I/O module 1 … 5 Lets you select the I/O modules that are installed to the I/O module slots of a SCALEXIO FPGA base board. The RTI FPGA Programming Blockset automatically loads the frameworks to support the selected modules.

The number of the I/O module indicates the used I/O module slot. For example: I/O module 1 is connected to the I/O module slot 1 of the FPGA base board.

You can select the following I/O modules:

- *DS2655M1 I/O Module*
- *DS2655M2 I/O Module*
- *DS6651 Multi-I/O Module*
- *Inter-FPGA Interface*

In ConfigurationDesk, you can assign the hardware only if the connected hardware matches module type and module slot. For example, if you have selected a *DS2655M1 I/O Module* for I/O module 1 and a *DS2655M2 I/O Module* for I/O module 3, the hardware registered in ConfigurationDesk must provide at least a DS2655M1 Multi-IO Module on I/O module slot 1 and a DS2655M2 Digital IO Module on I/O module slot 3.

MGT module Lets you select the Multi-Gigabit Transceiver (MGT) module that is connected to the MGT connector. MGT connectors are provided only by the DS6601 and DS6602 FPGA base boards of a SCALEXIO system.

You can select the following modules:

- *Samtex FireFly ECUO-B04 (QSFP+)*

  The RTI FPGA Programming Blockset automatically loads the DS660X_MGT framework to support the selected module.

Author Displays the author of the framework INI file that you selected by the **I/O module** setting.

Related topics

Basics

Implementing Inter-FPGA Communication via I/O Module Slots (RTI FPGA Programming Blockset Guide 📖)

References

Block Description (FPGA_SETUP_BL).. 28
ConfigurationDesk Interface Page (FPGA_SETUP_BL).. 38
FPGA Access Page (FPGA_SETUP_BL).. 35

Parameters Page (FPGA_SETUP_BL)

Purpose

To specify general settings for all the FPGA Interface blocks used and to perform several FPGA model actions.

Dialog settings

FPGA application name Lets you specify the name of the FPGA application if **Same as subsystem / model** is cleared. The name is used as a description for the built FPGA application and can be displayed in your experiment software, such as ControlDesk. For SCALEXIO systems, the FPGA application name is used as custom function block type.

If you use the same FPGA application name for variants of the FPGA model for the MicroAutoBox III or SCALEXIO systems, all the variants have the same custom function block type. This helps you reuse FPGA variants in ConfigurationDesk, because you can exchange the FPGA application without changing the custom function block in the signal chain.

Same as subsystem / model Lets you use the name of the subsystem that contains the FPGA model as FPGA application name. The FPGA model is the subsystem to which the **FPGA_SETUP** block is added to. **FPGA application name** displays the current name.

FPGA build directory Lets you specify the directory for the build process and the build result if **Current directory** is cleared.

Note

During the build process, the path length can exceed the maximum path length specified by Windows. This might lead to error messages during file system operations. For more information, refer to Xilinx Answer Record 52787 (http://www.xilinx.com/support/answers.html).

- Keep the path length of the FPGA build directory short to avoid errors during the build process.

Current directory Lets you use the current MATLAB working directory as FPGA build directory. **FPGA build directory** displays the current build directory.

FPGA IF down-sample factor Lets you specify the sampling rate of all the FPGA Interface blocks in the subsystem. The sampling rate is the ratio of the hardware clock frequency and the given number of cycles. The value must be a positive integer.

Note

In offline simulation mode, the sample period is the product of the offline simulation period times the given number of cycles.

Offline simulation period Lets you specify the simulation period to be used in offline simulation mode. The value must be greater than or equal to the FPGA clock period.

FPGA clock period Displays the hardware clock period of the FPGA in seconds. The value is specified in the framework INI file.

For example, DS5203 (7K325) with onboard I/O: $1e^{-8}$ s ($1 \cdot 10^{-8}$ s = 10 ns)

Description Lets you enter a description of the FPGA model.

The description will be saved in the FPGA model INI file. When you specify the processor interface, the description can be displayed on the **Advanced** page of the processor setup block (**PROC_SETUP_BL**). When you add the FPGA model INI file to ConfigurationDesk, the **FPGA description** property of the FPGA custom function block displays the description.

FPGA model action Lets you select an action that you want to perform on the FPGA model. The selected action is executed when you click **Execute**.

You can choose between:

- FPGA Build

  Starts the build process for the FPGA model. See the MATLAB Command Window to follow the progress of the build process. You will find the resulting FPGA model INI file in `<FPGABuildDirectory>/<ModelName>_rtiFPGA/ini`. The name of a FPGA model INI file is `<FPGAApplicationName>_<ApplicationID>.ini`.

  The build process opens a temporary model `<FPGAModelName>_rtiFPGAtmp` to build the application. The build process closes the temporary model at the end.

- Remote FPGA Build

  Lets you start the build process to use the dSPACE FPGA Build Server for building the FPGA application. Refer to Using an FPGA Build Server (RTI FPGA Programming Blockset Guide 🕮).

  The RTI FPGA Programming Blockset opens a temporary model `<FPGAModelName>_rtiFPGAtmp` and closes the temporary model before the FPGA Build Server starts to build the FPGA application.

- HDL Simulation (Model only)

  Starts the offline simulation of the entire Simulink model based on generated HDL code. The simulation itself uses utilities from the Xilinx Blockset to create an HDL test bench. The test bench compares Simulink simulation results for the FPGA subsystem to those of the Xilinx Vivado Simulator based on the generated HDL code for the FPGA subsystem and therefore checks the correctness of the generated code. The simulation results are presented as text and waveform. For more information, refer to the *Vivado Design Suite User Guide* from Xilinx.

- Timing Analysis (Model only)

  If the build process has detected timing problems, you can execute the timing analysis to analyze the timing behavior of your FPGA subsystem. This timing analysis considers the custom FPGA model without the framework of the platform that is automatically added to your FPGA model during the build process. Timing Analysis starts the timing analysis utility from the Xilinx System Generator. For more information, refer to the *Vivado Design Suite User Guide* from Xilinx.
- Resource Analysis (Model only)

  If the build process detected resource problems, you can execute the resource analysis to analyze the FPGA utilization of the FPGA subsystem. This resource analysis considers the custom FPGA model without the framework of the platform that is automatically added to your FPGA model during the build process. Resource Analysis starts the resource analysis utility from the Xilinx System Generator. For more information, refer to the *Vivado Design Suite User Guide* from Xilinx.
- Show Last Timing Report

  Opens the timing report of the last timing analysis including the timing analysis that is executed during the build process.
- Show Last Resource Utilization Report

  Opens the resource utilization report of the last resource analysis inclusive the resource utilization analyses that is executed during the build process.

Thread(s) Lets you accelerate the build process by performing the FPGA build process in parallel with multiple build threads.

A parallel build needs more RAM than a serial build. If your PC slows down due to a lack of RAM and multiple build threads, you can decrement the number of build threads.

Automatic bus (re-)analysis for FPGA_XDATA blocks Lets you save time by deactivating the analysis of the buses that are connected to the Buffer64 Out and Buffer64 In blocks. The bus analysis checks if the connected bus topology matches the bus topology of the **Data** port.

If you use FPGA_XDATA blocks in the bus transfer mode, the bus analysis is recommended. The bus transfer mode is supported only by SCALEXIO systems and the MicroAutoBox III.

Related topics

References

Subsystem Clocks Page (FPGA_SETUP_BL)

Purpose

To use multiple clock domains for modeling parts of the FPGA design with individual clock periods.

Platform support

All platforms except PHS-bus-based systems support multiple clock domains.

Dialog settings

Subsystem 1 ... 10 Lets you select the subsystem to specify an individual clock period and offline simulation period. The subsystem must be a part of the FPGA model.

If you select a subsystem, the FPGA model use multiple clock domains. To use multiple clock domains, the FPGA design must be partitioned into subsystems. Each subsystem has an individual clock period. For more information, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

Clock period Lets you specify an individual clock period for the selected subsystem. The value must be in the range 1.6e-7 s ... 1.25e-9 s (6.25 MHz ... 800 MHz).

Offl. sim. period Lets you specify the simulation period to be used in offline simulation mode for the selected subsystem. The value must be greater than or equal to the clock period of the selected subsystem.

Related topics

HowTos

How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮)

References

Block Description (FPGA_SETUP_BL).. 28
ConfigurationDesk Interface Page (FPGA_SETUP_BL).......................... 38
FPGA Access Page (FPGA_SETUP_BL).. 35
FPGA_SETUP_BL.. 27
Parameters Page (FPGA_SETUP_BL).. 31
Unit Page (FPGA_SETUP_BL)... 29

FPGA Access Page (FPGA_SETUP_BL)

Purpose

To make FPGA signals accessible for experiment software.

Platform support

The following table gives you an overview of the platforms that support the access of FPGA applications and the methods that are supported.

Platform	FPGA Signal Tracing	Tunable FPGA Constants	FPGA Test Access	FPGA Scaling		
				Scaling Analog I/O Signals	Inverting Digital I/O Signals	Inverting RS232/485 Signals
SCALEXIO	✓[1]	✓	✓	✓	✓	✓
MicroLabBox	✓	✓	✓	✓	✓	–
MicroAutoBox II	✓	✓	✓	✓	✓	–
MicroAutoBox III	✓	✓	✓	✓	✓	–
PHS-bus-based system	–	–	–	–	–	–

[1] ✓ = Supported, – = Not supported

Dialog settings

Enable FPGA tracing Lets you specify that FPGA signals are traceable in the FPGA application.

Traceable signals require FPGA resources, build time, and there are some limitations to experimenting with FPGA variables. Refer to Basics on Tracing FPGA Signals (RTI FPGA Programming Blockset Guide 🕮).

Note

Running applications might stop, if too many FPGA signals are traced

If you trace more than 100 signals with 32-bit values (or 50 signals with 64-bit values) every millisecond with your experiment software, tracing might cause a task overrun that stops the application.

These measures reduce the number of traced FPGA signals per millisecond:

- Disable signal tracing if there is no need to trace FPGA signals.
- Reduce the number of traceable signals. Refer to Basics on Tracing FPGA Signals (RTI FPGA Programming Blockset Guide 📖).
- Reduce the number of signals that that you trace with the experiment software. Only the values of signals traced with an experiment software are sent to the real-time processor and can cause a task overrun.

Enable tunable FPGA constants Lets you enable that tunable FPGA constants are provided and their values can be changed in the experiment software.

This setting can be selected only if FPGA tracing is enabled.

Enable FPGA test access and scaling Lets you enable FPGA test access and FPGA scaling.

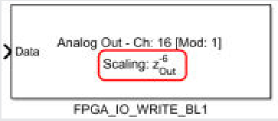
FPGA test access enables intervention points for the experiment software. The intervention points let you set values for the I/O interface and for data values that are exchanged with the processor interface.

FPGA scaling includes scaling, saturating, and inverting FPGA I/O signals with your experiment software. The **Scaling** page of the **FPGA_IO_READ_BL** and **FPGA_IO_WRITE_BL** blocks dialogs provide parameters to specify FPGA scaling.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



This setting is selectable only if FPGA tracing and tunable FPGA constants are enabled.

Select subsystems to be accessed Lets you select the subsystems that can be accessed with your experiment software by clicking the subsystem's name in the tree view. If you click a name, you select or clear the subsystem and its subelements.

The selection affects only FPGA signal tracing and the adjusting of tunable FPGA constants.

Trace input and output ports of subsystem Lets you specify that incoming and outgoing signals of FPGA subsystems are traceable.

Trace all subsystem internal signals Lets you specify that all signals used internally in FPGA subsystems are traceable.

Trace buses Lets you specify that all signals of Simulink Bus Creator blocks and Bus Selector blocks are traceable.

Analyze model Displays the last analysis of the following points:

- The number of signals that will be traceable in the FPGA application and the number of flip-flops that are additionally required to make the signal traceable via variables.
- The number of tunable constants and the number of flip-flops that are additionally required to make the constants tunable.
- The number of ports that support FPGA test access and scaling and the number of flip-flops that are additionally required for FPGA test access and scaling.

Analyze Lets you analyze the current FPGA model on the number of signals and constants that will be accessible with your experiment software.

For analysis, the dialog opens the temporary model `<FPGAModelName>_rtiFPGAtmp`. The dialog closes the temporary model at the end of the analysis.

Added FPGA variables for FPGA test access and scaling

For an overview of the types of FPGA variables that are added to the SDF file to support FPGA test access and scaling, refer to Using the experiment software for access (RTI FPGA Programming Blockset Guide 🕮).

Related topics

Basics

Accessing FPGA Applications with your Experiment Software (RTI FPGA Programming Blockset Guide 🕮)

References

ConfigurationDesk Interface Page (FPGA_SETUP_BL)

Purpose

To generate the processor interface, to configure the FPGA interface, and to export the build results of FPGA models designed with one of the following frameworks:

- SCALEXIO frameworks:
  - *DS2655 (7K160) FPGA Base Board* framework
  - *DS2655 (7K410) FPGA Base Board* framework
  - *DS6601 (KU035) FPGA Base Board* framework
  - *DS6602 (KU15P) FPGA Base Board* framework
- MicroAutoBox III frameworks:
  - *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
  - *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*
  - *FPGA1403Tp1 (7K325) with Engine Control Module (DS1554)*

Export all build results to ConfigurationDesk settings

Build result to export Displays the name of the FPGA model's build result and the date and time the build was executed.

Name of (new) project Lets you enter the name of the ConfigurationDesk project to which the build result will be exported. The name you enter must not contain a dot or whitespace as the first or last character.

Export to new project Lets you export the build result to the specified ConfigurationDesk project. If the specified project does not exist, the framework adds a new project to ConfigurationDesk.

If clicked, the framework performs the following steps:

- Starts ConfigurationDesk.
- Creates a new project in the root directory of ConfigurationDesk.
- Exports the displayed build result of the current FPGA application as a custom function block type to the project.

  If the entire model contains other subsystems with FPGA models for other FPGA boards, the framework also exports the build results of these FPGA models.
- Adds instances of the added custom function block types to the signal chain.

The framework performs the following steps only if the processor model can be separated and the processor interface is implemented:

- Separates and saves the processor models according to the settings of the **Model Separation Setup** block.

  In ConfigurationDesk, the processor models are implemented in Simulink models outside ConfigurationDesk.
- Exports the model interface of the processor models to ConfigurationDesk.
- Maps the function ports of the added custom functions to the model ports of the processor models (behavior model).

For details, refer to How to Export Build Results and Processor Models to ConfigurationDesk Projects (RTI FPGA Programming Blockset Guide 🕮).

Select recent project Lets you select an existing ConfigurationDesk project to which the build result will be exported.

Export to recent project Lets you export the build result to the selected ConfigurationDesk project.

If clicked, the framework performs the following steps:

- Starts ConfigurationDesk.
- Exports the displayed build result of the current FPGA application as a custom function block type to the selected ConfigurationDesk project.

  If the entire model contains other subsystems with FPGA models for other FPGA boards, the framework also exports the build results of these FPGA models.
- Adds instances of the added custom function block types to the signal chain.

The framework performs the following steps only if the processor model can be separated and the processor interface is implemented:

- Separates and saves the processor models according to the settings of the **Model Separation Setup** block.

  In ConfigurationDesk, the processor models are implemented in Simulink models outside ConfigurationDesk.
- Exports the model interface of the processor models to ConfigurationDesk.
- Maps the function ports of the added custom functions to the model ports of the processor models (behavior model).

For instructions, refer to How to Export Build Results and Processor Models to ConfigurationDesk Projects (RTI FPGA Programming Blockset Guide 🕮).

Refresh list Lets you refresh the list with the projects that were recently used in ConfigurationDesk.

If clicked, the framework starts ConfigurationDesk for importing the recently used ConfigurationDesk projects.

Import parameters Lets you import the settings of an instantiated FPGA application (FPGA custom function block). The FPGA application must be added to the ConfigurationDesk project that the **Export to recent project** displays.

The framework imports the following settings:

- Settings of the electrical interface.
- Settings of the scaling parameters.

Clicking the framework starts ConfigurationDesk to import the settings. If several FPGA custom function blocks are added to the signal chain of the project, a dialog lets you select the FPGA custom function block from which you import the settings.

For instructions, refer to How to Update FPGA Models with Imported Parameter Settings (RTI FPGA Programming Blockset Guide 🕮).

Processor Interface settings

Generate Lets you generate model port blocks to implement the processor interface.

If you click **Generate**, the related FPGA model is analyzed and a corresponding processor interface model is created with the blocks of the Model Interface Package for Simulink.

The model port blocks are automatically configured with the corresponding channel numbers, channel names, access type, and format parameters.

After you copy the generated model port blocks to the processor model, you can close the generated interface model without saving.

Select subsystems for multicore support settings

Lets you select the subsystems that can be accessed from a multicore processor application. For instructions, refer to How to Select Subsystems for Multicore Support (MicroAutoBox III, SCALEXIO) (RTI FPGA Programming Blockset Guide 🕮).

Resulting function types Displays the resulting number and types of function blocks after you exported the build results to ConfigurationDesk. For more information on function block types supporting multicore processor applications, refer to Aspects on FPGA Applications Supporting Multicore Processor Applications (RTI FPGA Programming Blockset Guide 🕮).

Related topics

HowTos

How to Generate a Processor Interface (RTI FPGA Programming Blockset Guide 🕮)
How to Prepare the Processor Models for Separating (MicroAutoBox III, SCALEXIO) (RTI FPGA Programming Blockset Guide 🕮)

References

Block Description (FPGA_SETUP_BL)	28
FPGA Access Page (FPGA_SETUP_BL)	35
FPGA_SETUP_BL	27
Parameters Page (FPGA_SETUP_BL)	31
Subsystem Clocks Page (FPGA_SETUP_BL)	34
Unit Page (FPGA_SETUP_BL)	29

FPGA_XDATA_READ_BL

Purpose

To implement read access to processor-bus data in the FPGA model.

Where to go from here

Information in this section

Information in other sections

Settings of the DS1202 FPGA I/O Type 1 frameworks (valid for MicroLabBox)

Settings of the DS2655 FPGA Base Board frameworks (valid for DS2655 (7K160) and DS2655 (7K410))

Settings of the DS6601 (KU035) FPGA Base Board framework

Settings of the DS6602 (KU15P) FPGA Base Board framework

Settings of the DS5203 with onboard I/O frameworks (valid for DS5203 (7K325) and DS5203 (7K410))

Parameters Page (FPGA_XDATA_READ_BL).......... 434
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL).......... 441
To provide detailed information about the selected access type.

Settings of the FPGA1401Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox II)

Parameters Page (FPGA_XDATA_READ_BL).......... 490
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL).......... 497
To provide detailed information about the selected access type.

Settings of the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox II)

Parameters Page (FPGA_XDATA_READ_BL).......... 551
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL).......... 558
To provide detailed information about the selected access type.

Settings of the FPGA1403Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox III)

Parameters Page (FPGA_XDATA_READ_BL).......... 598
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL).......... 606
To provide detailed information about the selected access type.

Settings of the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox III)

Parameters Page (FPGA_XDATA_READ_BL).......... 664
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL).......... 672
To provide detailed information about the selected access type.

Related RTI blocks

FPGA_SETUP_BL...27
To specify general settings of the FPGA model and to perform specific model actions.

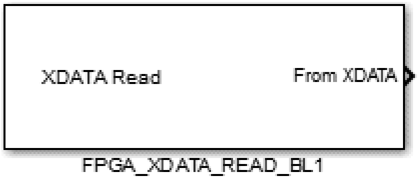
FPGA_XDATA_WRITE_BL...45
To implement write access to processor-bus data in the FPGA model.

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To write data from the processor model to the FPGA model via the board-specific bus.

Block Description (FPGA_XDATA_READ_BL)

Block overview

The figure below shows the block in unconfigured state.



Purpose

To specify read access by the FPGA model to the processor model via processor bus.

Description

The **FPGA_XDATA_READ_BL** block is used to model communication between the FPGA model and the processor model via processor bus. For further information on the processor bus, refer to Features of the RTI FPGA Programming Blockset on page 19. This block can read data from the processor bus. The number of available channels and their configurations depend on the specified framework or piggyback module. The related framework INI file contains the definitions for the available access types and channel numbers on the **Unit** page and their settings on the **Parameters** page.

I/O characteristics

The following table describes the ports of the block:

Port	Description
Output	
From XDATA	Represents the outport until the access type and channel number are defined.

Related topics

References

FPGA_XDATA_READ_BL.. 40
FPGA_XDATA_READ_BL (DS1202 FPGA I/O Type 1 Settings).. 67
FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings).. 122
FPGA_XDATA_READ_BL (DS5203 with Onboard I/O Settings).. 433
FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings).. 170
FPGA_XDATA_READ_BL (DS6602 FPGA Base Board Settings).. 220
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)............ 550
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings).............................. 489
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)............ 663
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings).............................. 597
Unit Page (FPGA_XDATA_READ_BL).. 44

Unit Page (FPGA_XDATA_READ_BL)

Purpose

To specify the general configuration for the FPGA board's processor bus read access.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Access type Lets you select the storage you want to read from. The available access types are defined in the specified framework INI file.

The settings on the **Parameters** page depend on the specified access type.

Channel number Lets you specify a channel number that corresponds to the storage. The available channel numbers are defined in the specified framework INI file and depend on the specified access type.

Subchannel number Lets you specify a subchannel to use Simulink buses for data exchange. Up to 256 subchannels can be used for each channel. Subchannel number 1 of a buffer channel must always be used, the other subchannels can be used in any order.

For more information on subchannels, refer to Using Subchannels for Data Exchange (RTI FPGA Programming Blockset Guide 🕮).

This setting is configurable only if the bus transfer mode is supported by the platform and enabled on the **Parameters** page. For more information, refer to

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

Channel name Lets you specify a user-specific name for the specified channel. The name is displayed in this block and in its corresponding PROC_XDATA_WRITE_BL block in the processor model.

Show Lets you see where the corresponding processor interface block of the selected block is used in the processor model.

Generate Lets you generate a processor interface block that corresponds to the register or buffer.

The block is added to the FPGA model. Move the block to the processor model to use it as a processor interface.

Related topics

HowTos

How to Generate a Processor Interface (RTI FPGA Programming Blockset Guide 🕮)

References

Block Description (FPGA_XDATA_READ_BL).. 43
FPGA_XDATA_READ_BL.. 40
FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings).. 122
FPGA_XDATA_READ_BL (DS5203 with Onboard I/O Settings).. 433
FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings).. 170
FPGA_XDATA_READ_BL (DS6602 FPGA Base Board Settings).. 220
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)................ 550
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)............................ 489
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)................ 663
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings)............................ 597

FPGA_XDATA_WRITE_BL

Purpose To implement write access to processor-bus data in the FPGA model.

Where to go from here

Information in this section

Information in other sections

Settings of the DS1202 FPGA I/O Type 1 frameworks (valid for MicroLabBox)

Parameters Page (FPGA_XDATA_WRITE_BL)..77
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..84
To provide detailed information about the selected access type.

Settings of the DS2655 FPGA Base Board frameworks (valid for DS2655 (7K160) and DS2655 (7K410))

Parameters Page (FPGA_XDATA_WRITE_BL)..132
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..143
To provide detailed information about the selected access type.

Settings of the DS6601 (KU035) FPGA Base Board framework

Parameters Page (FPGA_XDATA_WRITE_BL)..180
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..193
To provide detailed information about the selected access type.

Settings of the DS6602 (KU15P) FPGA Base Board framework

Parameters Page (FPGA_XDATA_WRITE_BL)..230
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..243
To provide detailed information about the selected access type.

Settings of the DS5203 with onboard I/O framework (valid for DS5203 (7K325) and DS5203 (7K410))

Parameters Page (FPGA_XDATA_WRITE_BL)..442
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..449
To provide detailed information about the selected access type.

Settings of the FPGA1401Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox II)

Parameters Page (FPGA_XDATA_WRITE_BL)....................................499
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)....................................506
To provide detailed information about the selected access type.

Settings of the framework FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) (valid for the MicroAutoBox II)

Parameters Page (FPGA_XDATA_WRITE_BL)....................................560
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)....................................567
To provide detailed information about the selected access type.

Settings of the FPGA1403Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox III)

Parameters Page (FPGA_XDATA_WRITE_BL)....................................608
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)....................................616
To provide detailed information about the selected access type.

Settings of the framework FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) (valid for the MicroAutoBox III)

Parameters Page (FPGA_XDATA_WRITE_BL)....................................674
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)....................................682
To provide detailed information about the selected access type.

Related RTI blocks

FPGA_SETUP_BL..27
To specify general settings of the FPGA model and to perform specific model actions.

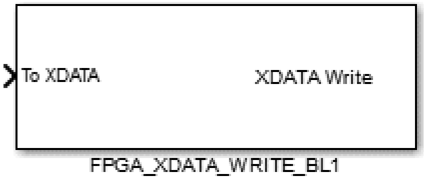
FPGA_XDATA_READ_BL...40
To implement read access to processor-bus data in the FPGA model.

PROC_XDATA_READ_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To read data in the processor model that comes from the FPGA model via the board-specific bus.

Block Description (FPGA_XDATA_WRITE_BL)

Block overview

The figure below shows the block in unconfigured state.



Purpose

To specify write access by the FPGA model to the processor model via processor bus.

Description

The **FPGA_XDATA_WRITE_BL** block is used to model communication between the FPGA model and the processor model via processor bus. For further information on the processor bus, refer to Features of the RTI FPGA Programming Blockset on page 19. This block can write data to the processor bus. The number of available channels and their configurations depends on the specified framework or piggyback module. The settings on the **Parameters** page depend on the specified access type.

I/O characteristics

The following table describes the ports of the block:

Port	Description
Input	
To XDATA	Represents the inport until the access type and the channel number are defined.

Related topics

References

FPGA_XDATA_WRITE_BL 45
FPGA_XDATA_WRITE_BL (DS1202 FPGA I/O Type 1 Settings) 76
FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings) 131
FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings) 441

Unit Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the general configuration for the FPGA board's processor bus write access.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Access type Lets you select the storage you want to write to. The available access types are defined in the specified framework INI file.

The settings on the **Parameters** page depend on the specified access type.

Channel number Lets you specify a channel number that corresponds to the storage. The available channel numbers are defined in the specified framework INI file and depend on the specified access type.

Subchannel number Lets you specify a subchannel to use Simulink buses for data exchange. Up to 256 subchannels can be used for each channel. Subchannel number 1 of a buffer channel must always be used, the other subchannels can be used in any order.

For more information on subchannels, refer to Using Subchannels for Data Exchange (RTI FPGA Programming Blockset Guide 📖).

This setting is configurable only if the bus transfer mode is supported by the platform and enabled on the **Parameters** page. For more information, refer to

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖).

Channel name Lets you specify a user-specific name for the specified channel. The name is displayed in this block and in its corresponding block in the processor model.

Show Lets you see where the corresponding processor interface block of the selected block is used in the entire model.

Generate Lets you generate a processor interface block that correspond to the register or buffer.

The blocks are added to the FPGA model. Move the blocks to the processor model to use them as a processor interface.

Related topics

HowTos

How to Generate a Processor Interface (RTI FPGA Programming Blockset Guide 📖)

References

Block Description (FPGA_XDATA_WRITE_BL).. 48
FPGA_XDATA_WRITE_BL.. 45
FPGA_XDATA_WRITE_BL (DS1202 FPGA I/O Type 1 Settings).. 76
FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings).. 131
FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings).. 441
FPGA_XDATA_WRITE_BL (DS6601 FPGA Base Board Settings).. 180
FPGA_XDATA_WRITE_BL (DS6602 FPGA Base Board Settings).. 230
FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)............... 559
FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)............................ 498
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)............... 673
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)............................ 607

FPGA_IO_READ_BL

Purpose To provide read access to an external device via a physical input channel.

Where to go from here

Information in this section

Information in other sections

Settings of the DS1202 FPGA I/O Type 1 frameworks (valid for MicroLabBox)

Parameters Page (FPGA_IO_READ_BL)..85
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_READ_BL)...92
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..95
To provide detailed information about the selected I/O function.

Settings of the DS2655 FPGA Base Board frameworks (valid for DS2655 (7K160) and DS2655 (7K410))

Parameters Page (FPGA_IO_READ_BL)..144
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..154
To provide detailed information about the selected I/O function.

Settings of the DS6601 (KU035) FPGA Base Board framework

Parameters Page (FPGA_IO_READ_BL)..194
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..205
To provide detailed information about the selected I/O function.

Settings of the DS6602 (KU15P) FPGA Base Board framework

Parameters Page (FPGA_IO_READ_BL)..244
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..255
To provide detailed information about the selected I/O function.

Settings of the DS2655M1 I/O Module framework

Parameters Page (FPGA_IO_READ_BL)..281
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_READ_BL)...287
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..290
To provide detailed information about the selected I/O function.

Settings of the DS2655M2 I/O Module framework

Settings of the DS6651 Multi-I/O Module framework

Settings of the Inter-FPGA Interface framework (valid for SCALEXIO)

Settings of the DS5203 with onboard I/O framework (valid for DS5203 (7K325) and DS5203 (7K410))

Settings of the DS5203 with Multi-I/O Module (DS5203M1) framework (valid for DS5203 (7K325) and DS5203 (7K410))

Settings of the FPGA1401Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox II)

Settings of the framework FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) (valid for the MicroAutoBox II)

Settings of the FPGA1403Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox III)

Settings of the framework FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) (valid for the MicroAutoBox III)

Related RTI blocks

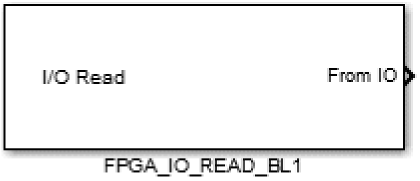
FPGA_SETUP_BL.......27
To specify general settings of the FPGA model and to perform specific model actions.

FPGA_IO_WRITE_BL.......56
To provide write access to an external device via a physical output channel.

Block Description (FPGA_IO_READ_BL)

Block overview

The figure below shows the block in unconfigured state.



Purpose

To implement read access to a physical input channel in the FPGA model.

Description

You can use this block to read data from external devices. The number of available input channels and their configurations depends on the specified framework or piggyback module.

I/O mapping

For information on the I/O mapping, refer to the description of the framework-specific **Parameters** page of this block.

I/O characteristics

The following table describes the ports of the block:

Port	Description
Output	
From IO	Represents the outport until the I/O channel is defined.

Related topics

References

FPGA_IO_READ_BL	50
FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)	85
FPGA_IO_READ_BL (DS2655 FPGA Base Board Settings)	144
FPGA_IO_READ_BL (DS2655M1 I/O Module Settings)	280
FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)	308
FPGA_IO_READ_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)	475
FPGA_IO_READ_BL (DS5203 with Onboard I/O Settings)	450
FPGA_IO_READ_BL (DS6601 FPGA Base Board Settings)	193
FPGA_IO_READ_BL (DS6602 FPGA Base Board Settings)	243
FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings)	346
FPGA_IO_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)	568
FPGA_IO_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)	507
FPGA_IO_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)	683
FPGA_IO_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings)	617
FPGA_IO_READ_BL (Inter-FPGA Interface Settings)	418
FPGA_IO_READ_BL (MGT In Settings)	402
Unit Page (FPGA_IO_READ_BL)	55

Unit Page (FPGA_IO_READ_BL)

Purpose

To specify the I/O type and channel to be used for read access.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Channel name Lets you specify a user-specific name for the specified channel.

I/O type Lets you filter the available input channels.

- All: The list contains all available input channels.
- All other selectable I/O types are defined in the specified framework or piggyback module.

The list below the I/O type setting displays all the available channels of the specified I/O type. Channels that were already assigned to other blocks are not displayed in the list.

Note

The dialog settings on the **Parameters** page depend on the selected I/O function.

Related topics

References

FPGA_IO_WRITE_BL

Purpose

To provide write access to an external device via a physical output channel.

Where to go from here

Information in this section

Block Description (FPGA_IO_WRITE_BL).......... 60
To implement write access to a physical output channel in the FPGA model.

Unit Page (FPGA_IO_WRITE_BL).......... 61
To specify the I/O type and channel to be used for write access.

Information in other sections

Settings of the DS1202 FPGA I/O Type 1 frameworks (valid for MicroLabBox)

Parameters Page (FPGA_IO_WRITE_BL)....................97
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_WRITE_BL)....................114
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)....................118
To provide detailed information about the selected I/O function.

Settings of the DS2655 FPGA Base Board frameworks (valid for DS2655 (7K160) and DS2655 (7K410))

Parameters Page (FPGA_IO_WRITE_BL)....................156
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)....................165
To provide detailed information about the selected I/O function.

Settings of the DS6601 (KU035) FPGA Base Board framework

Parameters Page (FPGA_IO_WRITE_BL)....................206
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)....................215
To provide detailed information about the selected I/O function.

Settings of the DS6602 (KU15P) FPGA Base Board framework

Parameters Page (FPGA_IO_WRITE_BL)....................256
To specify relevant settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)....................275
To provide detailed information about the selected I/O function.

Settings of the DS2655M1 I/O Module framework

Parameters Page (FPGA_IO_WRITE_BL)....................291
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_WRITE_BL)....................302
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)....................306
To provide detailed information about the selected I/O function.

Settings of the DS2655M2 I/O Module framework

Settings of the DS6651 Multi-I/O Module framework

Settings of the Inter-FPGA Interface framework (valid for SCALEXIO)

Settings of the DS5203 with onboard I/O framework (valid for DS5203 (7K325) and DS5203 (7K410))

Settings of the DS5203 with Multi-I/O Module (DS5203M1) framework (valid for DS5203 (7K325) and DS5203 (7K410))

Settings of the FPGA1401Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox II)

Settings of the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox II)

Settings of the FPGA1403Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox III)

Settings of the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox III)

Related RTI blocks

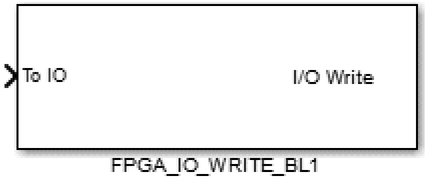
FPGA_SETUP_BL.........27
To specify general settings of the FPGA model and to perform specific model actions.

FPGA_IO_READ_BL.........50
To provide read access to an external device via a physical input channel.

Block Description (FPGA_IO_WRITE_BL)

Block overview

The figure below shows the block in unconfigured state.



Purpose

To implement write access to a physical output channel in the FPGA model.

Description

You can use this block to write data to external devices. The number of available output channels and their configurations depends on the specified framework or piggyback module.

I/O mapping

For information on the I/O mapping, refer to the description of the framework-specific **Parameters** page of this block.

I/O characteristics

The following table describes the ports of the block:

Port	Description
Input	
To IO	Represents the inport until the I/O channel is defined.

Related topics

References

FPGA_IO_WRITE_BL	56
FPGA_IO_WRITE_BL (DS1202 FPGA I/O Type 1 Settings)	96
FPGA_IO_WRITE_BL (DS2655 FPGA Base Board Settings)	155
FPGA_IO_WRITE_BL (DS2655M1 I/O Module Settings)	291
FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)	320
FPGA_IO_WRITE_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)	482
FPGA_IO_WRITE_BL (DS5203 with Onboard I/O Settings)	461
FPGA_IO_WRITE_BL (DS6601 FPGA Base Board Settings)	205
FPGA_IO_WRITE_BL (DS6602 FPGA Base Board Settings)	255
FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)	364
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)	584
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)	525
FPGA_IO_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)	698
FPGA_IO_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)	637
FPGA_IO_WRITE_BL (Inter-FPGA Interface Settings)	425
FPGA_IO_WRITE_BL (MGT Out Settings)	409
Unit Page (FPGA_IO_WRITE_BL)	61

Unit Page (FPGA_IO_WRITE_BL)

Purpose

To specify the I/O type and channel to be used for write access.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Channel name Lets you specify a user-specific name for the specified channel.

I/O type Lets you filter and select the available output channels according to I/O type.

- All: The list contains all available output channels.
- All other selectable I/O types are defined in the specified framework or piggyback module.

The list below the I/O type setting displays all the available channels of the specified I/O type. Channels that were already assigned to other blocks are not displayed in the list.

Note

The dialog settings on the **Parameters** page depend on the selected I/O function.

Related topics

References

FPGA_INT_BL

Purpose

To provide interrupts generated in the FPGA model to the processor model.

Where to go from here

Information in this section

To provide an interrupt for triggering a task in the processor model.

To specify the interrupt channel used to trigger a task in the processor model.

Information in other sections

Settings of the DS1202 FPGA I/O Type 1 frameworks (valid for MicroLabBox)

Parameters Page (FPGA_INT_BL)..119
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..120
To provide detailed information about the selected I/O function.

Settings of the DS2655 FPGA Base Board framework (valid for DS2655 (7K160) and DS2655 (7K410))

Parameters Page (FPGA_INT_BL)..166
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..167
To provide detailed information about the selected I/O function.

Settings of the DS6601 (KU035) FPGA Base Board framework

Parameters Page (FPGA_INT_BL)..217
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..218
To provide detailed information about the selected I/O function.

Settings of the DS6602 (KU15P) FPGA Base Board framework

Parameters Page (FPGA_INT_BL)..277
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..278
To provide detailed information about the selected I/O function.

Settings of the DS5203 with onboard I/O framework (valid for DS5203 (7K325) and DS5203 (7K410))

Parameters Page (FPGA_INT_BL)..471
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..472
To provide detailed information about the selected I/O function.

Settings of the FPGA1401Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox II)

Parameters Page (FPGA_INT_BL)..548
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..549

To provide detailed information about the selected I/O function.

Settings of the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox II)

Parameters Page (FPGA_INT_BL)..594

To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..595

To provide detailed information about the selected I/O function.

Settings of the FPGA1403Tp1 with Multi-I/O Module frameworks (valid for the MicroAutoBox III)

Parameters Page (FPGA_INT_BL)..660

To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..661

To provide detailed information about the selected I/O function.

Settings of the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework (valid for the MicroAutoBox III)

Parameters Page (FPGA_INT_BL)..709

To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..710

To provide detailed information about the selected I/O function.

Related RTI blocks

FPGA_SETUP_BL..27

To specify general settings of the FPGA model and to perform specific model actions.

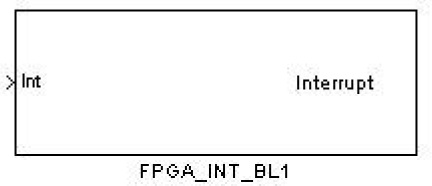
PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)

To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Block Description (FPGA_INT_BL)

Block overview

The figure below shows the block in unconfigured state.



Purpose

To provide an interrupt for triggering a task in the processor application.

Description

The block provides interrupt lines from the FPGA board that you can use to asynchronously trigger a task in the processor application. An interrupt is handled internally and transmitted via interrupt line on the processor bus. For further information on the processor bus, refer to Features of the RTI FPGA Programming Blockset on page 19.

The **Parameters** page of the dialog is empty until you specify an interrupt channel number.

I/O characteristics

The following table describes the ports of the block:

Port	Description
Input	
Int	Represents the inport until the channel number is defined.

Related topics

References

FPGA_INT_BL	62
FPGA_INT_BL (DS1202 FPGA I/O Type 1 Settings)	118
FPGA_INT_BL (DS2655 FPGA Base Board Settings)	165
FPGA_INT_BL (DS5203 with Onboard I/O Settings)	471
FPGA_INT_BL (DS6601 FPGA Base Board Settings)	216
FPGA_INT_BL (DS6602 FPGA Base Board Settings)	276
FPGA_INT_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)	594
FPGA_INT_BL (FPGA1401Tp1 with Multi-I/O Module Settings)	547
FPGA_INT_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)	708
FPGA_INT_BL (FPGA1403Tp1 with Multi-I/O Module Settings)	660
Unit Page (FPGA_INT_BL)	66

Unit Page (FPGA_INT_BL)

Purpose

To specify the interrupt channel used to trigger a task in the processor model.

Dialog settings

Board number Displays the board number in the range 1 … 16. If your system contains several boards of the same type, RTI uses the board number to distinguish between them.

For non-SCALEXIO systems, the mapping of the board number and FPGA application is specified in the processor setup block (PROC_SETUP_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)).

For SCALEXIO systems, you can set a board number to distinguish between multiple FPGA boards during modeling.

Channel number Lets you select a channel number. The range of the selectable interrupt channels depend on the specified framework or piggyback module.

Channels that were already assigned to other blocks are not displayed in the list.

Channel name Lets you specify a user-specific name for the specified channel. The name is displayed in this block and in the corresponding PROC_INT_BL block in the processor model.

Show Lets you see where the corresponding processor interface block of the selected block is used in the processor model.

Generate Lets you generate a processor interface block that corresponds to the register or buffer.

The block is added to the FPGA model. Move the block to the processor model to use it as an processor interface.

Related topics

HowTos

How to Trigger Interrupt-Driven Processor Tasks (RTI FPGA Programming Blockset Guide 🕮)

References

Block Description (FPGA_INT_BL)....................65
FPGA_INT_BL....................62
FPGA_INT_BL (DS1202 FPGA I/O Type 1 Settings)....................118
FPGA_INT_BL (DS2655 FPGA Base Board Settings)....................165
FPGA_INT_BL (DS5203 with Onboard I/O Settings)....................471
FPGA_INT_BL (DS6601 FPGA Base Board Settings)....................216
FPGA_INT_BL (DS6602 FPGA Base Board Settings)....................276
FPGA_INT_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)....................594
FPGA_INT_BL (FPGA1401Tp1 with Multi-I/O Module Settings)....................547
FPGA_INT_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)....................708
FPGA_INT_BL (FPGA1403Tp1 with Multi-I/O Module Settings)....................660

RTI Block Settings for the DS1202 FPGA I/O Type 1

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (DS1202 FPGA I/O Type 1 Settings)

Purpose

To configure read access to the local bus data in the FPGA model when using one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS1202 FPGA I/O Type 1* frameworks provide the following access types that you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from a local bus register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element.

If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Since register groups can be only accessed by one task, you have to explicitly group registers which are used by different tasks.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from a local bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.

- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

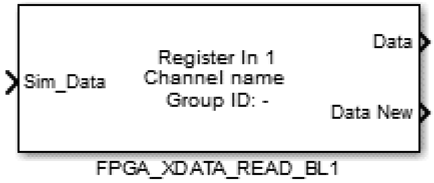
The fraction width is displayed in the Binary point position (or fraction width) setting.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates a local bus data exchange including floating-point to fixed-point data conversion. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from a local bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a

Port	Description
	new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

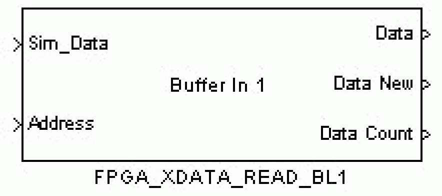
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 69.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are read from the local bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates a local bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater

Port	Description
	than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from a local bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 69.

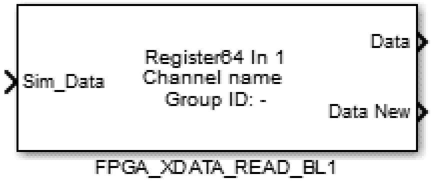
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates a local bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from a local bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

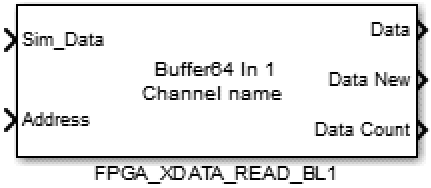
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 69.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the local bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer64 In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates a local bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from a local bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53

Port	Description
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 69.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Related topics

References

Description Page (FPGA_XDATA_READ_BL).. 75
FPGA_XDATA_READ_BL.. 40
FPGA_XDATA_READ_BL (DS1202 FPGA I/O Type 1 Settings).. 67

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *DS1202 FPGA I/O Type 1* frameworks is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_WRITE_BL (DS1202 FPGA I/O Type 1 Settings)

Purpose

To configure write access to the local bus data in the FPGA model when using one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_WRITE_BL)..77
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..84
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)..48
To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)..49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

FPGA_XDATA_READ_BL..40
To implement read access to processor-bus data in the FPGA model.

FPGA_XDATA_READ_BL (DS1202 FPGA I/O Type 1 Settings)...................67
To configure read access to the local bus data in the FPGA model when using one of the DS1202 FPGA I/O Type 1 frameworks.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS1202 FPGA I/O Type 1* frameworks provide the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to a local bus register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to a local bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

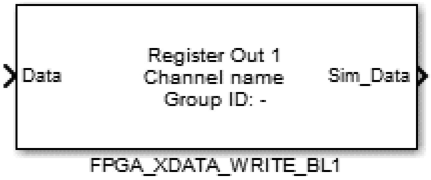
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to a local bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates a local bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

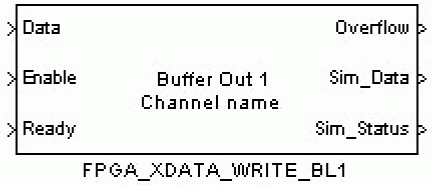
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 77.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the local bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to a local bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via local bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates a local bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector.

Port	Description
	▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 77.

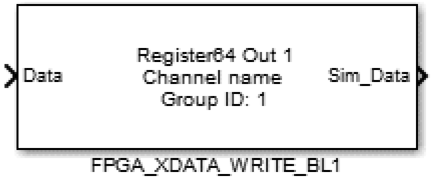
Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to a local bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53

Port	Description
Output	
Sim_Data	Simulates a local bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

> **Note**
>
> The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

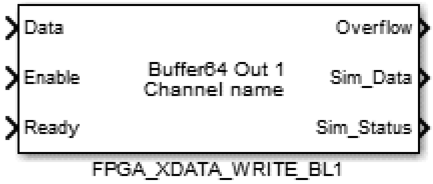
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 77.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the local bus.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to a local bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position>

Port	Description
	All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready o send, even if it is not completely filled. The buffer is switched and the data values are accessible via local bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates a local bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 77.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Related topics

References

Description Page (FPGA_XDATA_WRITE_BL).......................................84
FPGA_XDATA_WRITE_BL.......................................45
FPGA_XDATA_WRITE_BL (DS1202 FPGA I/O Type 1 Settings).......................................76

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *DS1202 FPGA I/O Type 1* frameworks is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_WRITE_BL.......................................45
FPGA_XDATA_WRITE_BL (DS1202 FPGA I/O Type 1 Settings).......................................76
Parameters Page (FPGA_XDATA_WRITE_BL).......................................77

FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The frameworks provide different I/O types, which you can select on the **Unit** page of the block. The number of the available physical connections on MicroLabBox's DS1302 determines the I/O functions that you can select:

- ADC 1 (Class 1) … ADC 24 (Class 1)
- ADC 1 (Class 2) … ADC 8 (Class 2)
- Resolver 1 ... Resolver 2
- Status In
- Proc App Status

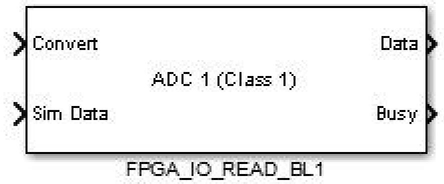
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BLx** block, except for the **Status In** function.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

ADC (Class 1) description

Block display If you select an ADC (Class 1) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
-10 V … +10 V	The output port range is: -32767 … +32767

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Convert	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The `Busy` outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Sim Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page.

Port	Description
	Data type: Double Data width: 1 Input voltage range: -10 V … +10 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 [1] Range: -32767 … +32767 Update rate: 1 Msps
Busy	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 1 to 0, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

I/O mapping The signals are available at the **Analog In** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- Analog I/O A Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 📖)
- Analog In and Analog Out Connectors (BNC) (MicroLabBox Hardware Installation and Configuration 📖)
- Analog In Class 1 Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 📖)

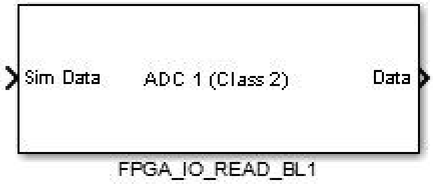
For detailed information on the channel characteristics, refer to Analog Class 1 Inputs (MicroLabBox Hardware Installation and Configuration 📖).

ADC (Class 1) settings

Only common dialog settings. Refer to Common settings on page 86.

ADC (Class 2) description

Block display If you select an ADC (Class 2) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–10 V … +10 V	The output port range is: –32768 … +32767

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -10 V … +10 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 1) Range: -32768 … +32767 Update rate: 10 Msps

1) You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

I/O mapping The signals are available at the **Analog In** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- Analog I/O A Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)
- Analog In and Analog Out Connectors (BNC) (MicroLabBox Hardware Installation and Configuration 🕮)
- Analog In Class 2 Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 🕮)

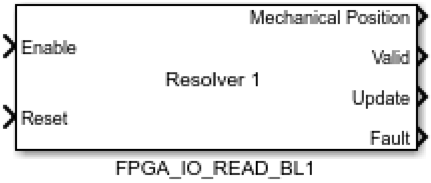
For detailed information on the channel characteristics, refer to Analog Class 2 Inputs (MicroLabBox Hardware Installation and Configuration 🕮).

ADC (Class 2) settings Only common dialog settings. Refer to Common settings on page 86.

Resolver description

Basics on the resolver interface Refer to Resolver Interface (MicroLabBox Features 🕮).

Block display If you select Resolver 1 or Resolver 2 channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for resolver input channels:

Port	Description
Input	
Enable	Enables the excitation voltage: - 0: The resolver interface provides no excitation voltage. - 1: The resolver interface provides the excitation voltage that you set on the **Parameters** page. Data type: UFix_1_0 Data width: 1
Reset	Resets the fault status of the resolver interface that is provided at the **Fault** port to 0: - 0: No reset. - 1: Resets the fault status. Data type: UFix_1_0 Data width: 1
Output	
Mechanical Position	Outputs the position of the resolver sensor as a 16-bit angle value. The 16-bit range of 0 … +65535 corresponds to 0° ... $(360 - 2^{-16})$°. Formula for angle calculation: `alpha[°] = Mechanical Position * 360°/2`^{16}. Data type: UFix_16_0 Range: 0 … +65535 Data width: 1
Valid	Outputs whether the angle position and fault status that are provided by the resolver sensor are valid. This port is used to evaluate whether the resolver interface is ready to receive data from the input signals: - 0: The hardware cannot get data from the input signals. The current values are not valid. - 1: Data values for the position and the fault status has been received. The current values are valid. Data type: UFix_1_0 Data width: 1
Update	Outputs a flag that indicates that a new position value or fault status is available. A high level acknowledges the update. The flag is set high in only one clock cycle.

Port	Description
	Data type: UFix_1_0 Data width: 1
Fault	Outputs the fault status of the resolver interface. The measured position might be valid only if no error is found. Each bit in the 8-bit value represents a specific fault if its value is 1: ▪ Bit 0 (LSB): Configuration parity error ▪ Bit 1: Phase lock ▪ Bit 2: Velocity too high ▪ Bit 3: Loss of tracking ▪ Bit 4: Degradation of signal mismatch ▪ Bit 5: Degradation of signal overrange ▪ Bit 6: Inputs loss of signal ▪ Bit 7: Inputs clipped Data type: UFix_1_0 Data width: 8 For more information on the status information, refer to Resolver Interface (MicroLabBox Features 📖).

I/O mapping The signals are available at the **Resolver** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

Each interface provides six signals:

- 2 differential analog output signals for EXC and $\overline{\text{EXC}}$
- 4 differential analog input signals for SIN, $\overline{\text{SIN}}$, COS and $\overline{\text{COS}}$

For a detailed connector pinout, refer to:

- Resolver Connectors (Sub-D) (MicroLabBox Hardware Installation and Configuration 📖)
- Resolver Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 📖)

For a detailed information on the channel characteristics, refer to Resolver Interfaces (MicroLabBox Hardware Installation and Configuration 📖).

Resolver settings

The **Parameters** page provides the following dialog settings.

Desired excitation frequency Lets you select the frequency of the sine signal to be used for the excitation of the resolver rotor in the range 2,000 Hz … 20,000 Hz in steps of 250 Hz.

Excitation RMS voltage Lets you select the voltage level of the excitation output signal:

- 3.0 V_{RMS}
- 7.0 V_{RMS}
- 10.0 V_{RMS}

Input RMS voltage Lets you select the voltage level of the sine and cosine input signals:

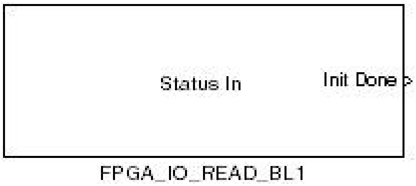
- 1.5 V_{RMS}
- 3.5 V_{RMS}
- 5.0 V_{RMS}

Maximum speed Lets you select the maximum speed to be measured in revolutions per minute. By selecting the speed range, you specify the related resolution.

- 150000 rpm (10 Bit): Specifies a maximum speed of 150,000 rpm and a resolution of 10 bits.
- 60000 rpm (12 Bit): Specifies a maximum speed of 60,000 rpm and a resolution of 12 bits.
- 30000 rpm (14 Bit): Specifies a maximum speed of 30,000 rpm and a resolution of 14 bits.
- 7500 rpm (16 Bit): Specifies a maximum speed of 7500 rpm and a resolution of 16 bits.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

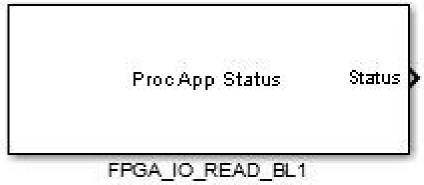
I/O mapping No external connection.

Status In settings

None

Proc App Status description

Block display If you select the Proc App Status channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Status	Outputs the state of the processor application. The state can be used to control the FPGA application. Data type: UFix_1_0 ▪ 0: Processor application is stopped. ▪ 1: Processor application is running.

I/O mapping No external connection.

Proc App Status settings

None

Related topics

References

Description Page (FPGA_IO_READ_BL)........ 95
FPGA_IO_READ_BL........ 50
FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)........ 85
Scaling Page (FPGA_IO_READ_BL)........ 92

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

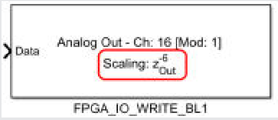
ADC (class 1 and class 2) settings

The following settings on the **Scaling** page are specific to the **ADC** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Status In settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Proc App Status settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Resolver settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_IO_READ_BL)........ 95
FPGA_IO_READ_BL........ 50
FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)........ 85
Parameters Page (FPGA_IO_READ_BL)........ 85

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the

text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *DS1202 FPGA I/O Type 1* frameworks is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL...50
FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)..........................85
Parameters Page (FPGA_IO_READ_BL)..85
Scaling Page (FPGA_IO_READ_BL)...92

FPGA_IO_WRITE_BL (DS1202 FPGA I/O Type 1 Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model when using one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

Parameters Page (FPGA_IO_WRITE_BL)..........97
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_WRITE_BL)..........114
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL)..........118
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Block Description (FPGA_IO_WRITE_BL)..........60
To implement write access to a physical output channel in the FPGA model.

Unit Page (FPGA_IO_WRITE_BL)..........61
To specify the I/O type and channel to be used for write access.

Related RTI blocks

FPGA_IO_READ_BL...50
To provide read access to an external device via a physical input channel.

FPGA_IO_READ_BL (DS1202 FPGA I/O Type 1 Settings)..............................85
To configure read access to analog and digital input signals in the FPGA model when using one of the DS1202 FPGA I/O Type 1 frameworks.

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The frameworks provide the three I/O types *Analog*, *Digital*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on MicroLabBox's DS1302 board determines the I/O functions that you can select:

- DAC 1 (Class 1) … DAC 16 (Class 1)
- Buzzer
- Digital InOut 1 (Class 1) … Digital InOut 48 (Class 1)
- Digital InOut 1 (Class 2) … Digital InOut 12 (Class 2)
- LED Out 1 … LED Out 4
- UART 1 (RS232) … UART 2 (RS232)
- UART 1 (RS422/485) … UART 2 (RS422/485)

Common settings

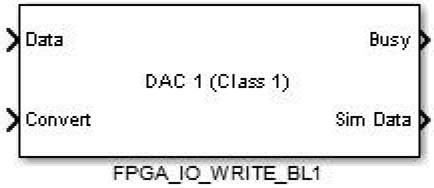
The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out function.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

- If you have selected one of the UART functions, this setting is replaced by function-specific simulation settings.
- For the Digital InOut functions, there are separate settings for enabling the input and output simulation ports.

DAC (Class 1) description

Block display If you select a DAC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog output voltage and the input of the block is:

Output Voltage Range	Simulink Input
-10 V … +10 V	The input port range is: -32767 … +32767

The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data type: Fix_16_0 1) Output voltage range: -32767 … +32767 The range can be exceeded, and saturation is performed to a minimum or maximum value. Hardware update rate: 2.78 Msps If the values are updated at a higher FPGA model rate, intermediate values are not updated by the DAC.
Convert	Triggers the sampling of the D/A converter. When the value is set to 1 for at least one clock cycle, the DAC starts the conversion. The port allows a precise definition of the starting point of DAC sampling. The **Busy** outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Busy	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the DAC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1
Sim Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Port	Description
	Output voltage range: -10 V … +10 V

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The signals are available at the **Analog Out** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- Analog I/O B Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)
- Analog In and Analog Out Connectors (BNC) (MicroLabBox Hardware Installation and Configuration 🕮)
- Analog Out Class 1 Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 🕮)

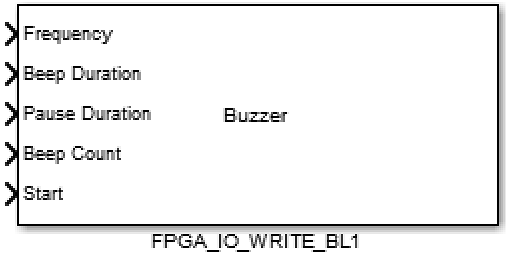
For detailed information on the channel characteristics, refer to Analog Class 1 Outputs (MicroLabBox Hardware Installation and Configuration 🕮).

DAC (Class 1) settings

Only common dialog settings. Refer to Common settings on page 97.

Buzzer description

Block display If you select Buzzer from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Frequency	Specifies the period of the acoustic signal in steps of 40 µs. You calculate the frequency as follows: frequency [Hz] = 1 / period [s] Data type: UFix8_0 Value range: 0 … 255 - 0 : No acoustic signal - 1: 40 µs (25 kHz) - … - 255: 10200 µs (98 Hz)

Port	Description
Beep Duration	Specifies the duration of one beep of the acoustic signal in steps of 10 ms. Data type: UFix8_0 Value range: 0 … 255 ▪ 0: No acoustic signal ▪ 1 ... 254: 10 ms ... 2540 ms ▪ 255: Beep is generated permanently
Pause Duration	Specifies the duration of a pause between two beeps of the buzzer in steps of 10 ms. Data type: UFix8_0 Value range: 0 … 255: 0 ms ... 2550 ms
Beep Count	Specifies the number of beeps to be generated. Data type: UFix8_0 Value range: 0 … 255 ▪ 0: No acoustic signal ▪ 255: The number of beeps is infinite
Start	Starts the buzzer if the value is 1 for one clock cycle. The started buzzer outputs the specified acoustic signal. New values of the **Frequency**, **Beep Duration**, **Pause Duration**, and **Beep Count** ports take effect immediately. For example: If you change the data value of the **Frequency** port to 0, the buzzer stops the generation of an acoustic signal immediately. Data type: UFix1_0 Range: 0 or 1

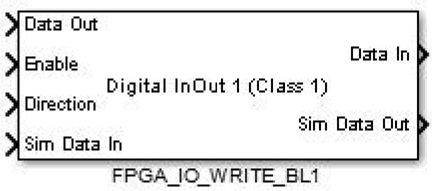
I/O mapping No external connection

Buzzer settings

None

Digital InOut (Class 1) description

Block display If you select a Digital InOut (Class 1) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Out	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is set to the specified high-supply voltage. The hardware output is only driven if the **Enable** port is set to 1, otherwise the output is set to high impedance (High-Z).

Port	Description
	Data Type: UFix_1_0 Update rate: 100 MHz **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1302 board, refer to Digital Class 1 I/O (Bidirectional) (MicroLabBox Hardware Installation and Configuration).
Enable	Controls the hardware output. Data values if the channel is used as digital input: - 0: The **Data In** outport is disabled. - 1: The **Data In** outport outputs the current results of the digital input channel. Data values if the channel is used as digital output: - 0: The hardware is set to High-Z. - 1: The hardware output reacts to the **Data Out** inport. Data type: UFix_1_0
Direction	Controls the direction of the digital channel. Data type: UFix_1_0 - 0: The channel is used as digital input channel. - 1: The channel is used as digital output channel.
Sim Data In	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable digital in simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 2 V
Output	
Data In	Outputs the current results of the digital input channel. Data type: UFix_1_0 The **Data In** outport is enabled only if the **Enable** port is set to 1. - 0: Input voltage falled below the threshold low voltage of 0.8 V. - 1: Input voltage exceeded the threshold high voltage of 2 V. Update rate: 100 MHz **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1302 board, refer to Digital Class 1 I/O (Bidirectional) (MicroLabBox Hardware Installation and Configuration).
Sim Data Out	Simulates an output signal in the same range as that specified for the real output signal.

Port	Description
	Available only if **Enable digital out simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Output voltage: 0 V, 2.5 V, 3.3 V, or 5 V

If the value of the **Data In** inport exceeds the specified data width, only the lowest bit is used.

I/O mapping The signals are available at the **Digital I/O** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- Digital I/O A Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)

  DIO1 ch 1 … DIO1 ch 32
- Digital I/O B Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)

  DIO1 ch 33 … DIO1 ch 48
- Digital I/O Class 1 Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 🕮)

  DIO1 ch 1 … DIO1 ch 48

For detailed information on the channel characteristics, refer to:

- Digital Class 1 I/O (Bidirectional) (MicroLabBox Hardware Installation and Configuration 🕮)

Digital InOut (Class 1) settings

The **Parameters** page provides the following dialog settings.

Invert values Lets you select whether to invert the input and output values of the digital channel.

Input filter Lets you specify the minimum pulse length for detecting a valid input in the range 0 … 10,000,000 ns.

High supply Lets you specify the high-level voltage for the digital outputs. You can select 2.5 V, 3.3 V or 5 V.

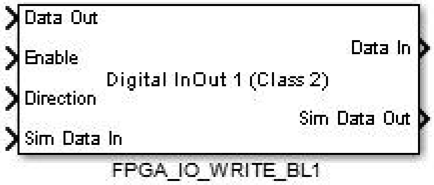
Rising edge delay Lets you specify whether to use a delay for the rising edge detection. The delay can be specified in the range 0 … 65500 ns.

Enable digital out simulation port Lets you enable an outport for offline simulation data. The **Sim Data Out** outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable digital in simulation port Lets you enable an inport for offline simulation data. The **Sim Data In** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model. This port is relevant only if the signal direction is in.

Digital InOut (Class 2) description

Block display If you select a Digital InOut (Class 2) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The I/O characteristics for the Digital InOut (Class 2) function are the same as for the Digital InOut (Class 1) function, refer to I/O characteristics on page 100.

I/O mapping The signals are available at the Digital I/O connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- Digital I/O B Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)
- Digital I/O Class 2 Connectors (Spring-Cage) (MicroLabBox Hardware Installation and Configuration 🕮)

For detailed information on the channel characteristics, refer to:

- Digital Class 2 I/O (Bidirectional) (MicroLabBox Hardware Installation and Configuration 🕮)

Digital InOut (Class 2) settings

The Parameters page provides the following dialog settings.

Invert values Lets you select whether to invert the input and output values of the digital channel.

Input filter Lets you specify the minimum pulse length for detecting a valid input in the range 0 … 10,000,000 ns.

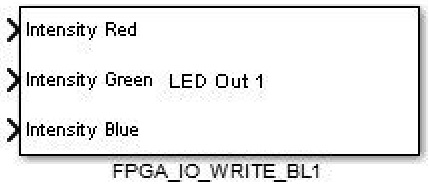
Rising edge delay Lets you specify whether to use a delay for the rising edge detection. The delay can be specified in the range 0 … 65500 ns.

Enable digital out simulation port Lets you enable an outport for offline simulation data. The Sim Data Out outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable digital in simulation port Lets you enable an inport for offline simulation data. The Sim Data In inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model. This port is relevant only if the signal direction is in.

LED Out description

Block display If you select an LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Intensity Red	Controls the color of one of the four FPGA status LEDs placed near the connectors. With this inport you specify the red component of the RGB color value. Data type: UFix_8_0 Value range: 0 … 255
Intensity Green	Controls the color of one of the four FPGA status LEDs placed near the connectors. With this inport you specify the green component of the RGB color value. Data type: UFix_8_0 Value range: 0 … 255
Intensity Blue	Controls the color of one of the four FPGA status LEDs placed near the connectors. With this inport you specify the blue component of the RGB color value. Data type: UFix_8_0 Value range: 0 … 255

If the value of the Data inport exceeds the specified data width, only the lowest 8 bits are used.

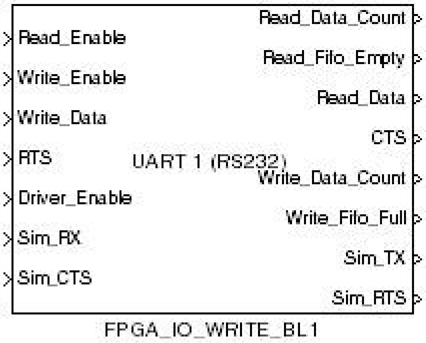
I/O mapping No external connection.

LED Out settings

None

UART (RS232) description

Block display If you select an UART (RS232) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next **Read_Enable** signal. Before you read data from the RX FIFO buffer, you should check the **Read_Fifo_Empty** signal not to be set. The **Read_Fifo_Empty** signal switches one clock cycle after the RX FIFO value has been read. Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle. Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud

Port	Description
	rate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
RTS	Specifies the Ready-To-Send (RTS) signal. The RTS/CTS handshake is handled by the user, the RTS signal is just passed through and adapted to the physical layer. Data type: UFix_1_0 Data width: 1 The hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the Parameters page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_CTS	Simulates the Clear-To-Send (CTS) hardware signal. Available only if **Enable simulation CTS port** is set on the Parameters page. The RTS/CTS handshake is handled by the user. The **Sim_CTS** signal is just passed through to CTS. Data type: Double Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries. If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this.

Port	Description
	Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible. The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
CTS	Outputs the state of the Clear-To-Send (CTS) hardware port. RTS/CTS handshake is handled by the user. CTS is just passed through with conversion to logical 1 and 0. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active The CTS hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive). Range exceeding is not possible.
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already. Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its

Port	Description
	value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the TX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the Parameters page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_RTS	Simulates the Ready-To-Send (RTS) hardware signal. Available only if **Enable simulation RTS port** is set on the Parameters page. The signal is in logical format and only passed through to the RTS signal. Data type: Double Data width: 1 Range: ▪ 0: RTS inactive ▪ 1: RTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

I/O mapping The signals are available at the **RS232 (422/485)** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- RS232 (422/485) Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 📖)

For detailed information on the channel characteristics, refer to:

- Communication Interfaces (MicroLabBox Hardware Installation and Configuration 🕮)

UART (RS232) settings

The **Parameters** page provides the following dialog settings.

Baud rate Lets you select the baud rate in the range 50 … 1,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

> **Note**
>
> The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The **Write_Data** and **Read_Data** ports handle raw bits. If you want to use a parity bit with the **Write_Data** port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the **Read_Data** port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

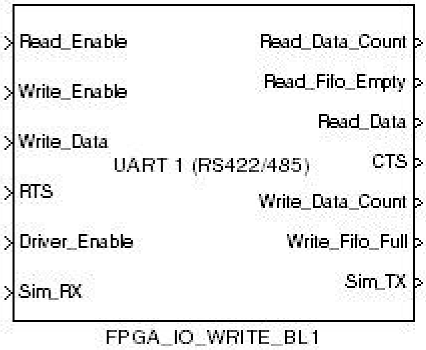
Enable simulation CTS port Lets you enable an inport for offline simulation data. The Sim_CTS inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable simulation RTS port Lets you enable an outport for offline simulation data. The Sim_RTS outport is added to the block to connect it to a Simulink-based I/O environment model.

UART (RS422/485) description

Block display If you select an UART (RS422/485) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next **Read_Enable** signal. Before you read data from the RX FIFO buffer, you should check the **Read_Fifo_Empty** signal not to be set. The **Read_Fifo_Empty** signal switches one clock cycle after the RX FIFO value has been read. Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle. Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud rate.

Port	Description
	The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
Driver_Enable	Specifies to enable the output driver in the transceiver for data transmission. If you use the UART (RS485/422) function in half-duplex mode, the output driver must be disabled while receiving data. Data type: UFix_1_0 Data width: 1
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the Parameters page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries. If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible.

Port	Description
	The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already. Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the RX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the Parameters page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High

Port	Description
	Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

I/O mapping The signals are available at the **RS232 (422/485)** connector.

MicroLabBox has static mapping between I/O signals and I/O pins. The signal name is also printed on the housing of MicroLabBox. You have to consider only the connector panel type.

For a detailed connector pinout, refer to:

- RS232 (422/485) Connector (Sub-D) (MicroLabBox Hardware Installation and Configuration 🕮)

For detailed information on the channel characteristics, refer to:

- Communication Interfaces (MicroLabBox Hardware Installation and Configuration 🕮)

UART (RS422/485) settings

The Parameters page provides the following dialog settings.

Baud rate Lets you select the baud rate in the range 50 … 10,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

> **Note**
>
> The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The **Write_Data** and **Read_Data** ports handle raw bits. If you want to use a parity bit with the **Write_Data** port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the **Read_Data** port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Mode Lets you select the mode for receiving messages.

- Full duplex mode

  You can simultaneously send and receive signals on the UART channel.
- Half duplex mode

  You can send or receive signals on the UART channel, but you cannot do both at the same time.

Termination Lets you enable an internal termination between RX- and RX+ and between TX- and TX+.

Setting	Meaning
Open	No termination
Terminated	The RX and TX signals are terminated via an internal 120 Ω resistor.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........118
FPGA_IO_WRITE_BL........56
FPGA_IO_WRITE_BL (DS1202 FPGA I/O Type 1 Settings)........96
Scaling Page (FPGA_IO_WRITE_BL)........114

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital InOut settings

The following settings on the **Scaling** page are specific to the **Digital InOut** I/O function.

Invert input polarity Lets you invert the measured values of the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A high-level voltage results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-level voltage results in a 0 and vice versa.

Invert output polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If driven with 1, the I/O function outputs the high-level voltage as selected with the **High supply** parameter on the **Parameters** page.

  If driven with 0, the I/O function outputs the low-level voltage.
- **Enabled**:

  If driven with 1, the I/O function outputs the low-level voltage (0 V).

  If driven with 0, the I/O function outputs the high-level voltage.

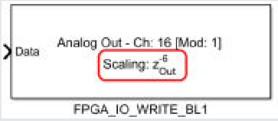
DAC settings

The following settings on the **Scaling** page are specific to the **DAC** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling factor** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling offset** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation minimum value Lets you specify the minimum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum**

value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

UART settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

LED Out settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Buzzer settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

References

Description Page (FPGA_IO_WRITE_BL)..118
FPGA_IO_WRITE_BL..56

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *DS1202 FPGA I/O Type 1* frameworks is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL (DS1202 FPGA I/O Type 1 Settings)

Purpose

To configure the FPGA interrupt channel when using one of the following frameworks:

- *DS1202 FPGA I/O Type 1*
- *DS1202 FPGA I/O Type 1* (Flexible I/O)

Where to go from here

Information in this section

To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..120
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL)..66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

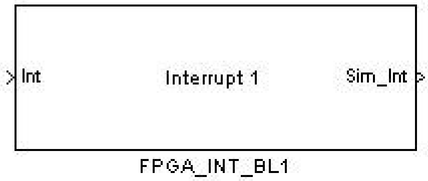
Description

The *DS1202 FPGA I/O Type 1* frameworks provide 32 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line.

Port	Description
	Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The **Sim_Int** outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)..120
FPGA_INT_BL..62
FPGA_INT_BL (DS1202 FPGA I/O Type 1 Settings)..118

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *DS1202 FPGA I/O Type 1* frameworks is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL..62
FPGA_INT_BL (DS1202 FPGA I/O Type 1 Settings)..118
Parameters Page (FPGA_INT_BL)..119

RTI Block Settings for the DS2655 FPGA Base Board Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings)

Purpose

To configure read access to IOCNET data in the FPGA model when using one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)..122
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)..131
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings)................131
To configure write access to IOCNET data in the FPGA model when using one of the DS2655 FPGA Base Board frameworks

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The frameworks of the DS2655 FPGA Boards provide the following access types that you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element.

  If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

  Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Since register groups can be only accessed by one task, you have to explicitly group registers which are used by different tasks.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from an IOCNET buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

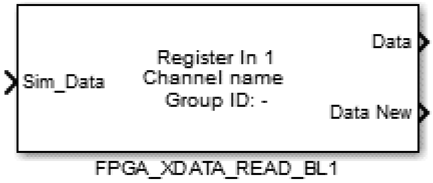
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from a IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format

Port	Description
	UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

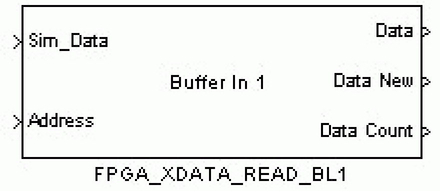
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 123.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page.

Port	Description
	Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 123.

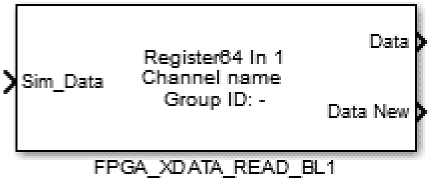
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a

Port	Description
	new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 123.

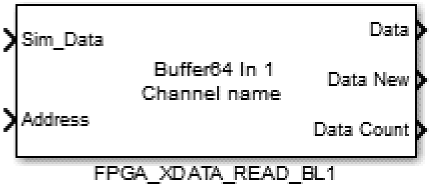
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

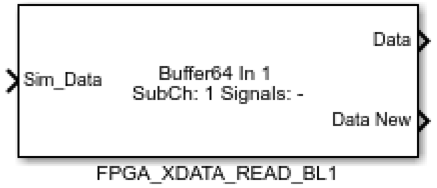
Buffer64 In description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is copied via **Copy bus topology from gcb** on the **Parameters** page. The maximum bit wide is 64 bits. The resolution of the data types is restricted to 53 bits, because the data type of the received bus signals from the processor application is double and the block converts the signals to the signal data types.
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. Available only if the bus transfer mode is disabled on the **Parameters** page.

Port	Description
	The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 123.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Copy bus topology from gcb Lets you copy an existing FPGA bus topology from the selected Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** block.

You cannot copy a bus topology from the processor model, because these topologies do not include the FPGA data types. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port.

This button is clickable only if the bus transfer mode is enabled.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)..131
FPGA_XDATA_READ_BL..40

FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings).. 122

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_READ_BL.. 40
FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings).. 122
Parameters Page (FPGA_XDATA_READ_BL).. 122

FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings)

Purpose

To configure write access to IOCNET data in the FPGA model when using one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_WRITE_BL)..132
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..143
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)..48
To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)..49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

FPGA_XDATA_READ_BL..40
To implement read access to processor-bus data in the FPGA model.

FPGA_XDATA_READ_BL (DS2655 FPGA Base Board Settings)................122
To configure read access to IOCNET data in the FPGA model when using one of the DS2655 FPGA Base Board frameworks.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The frameworks of the DS2655 FPGA Boards provide the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

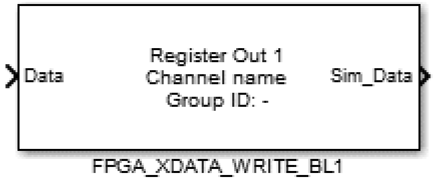
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

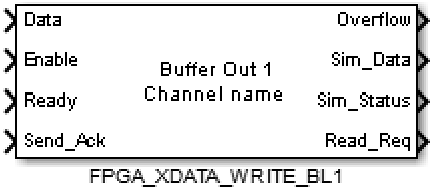
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 133.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a

Port	Description
	new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a

Port	**Description**
	new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: No data transmission is requested. ▪ 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 133.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Note

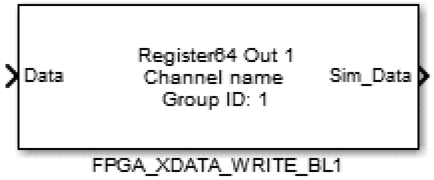
The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the **Read_Req** and **Send_Ack** ports to the block to trigger a processor synchronous data exchange.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

> **Note**
>
> The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 Out settings

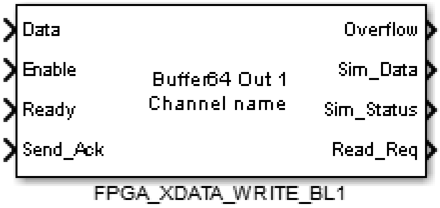
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 133.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

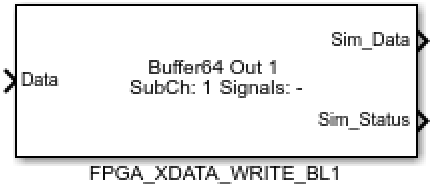
Buffer64 Out description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is set via **Analyze bus topology of input** on the **Parameters** page. The maximum bit width is 64 bits. The resolution of the data type is restricted to 53 bits, because the block converts all data values to double for transmission.
Enable	Specifies the current valid **Data** port value. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is

Port	Description
	completely filled, it is automatically switched, and the data values are stored in a new buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size or the number of bus signals.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32

Port	Description
	Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: No data transmission is requested. ▪ 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged. **Note** In the bus transfer mode, a data transmission request automatically triggers a data transmission.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 133.

Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the Read_Req and Send_Ack ports to the block to trigger a processor synchronous data exchange.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the following parameters are not configurable:

- Binary point position (or fraction width)
- Format
- Buffer size
- Enable Read_Req and Send_Ack ports for explicit data transmit

Analyze bus topology of input Lets you set the Data inport to the bus topology of the connected Simulink bus.

If clicked, the RTI FPGA Programming Blockset analyzes the connected Simulink bus and sets the Data port to a matching bus topology. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the Data port.

This button is clickable only if the bus transfer mode is enabled.

Bus data transmission method Lets you select the method for transmitting data to the processor application if the bus transfer mode is enabled:

- Synchronous to Read_Req method

  Select this method to transmit data that is captured synchronously to the processor task.

  The FPGA application writes data to the swinging buffer when the processor application makes a read request. After the data is written to the buffer, the buffer swings and sends the data to the processor application.

- Free running method

  Select this method if the transmission time is crucial.

  The FPGA application continuously writes data to the swinging buffer. A read request of the processor application immediately transmits the last complete data set of the swinging buffer to the processor application.

The bus data transmission method is selectable only for subchannel 1 and the bus transfer mode must be enabled. The selection applies to all subchannels of the selected channel.

For instructions, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 🕮).

Related topics

HowTos

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_XDATA_WRITE_BL).. 143
FPGA_XDATA_WRITE_BL... 45
FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings).. 131

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_WRITE_BL... 45
FPGA_XDATA_WRITE_BL (DS2655 FPGA Base Board Settings).. 131
Parameters Page (FPGA_XDATA_WRITE_BL).. 132

FPGA_IO_READ_BL (DS2655 FPGA Base Board Settings)

Purpose

To configure read access to information signals in the FPGA model when using one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

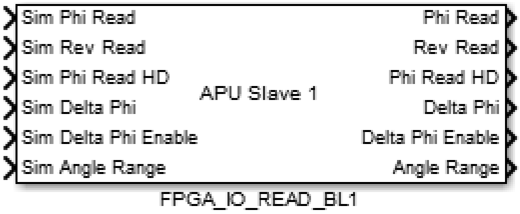
- APU Slave 1 … 6
- CN APP Status
- I-FPGA In 1 (IOCNET) ... I-FPGA In 32 (IOCNET)
- I-FPGA64 In 1 (IOCNET) ... I-FPGA64 In 32 (IOCNET)
- IOCNET Global Time
- Status In

Common settings

No common settings.

APU Slave description

Block display If you select the APU Slave channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the port of the block for digital input channels:

Port	Description
Input	
Sim Phi Read	Simulates the APU bus value for Phi Read. Available only if **Enable simulation Phi Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range depends on the angle range of the simulated APU bus: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns The range can be exceeded, and saturation is performed to a minimum or maximum value.
Sim Rev Read	Simulates the hardware input value for Rev Read. Available only if **Enable simulation Rev Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The 37 Bit range is -2^{36} ... $+2^{36}$ - 1. APU bus clock cycle: 8 ns

Port	Description
	Range exceeding is not possible.
Sim Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Angle Range	Simulates the angle range of the APU bus. Available only if **Enable simulation Angle Range port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range
Output	
Phi Read	Outputs the angle value that **APU Slave** reads from the APU bus. The angle value is independent from the angle range of the APU bus. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the angle range of the APU bus: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value for the APU bus. Data type: Double Data width: 1 The 37 Bit range is -2^{36} ... 2^{36} - 1. The APU bus clock cycle is 8 ns. Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle rangeThe angle range is 0° ... 720° ▪ 1 (high): 360° angle range

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Slave settings

The Parameters page provide the following dialog setting:

Enable Phi Read HD port The Phi Read HD port is available in the block only if you enable it.

Enable Delta Phi port The Delta Phi port is available in the block only if you enable it.

Enable Delta Phi Enable port The Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Phi Read port The Sim Phi Read port is available in the block only if you enable it.

Enable simulation Rev Read port The Sim Rev Read port is available in the block only if you enable it.

Enable simulation Phi Read HD port The Sim Phi Read HD port is available in the block only if you enable it.

Enable simulation Delta Phi port The Sim Delta Phi port is available in the block only if you enable it.

Enable simulation Delta Phi Enable port The Sim Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Angle Range port The Sim Angle Range port is available in the block only if you enable it.

Angle range For the slave APU, you can inherit the angle range of the APU bus or you specify a local angle range independent from the APU bus. The following table shows you the possible combinations of angle range settings.

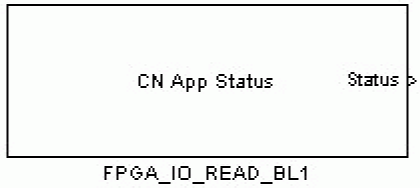
APU Bus Setting	Slave APU Setting	Resulting Angle Range of the Slave APU
360°	Inherit	360°
720°		720°
360°	360°	360°
	720°	720° (1)
720°	360°	360°(2)
	720°	720°

(1) Two engine cycles are required to run through the 720° angle range. If you simulate a four-stroke piston engine, for example, the angle-values of the function block are not clearly related to the camshaft position.

(2) One engine cycle runs twice through the 360° angle range.

CN App Status description

Block display If you select the CN App Status channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Status	Outputs the state of the application that is running on the computation node. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The application on the computation node is stopped. ▪ 1: The application on the computation node is running.

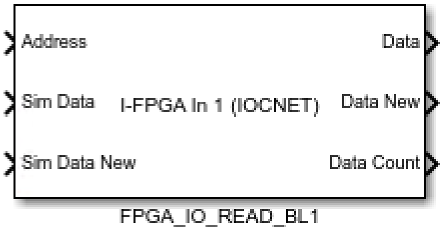
I/O mapping No external connection.

CN App Status settings

None

I-FPGA In (IOCNET) description

Block display If you select the I-FPGA In (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the **Data** outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the **Buffer size** on the **Parameters** page. The address range with valid data values can be derived from the value of the **Data Count** port.

Port	Description
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the **Data** port and written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024 Range: Single precision value range
Sim Data New	Simulates the reception of new data values. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated. ▪ 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.
Output	
Data	Outputs a 32-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_32_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1 The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA In (IOCNET) settings

The following dialog settings are specific for the **I-FPGA In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 145.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

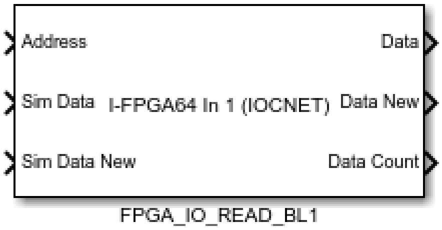
The **Sim Data** inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the **Data** outport.

The **Sim Data New** inport is added to the block.

I-FPGA64 In (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	**Description**
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the **Data** outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the **Buffer size** on the **Parameters** page. The address range with valid data values can be derived from the value of the **Data Count** port.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the **Data** port and written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 512 Range: Double precision value range
Sim Data New	Simulates the reception of new data values. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated. ▪ 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.
Output	
Data	Outputs a 64-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_64_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1

Port	Description
	The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.

The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 In (IOCNET) settings

The following dialog settings are specific for the **I-FPGA64 In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 145.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 512. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

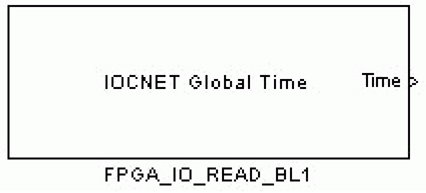
The **Sim Data** inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the **Data** outport.

The **Sim Data New** inport is added to the block.

IOCNET Global Time description

Block display If you select the IOCNET Global Time channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Time	Outputs the number of hardware ticks that occurred since the SCALEXIO system power was switched to on. If you use a multiprocessor system, the value is set to zero each time an application is reloaded and restarted. Data type: 56 bit Tick step-width: 8.5 ns

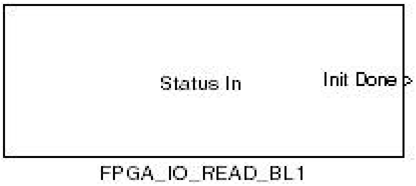
I/O mapping No external connection.

IOCNET Global Time settings

None

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

I/O mapping No external connection.

Status In settings

None

Related topics

References

Description Page (FPGA_IO_READ_BL) 154
FPGA_IO_READ_BL 50
FPGA_IO_READ_BL (DS2655 FPGA Base Board Settings) 144

Scaling Page (FPGA_IO_READ_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL (DS2655 FPGA Base Board Settings)

Purpose

To configure write access to information signals in the FPGA model when using one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

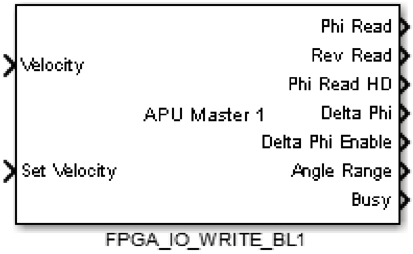
- APU Master 1 … APU Master 6
- I-FPGA Out (IOCNET)
- I-FPGA64 Out (IOCNET)
- LED Out

Common settings

None

APU Master 1 … 6 description

Block display If you select the APU Master channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Velocity	Specifies a velocity value in degree/second to be applied as APU Master speed. The value will be applied if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: Fix_32_10 Data width: 1 Range: -1,200,000 °/s ... +1,200,000 °/s Range exceeding is not possible. The port is saturated at the higher or lower limit.
Set Velocity	Specifies the current value of **Velocity** as new velocity value. The new value is set only if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: UFix_1_0 Data width: 1 To distribute and execute a new velocity value takes at least 10 µs, depending on the IOCNET structure.

Port	Description
Output	
Phi Read	Outputs the angle counter value of the APU that the **APU Master** writes to the APU bus. The step size of the angle counter is approximately 0.011°. The step size is independent from the angle range. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the **Angle range** on the **Parameters** page: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value that the **APU Master** reads from the APU bus. Data type: Fix_37_0 Data width: 1 The 37 Bit range is -2^{36} ... 2^{36} - 1. APU bus clock cycle: 8 ns Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. You set the **Angle range** on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range
Busy	Outputs whether **APU Master** is busy to set the last velocity value. Data type: UFix_1_0 Data width: 1 If **Busy** is 1 (high), new velocity values cannot be set. **Busy** stays active for at least 10 µs depending on the IOCNET structure.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Master 1 … 6 settings

The **Parameters** page provide the following dialog setting:

Angle range Lets you select the angle range of the APU in degree.

- 360: The angle range is 360° and cannot be changed in ConfigurationDesk.
- 720: The angle range is 720° and cannot be changed in ConfigurationDesk.

- Individual: The **Angle range** property of the FPGA custom function block in ConfigurationDesk lets you set the angle range of the APU. The default value is 720°.

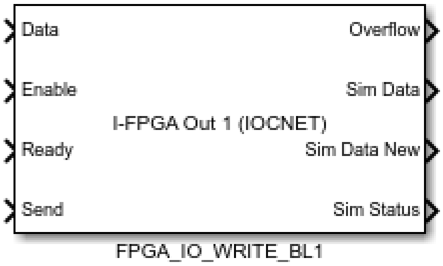
Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Enable advanced ports The internal ports for the dSPACE Electric Motor HIL Solution are available in the block only if you enable it.

I-FPGA Out (IOCNET) description

Block display If you select the I-FPGA Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit raw data value to be written to an IOCNET buffer. Data type: UFix_32_0 Data width: 1
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock

Port	Description
	cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: - 0: The buffer is not ready to send. - 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1 Values: - 0: **Data** values are not acknowledged for transmission. - 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: - 0: No overflow occurred. - 1: An overflow occurred. This value is set for one clock cycle.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a simulated IOCNET buffer, including fixed-point to floating-point data conversion. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024, depends on the **Buffer size** parameter. Range: Single precision value range
Sim Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport.

Port	Description
	Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. ▪ Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA Out (IOCNET) settings

The following dialog settings are specific for the **I-FPGA Out** I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The **Sim Data** inport is added to the block.

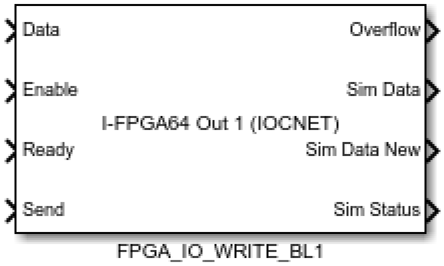
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the **Sim Data** inport.

The **Sim Data New** outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The **Sim Status** outport is added to the block.

I-FPGA64 Out (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit raw data value to be written to an IOCNET buffer. Data type: UFix_64_0 Data width: 1
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 512, depends on the **Buffer size** parameter. Range: Double precision value range
Sim_Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. ▪ Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 Out (IOCNET) settings

The following dialog settings are specific for the **I-FPGA Out** I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 512.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The **Sim Data** inport is added to the block.

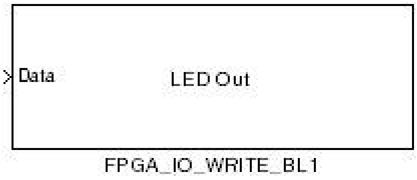
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the **Sim Data** inport.

The **Sim Data New** outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The **Sim Status** outport is added to the block.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the LED on the DS2655 FPGA Base Board. Data type: UFix_1_0 Values: ▪ 0: LED lights green. ▪ 1: LED lights orange.

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used (=1).

I/O mapping No external connection.

LED Out settings

None

Related topics

References

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (DS2655 FPGA Base Board Settings)........ 155
Parameters Page (FPGA_IO_WRITE_BL)........ 156

FPGA_INT_BL (DS2655 FPGA Base Board Settings)

Purpose

To configure the FPGA interrupt channel when using one of the following frameworks:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL).. 166
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL).. 167
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL).. 66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

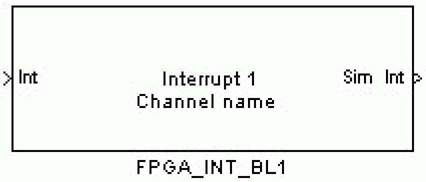
Description

The DS2655 FPGA Base Board provides 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

---

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The **Sim_Int** outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)........ 167
FPGA_INT_BL........ 62
FPGA_INT_BL (DS2655 FPGA Base Board Settings)........ 165

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *DS2655 (7K160) FPGA Base Board*
- *DS2655 (7K410) FPGA Base Board*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL.. 62
FPGA_INT_BL (DS2655 FPGA Base Board Settings).. 165
Parameters Page (FPGA_INT_BL)... 166

RTI Block Settings for the DS6601 FPGA Base Board Framework

Introduction

The block dialogs provide hardware-specific settings after you load the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings)................. 170
To configure read access to IOCNET data in the FPGA model when using the DS6601 (KU035) FPGA Base Board framework.

FPGA_XDATA_WRITE_BL (DS6601 FPGA Base Board Settings)................ 180
To configure write access to IOCNET data in the FPGA model when using the DS6601 (KU035) FPGA Base Board framework.

FPGA_IO_READ_BL (DS6601 FPGA Base Board Settings)....................... 193
To configure read access to analog and digital input signals in the FPGA model when using the DS6601 (KU035) FPGA Base Board framework.

FPGA_IO_WRITE_BL (DS6601 FPGA Base Board Settings)...................... 205
To configure write access to analog and digital output signals in the FPGA model when using the DS6601 (KU035) FPGA Base Board framework.

FPGA_INT_BL (DS6601 FPGA Base Board Settings)................................ 216
To configure the FPGA interrupt channel when using the DS6601 (KU035) FPGA Base Board framework.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

RTI Block Settings for the DS2655M1 I/O Module Framework................ 279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

RTI Block Settings for the DS2655M2 I/O Module Framework................ 307
The block dialogs provide hardware-specific settings after you load the DS2655M2 I/O Module framework.

RTI Block Settings for the DS6651 Multi-I/O Module Framework............ 345
The block dialogs provide hardware-specific settings after you load the DS6651 Multi-I/O Module framework.

RTI Block Settings for the Inter-FPGA Interface Framework.................... 417
The block dialogs provide hardware-specific settings after you load the Inter-FPGA Interface framework for SCALEXIO.

RTI Block Settings for the DS660X_MGT Framework..........................401
The block dialogs provide hardware-specific settings after you load the DS660X_MGT framework for SCALEXIO.

FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings)

Purpose

To configure read access to IOCNET data in the FPGA model when using the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (DS6601 FPGA Base Board Settings)................180
To configure write access to IOCNET data in the FPGA model when using the DS6601 (KU035) FPGA Base Board framework.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS6601 (KU035) FPGA Base Board* framework provides the following access types that you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element.

  If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

  Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Since register groups can be only accessed by one task, you have to explicitly group registers which are used by different tasks.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from an IOCNET buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

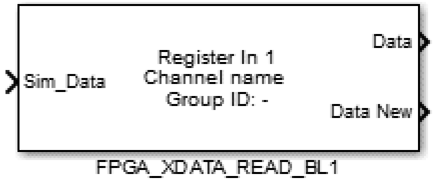
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Port	Description
Output	
Data	Outputs a 32-bit data value to be read from a IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

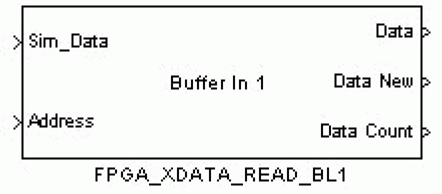
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 171.

Register group ID Lets you specify a number in the range 1 ... 63 to create register groups. Registers which you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 171.

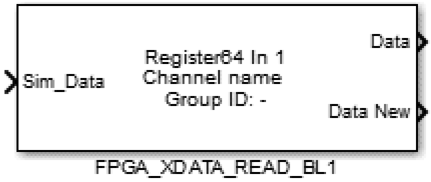
Buffer size Lets you specify the size of the buffer in the range 1 ... 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related dialog settings.

Port	Description
	Data type: - Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. - Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 171.

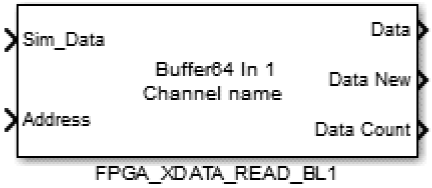
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

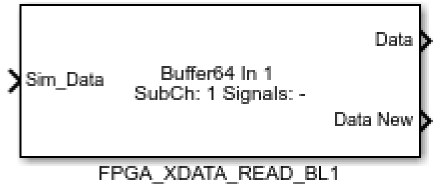
Buffer64 In description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is copied via **Copy bus topology from gcb** on the **Parameters** page. The maximum bit wide is 64 bits. The resolution of the data types is restricted to 53 bits, because the data type of the received bus signals from the processor

Port	Description
	application is double and the block converts the signals to the signal data types.
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 171.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Copy bus topology from gcb Lets you copy an existing FPGA bus topology from the selected Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** block.

You cannot copy a bus topology from the processor model, because these topologies do not include the FPGA data types. For instructions, refer to How to

Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port. This button is clickable only if the bus transfer mode is enabled.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)	179
FPGA_XDATA_READ_BL	40
FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings)	170

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6601 (KU035) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_READ_BL	40
FPGA_XDATA_READ_BL (DS6601 FPGA Base Board Settings)	170
Parameters Page (FPGA_XDATA_READ_BL)	171

FPGA_XDATA_WRITE_BL (DS6601 FPGA Base Board Settings)

Purpose

To configure write access to IOCNET data in the FPGA model when using the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS6601 (KU035) FPGA Base Board* framework provides the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** inport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

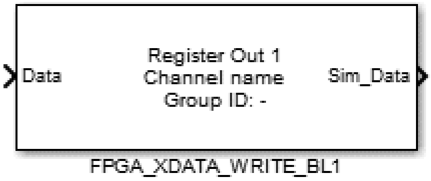
The fraction width is displayed in the Binary point position (or fraction width) setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The Sim_Data outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data and Sim_Status. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

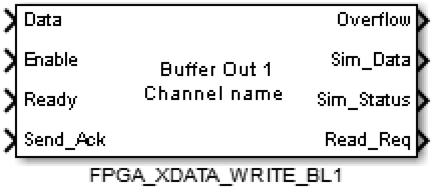
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 181.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are

Port	Description
	therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred.

Port	Description
	▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: No data transmission is requested. ▪ 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 181.

Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

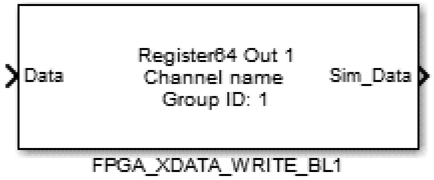
> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the **Read_Req** and **Send_Ack** ports to the block to trigger a processor synchronous data exchange.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion.

Port	Description
	Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

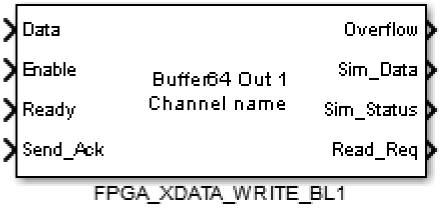
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 181.

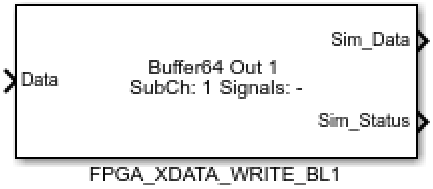
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is set via **Analyze bus topology of input** on the **Parameters** page. The maximum bit width is 64 bits. The resolution of the data type is restricted to 53 bits, because the block converts all data values to double for transmission.
Enable	Specifies the current valid **Data** port value. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock

Port	Description
	cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size or the number of bus signals.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read.

Port	Description
	Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you

Port	Description
	write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: No data transmission is requested. ▪ 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged. **Note** In the bus transfer mode, a data transmission request automatically triggers a data transmission.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 181.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the **Read_Req** and **Send_Ack** ports to the block to trigger a processor synchronous data exchange.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the following parameters are not configurable:

- Binary point position (or fraction width)
- Format
- Buffer size
- Enable Read_Req and Send_Ack ports for explicit data transmit

Analyze bus topology of input Lets you set the Data inport to the bus topology of the connected Simulink bus.

If clicked, the RTI FPGA Programming Blockset analyzes the connected Simulink bus and sets the Data port to a matching bus topology. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the Data port.

This button is clickable only if the bus transfer mode is enabled.

Bus data transmission method Lets you select the method for transmitting data to the processor application if the bus transfer mode is enabled:

- Synchronous to Read_Req method

  Select this method to transmit data that is captured synchronously to the processor task.

  The FPGA application writes data to the swinging buffer when the processor application makes a read request. After the data is written to the buffer, the buffer swings and sends the data to the processor application.
- Free running method

  Select this method if the transmission time is crucial.

  The FPGA application continuously writes data to the swinging buffer. A read request of the processor application immediately transmits the last complete data set of the swinging buffer to the processor application.

The bus data transmission method is selectable only for subchannel 1 and the bus transfer mode must be enabled. The selection applies to all subchannels of the selected channel.

For instructions, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 🕮).

Related topics

HowTos

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_XDATA_WRITE_BL).......... 193
FPGA_XDATA_WRITE_BL.......... 45
FPGA_XDATA_WRITE_BL (DS6601 FPGA Base Board Settings).......... 180

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6601 (KU035) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL (DS6601 FPGA Base Board Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_IO_READ_BL)..194
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_READ_BL)..204
The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Description Page (FPGA_IO_READ_BL)..205
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- APU Slave 1 … APU Slave 6
- CN APP Status
- I-FPGA In 1 (IOCNET) ... I-FPGA In 32 (IOCNET)
- I-FPGA64 In 1 (IOCNET) ... I-FPGA64 In 32 (IOCNET)
- IOCNET Global Time
- Status In
- Temperature

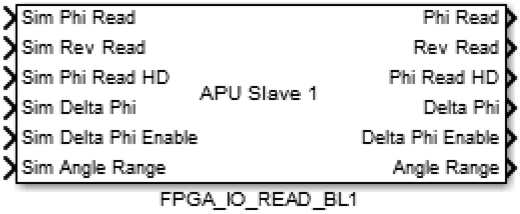
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to simulation data coming from a Simulink-based I/O environment model.

APU Slave 1 ... 6 description

Block display If you select the APU Slave channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the port of the block for digital input channels:

Port	Description
Input	
Sim Phi Read	Simulates the APU bus value for Phi Read. Available only if **Enable simulation Phi Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range depends on the angle range of the simulated APU bus: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns The range can be exceeded, and saturation is performed to a minimum or maximum value.
Sim Rev Read	Simulates the hardware input value for Rev Read. Available only if **Enable simulation Rev Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The 37 Bit range is -2^{36} ... $+2^{36}$ - 1. APU bus clock cycle: 8 ns Range exceeding is not possible.
Sim Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Angle Range	Simulates the angle range of the APU bus. Available only if **Enable simulation Angle Range port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range

Port	Description
Output	
Phi Read	Outputs the angle value that APU Slave reads from the APU bus. The angle value is independent from the angle range of the APU bus. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the angle range of the APU bus: ▪ 360° angle range: 0 … 32767 (2^{15}-1) ▪ 720° angle range: 0 … 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value for the APU bus. Data type: Double Data width: 1 The APU bus clock cycle is 8 ns. The 37 Bit range is -2^{36} … 2^{36} - 1. Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle rangeThe angle range is 0° … 720° ▪ 1 (high): 360° angle range

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Slave 1 … 6 settings

The Parameters page provides the following dialog setting:

Enable Phi Read HD port The Phi Read HD port is available in the block only if you enable it.

Enable Delta Phi port The Delta Phi port is available in the block only if you enable it.

Enable Delta Phi Enable port The Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Phi Read port The Sim Phi Read port is available in the block only if you enable it.

Enable simulation Rev Read port The Sim Rev Read port is available in the block only if you enable it.

Enable simulation Phi Read HD port The Sim Phi Read HD port is available in the block only if you enable it.

Enable simulation Delta Phi port The Sim Delta Phi port is available in the block only if you enable it.

Enable simulation Delta Phi Enable port The Sim Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Angle Range port The Sim Angle Range port is available in the block only if you enable it.

Angle range For the slave APU, you can inherit the angle range of the APU bus or you specify a local angle range independent from the APU bus. The following table shows you the possible combinations of angle range settings.

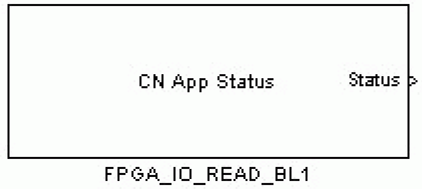
APU Bus Setting	Slave APU Setting	Resulting Angle Range of the Slave APU
360°	Inherit	360°
720°		720°
360°	360°	360°
	720°	720° 1)
720°	360°	360°2)
	720°	720°

1) Two engine cycles are required to run through the 720° angle range. If you simulate a four-stroke piston engine, for example, the angle-values of the function block are not clearly related to the camshaft position.

2) One engine cycle runs twice through the 360° angle range.

CN App Status description

Block display If you select the CN App Status channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Status	Outputs the state of the application that is running on the computation node. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The application on the computation node is stopped.

Port	Description
	- 1: The application on the computation node is running.

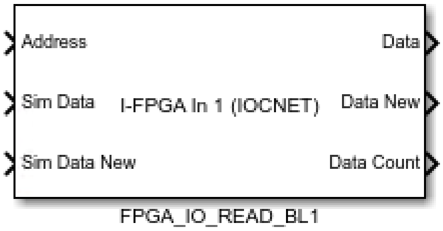
I/O mapping No external connection.

CN App Status settings

None

I-FPGA In (IOCNET) description

Block display If you select the I-FPGA In (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the **Data** outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the **Buffer size** on the **Parameters** page. The address range with valid data values can be derived from the value of the **Data Count** port.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the **Data** port and written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024 Range: Single precision value range
Sim Data New	Simulates the reception of new data values. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated. ▪ 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.

Port	Description
Output	
Data	Outputs a 32-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_32_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1 The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA In (IOCNET) settings

The following dialog settings are specific for the **I-FPGA In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 194.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

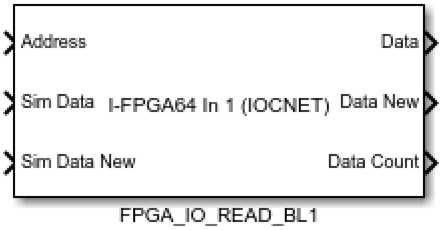
The Sim Data inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the Data outport.

The Sim Data New inport is added to the block.

I-FPGA64 In (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the Data outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the Buffer size on the Parameters page. The address range with valid data values can be derived from the value of the Data Count port.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the Data port and written to a simulated IOCNET buffer. Available only if the Enable simulation data port is set on the Parameters page. Data type: Double Data width: 1 … 512 Range: Double precision value range
Sim Data New	Simulates the reception of new data values. Available only if the Enable simulation data new port is set on the Parameters page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated.

Port	Description
	- 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.
Output	
Data	Outputs a 64-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_64_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1 The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 In (IOCNET) settings The following dialog settings are specific for the **I-FPGA64 In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 194.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 512. The maximum range of the Address inport depends on the buffer size.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

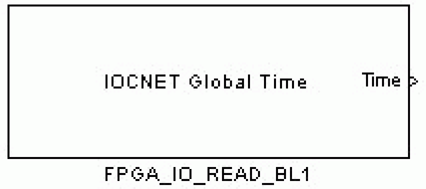
The Sim Data inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the Data outport.

The Sim Data New inport is added to the block.

IOCNET Global Time description

Block display If you select the IOCNET Global Time channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Time	Outputs the number of hardware ticks that occurred since the SCALEXIO system power was switched to on. If you use a multiprocessor system, the value is set to zero each time an application is reloaded and restarted. Data type: 56 bit Tick step-width: 8.5 ns

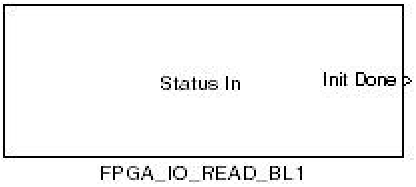
I/O mapping No external connection.

IOCNET Global Time settings

None

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

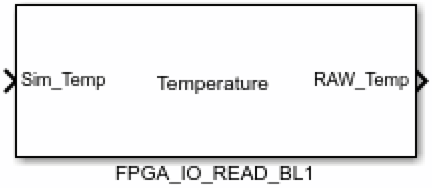
I/O mapping No external connection.

Status In settings

None

Temperature description

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C

Port	Description
	The range can be exceeded. The values are then saturated to the minimum or maximum values.
Output	
RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature. Equation for calculating the die temperature: `Temperature [°C] = (float)(Temperature[hex] & 0xFFF0) · 503.975 / 65536 - 273.15` Data type: UFix_16_0 Data width: 1 Value range: 0 ... 65536

I/O mapping No external connection.

Temperature settings

Only common dialog settings. Refer to Common settings on page 194.

Related topics

References

Description Page (FPGA_IO_READ_BL)........205
FPGA_IO_READ_BL........50
FPGA_IO_READ_BL (DS6601 FPGA Base Board Settings)........193

Scaling Page (FPGA_IO_READ_BL)

Description

The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6601 (KU035) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (DS6601 FPGA Base Board Settings).. 193
Parameters Page (FPGA_IO_READ_BL).. 194

FPGA_IO_WRITE_BL (DS6601 FPGA Base Board Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model when using the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

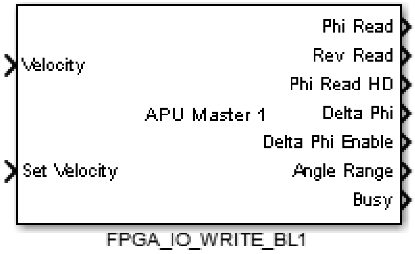
- APU Master 1 … APU Master 6
- I-FPGA Out (IOCNET)
- I-FPGA64 Out (IOCNET)
- LED Out

Common settings

None

APU Master 1 … 6 description

Block display If you select the APU Master channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Velocity	Specifies a velocity value in degree/second to be applied as APU Master speed. The value will be applied if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: Fix_32_10 Data width: 1 Range: -1,200,000 °/s … +1,200,000 °/s Range exceeding is not possible. The port is saturated at the higher or lower limit.
Set Velocity	Specifies the current value of **Velocity** as new velocity value. The new value is set only if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: UFix_1_0 Data width: 1 To distribute and execute a new velocity value takes at least 10 µs, depending on the IOCNET structure.
Output	
Phi Read	Outputs the angle counter value of the APU that the **APU Master** writes to the APU bus. The step size of the angle counter is approximately 0.011°. The step size is independent from the angle range. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the **Angle range** on the **Parameters** page.: ▪ 360° angle range: 0 … 32767 (2^{15}-1) ▪ 720° angle range: 0 … 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value that **APU Master** reads from the APU bus. Data type: Fix_37_0 Data width: 1 The 37 Bit range is -2^{36} … 2^{36} - 1. APU bus clock cycle: 8 ns

Port	Description
	Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. You set the angle range on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range
Busy	Outputs whether **APU Master** is busy to set the last velocity value. Data type: UFix_1_0 Data width: 1 If the **Busy** port is 1 (high), new velocity values cannot be set. The **Busy** port stays active for at least 10 µs depending on the IOCNET structure.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Master 1 … 6 settings

The **Parameters** page provide the following dialog setting:

Angle range Lets you select the angle range of the APU in degree.

- 360: The angle range is 360° and cannot be changed in ConfigurationDesk.
- 720: The angle range is 720° and cannot be changed in ConfigurationDesk.
- Individual: The **Angle range** property of the FPGA custom function block in ConfigurationDesk lets you set the angle range of the APU. The default value is 720°.

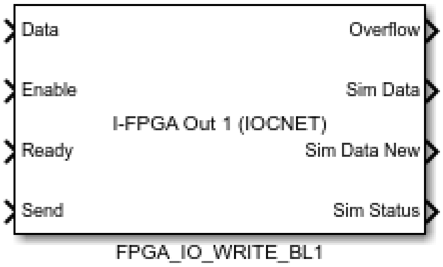
Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Enable advanced ports The internal ports for the dSPACE Electric Motor HIL Solution are available in the block only if you enable it.

I-FPGA Out (IOCNET) description

Block display If you select the I-FPGA Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit raw data value to be written to an IOCNET buffer. Data type: UFix_32_0 Data width: 1
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a simulated IOCNET buffer, including fixed-point to floating-point data conversion. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024, depends on the **Buffer size** parameter. Range: Single precision value range
Sim Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. ▪ Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA Out (IOCNET) settings

The following dialog settings are specific for the **I-FPGA Out** I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The **Sim Data** inport is added to the block.

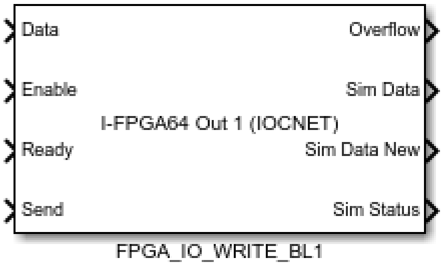
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the **Sim Data** inport.

The **Sim Data New** outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The **Sim Status** outport is added to the block.

I-FPGA64 Out (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit raw data value to be written to an IOCNET buffer. Data type: UFix_64_0 Data width: 1
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No overflow occurred.

Port	Description
	- 1: An overflow occurred. This value is set for one clock cycle.
Sim_Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 512, depends on the **Buffer size** parameter. Range: Double precision value range
Sim_Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 - Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. - Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. - Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.

The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 Out (IOCNET) settings

The following dialog settings are specific for the I-FPGA Out I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 ... 512.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The Sim Data inport is added to the block.

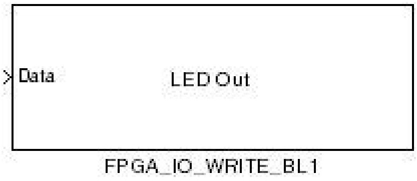
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the Sim Data inport.

The Sim Data New outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The Sim Status outport is added to the block.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the LED on the DS6601 FPGA Base Board. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: LED lights green. ▪ 1: LED lights orange. If the value of the Data port exceeds the specified data width, only the lowest bit is used (=1).

I/O mapping No external connection.

LED Out settings

None

Related topics

References

Description Page (FPGA_IO_WRITE_BL)....................215
FPGA_IO_WRITE_BL....................56
FPGA_IO_WRITE_BL (DS6601 FPGA Base Board Settings)....................205

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the

text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6601 (KU035) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL..56
FPGA_IO_WRITE_BL (DS6601 FPGA Base Board Settings)..205
Parameters Page (FPGA_IO_WRITE_BL)..206

FPGA_INT_BL (DS6601 FPGA Base Board Settings)

Purpose

To configure the FPGA interrupt channel when using the *DS6601 (KU035) FPGA Base Board* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL)..217
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..218
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL)..66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

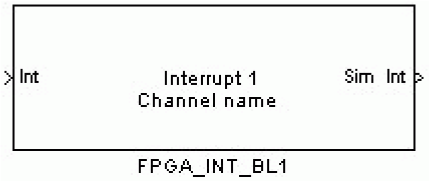
Description

The DS6601 FPGA Base Board provides 16 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The Sim_Int outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL) 218
FPGA_INT_BL 62
FPGA_INT_BL (DS6601 FPGA Base Board Settings) 216

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6601 (KU035) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

RTI Block Settings for the DS6602 FPGA Base Board Framework

Introduction

The block dialogs provide hardware-specific settings after you load the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

FPGA_XDATA_READ_BL (DS6602 FPGA Base Board Settings)

Purpose

To configure read access to IOCNET data in the FPGA model when using the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS6602 (KU15P) FPGA Base Board* framework provides the following access types that you can select on the **Unit** page of the block dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element.

  If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

  Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Because register groups can be accessed only by one task, you have to explicitly group registers which are used by different tasks.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from an IOCNET buffer. 32 buffers are available that provide elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.

- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

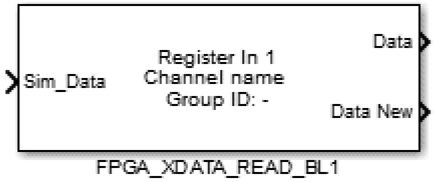
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits inheriting the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Port	Description
Output	
Data	Outputs a 32-bit data value to be read from a IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 within only one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

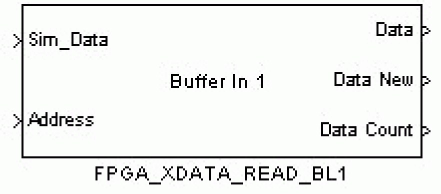
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 221.

Register group ID Lets you specify a number in the range 1 ... 63 to create register groups. Registers that you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element you want to read in the buffer. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 221.

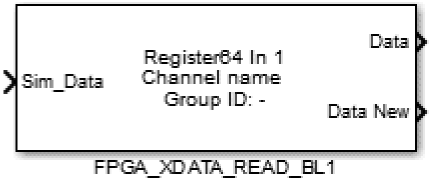
Buffer size Lets you specify the size of the buffer in the range 1 ... 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET register. The data format depends on the related dialog settings.

Port	Description
	Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 within only one clock cycle. Data type: UFix_1_0

> **Note**
>
> The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 221.

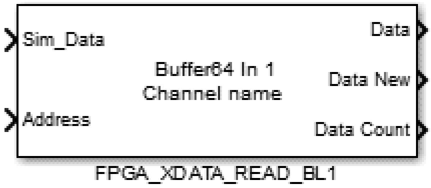
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the IOCNET sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

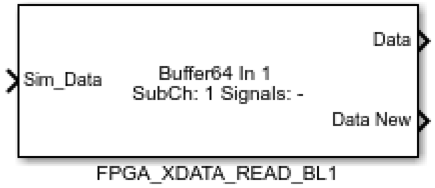
Buffer64 In description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an IOCNET data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is copied via **Copy bus topology from gcb** on the **Parameters** page. The maximum bit wide is 64 bits. The resolution of the data types is restricted to 53 bits, because the data type of the received bus signals from the processor

Port	Description
	application is double and the block converts the signals to the signal data types.
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 221.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Copy bus topology from gcb Lets you copy an existing FPGA bus topology from the selected Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** block.

You cannot copy a bus topology from the processor model, because these topologies do not include the FPGA data types. For instructions, refer to How to

Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port. This button is clickable only if the bus transfer mode is enabled.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)........229
FPGA_XDATA_READ_BL........40
FPGA_XDATA_READ_BL (DS6602 FPGA Base Board Settings)........220

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6602 (KU15P) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_READ_BL........40
FPGA_XDATA_READ_BL (DS6602 FPGA Base Board Settings)........220
Parameters Page (FPGA_XDATA_READ_BL)........221

FPGA_XDATA_WRITE_BL (DS6602 FPGA Base Board Settings)

Purpose

To configure write access to IOCNET data in the FPGA model when using the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The *DS6602 (KU15P) FPGA Base Board* framework provides the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an IOCNET register. 256 registers are available with a data width of 32 bits each and 256 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an IOCNET buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For more information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

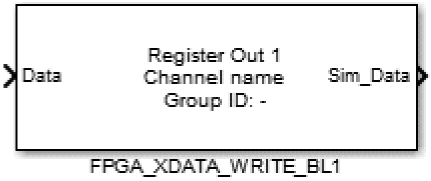
The fraction width is displayed in the Binary point position (or fraction width) setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The Sim_Data outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data and Sim_Status. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

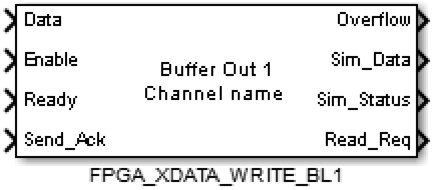
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 231.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers that you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are

Port	Description
	therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred.

Port	Description
	- 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 - Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. - Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. - Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 - 0: No data transmission is requested. - 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 231.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

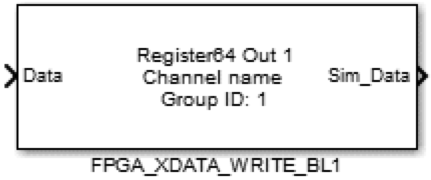
> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the **Read_Req** and **Send_Ack** ports to the block to trigger a processor synchronous data exchange.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion.

Port	Description
	Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

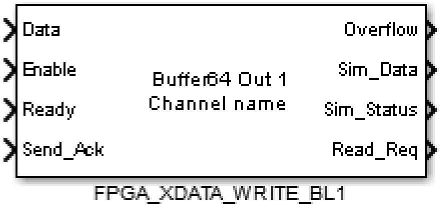
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 231.

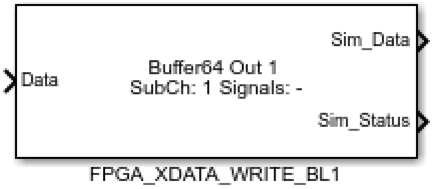
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form a consistent data group that is written to the IOCNET.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an IOCNET buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is set via **Analyze bus topology of input** on the **Parameters** page. The maximum bit width is 64 bits. The resolution of the data type is restricted to 53 bits, because the block converts all data values to double for transmission.
Enable	Specifies the current valid **Data** port value. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock

Port	Description
	cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via IOCNET in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send_Ack	Triggers a data transmission to IOCNET. With **Send_Ack** and **Read_Req** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged.
Output	
Sim_Data	Simulates an IOCNET data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size or the number of bus signals.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read.

Port	Description
	Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.
Read_Req	Outputs a flag that indicates that a data transmission is requested via IOCNET. With **Read_Req** and **Send_Ack** you can trigger a processor synchronous data exchange. A data transmission is always requested at the beginning of a task, before the processor model is computed. Each time a new data transmission is requested by **Read_Req**, **Send_Ack** must explicitly acknowledge the **Data** values for transmission within one task period. To send current data you can delay the transmission. After a new data transmission is requested, you

Port	Description
	write the current data values to the buffer. Then you must acknowledge the new data for transmission. Available only if **Enable Read_Req and Send_Ack ports for explicit data transmit** is set on the **Parameters** page. If **Enable Read_Req and Send_Ack ports for explicit data transmit** is not set, each data transmission request will instantly be acknowledged. Data type: UFix_1_0 ▪ 0: No data transmission is requested. ▪ 1: A data transmission is requested. This value is set for one clock cycle. A data transmission request that is not acknowledged by **Send_Ack** leads to task overrun in the processor application. A task overrun will be logged as an I/O error in the **Message Viewer** of the SCALEXIO web interface. The FPGA buffer that caused the task overrun will also be logged. **Note** In the bus transfer mode, a data transmission request automatically triggers a data transmission.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 231.

Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable Read_Req and Send_Ack ports for explicit data transmit Lets you add the **Read_Req** and **Send_Ack** ports to the block to trigger a processor synchronous data exchange.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the following parameters are not configurable:

- Binary point position (or fraction width)
- Format
- Buffer size
- Enable Read_Req and Send_Ack ports for explicit data transmit

Analyze bus topology of input Lets you set the Data inport to the bus topology of the connected Simulink bus.

If clicked, the RTI FPGA Programming Blockset analyzes the connected Simulink bus and sets the Data port to a matching bus topology. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the Data port.

This button is clickable only if the bus transfer mode is enabled.

Bus data transmission method Lets you select the method for transmitting data to the processor application if the bus transfer mode is enabled:

- Synchronous to Read_Req method

  Select this method to transmit data that is captured synchronously to the processor task.

  The FPGA application writes data to the swinging buffer when the processor application makes a read request. After the data is written to the buffer, the buffer swings and sends the data to the processor application.
- Free running method

  Select this method if the transmission time is crucial.

  The FPGA application continuously writes data to the swinging buffer. A read request of the processor application immediately transmits the last complete data set of the swinging buffer to the processor application.

The bus data transmission method is selectable only for subchannel 1 and the bus transfer mode must be enabled. The selection applies to all subchannels of the selected channel.

For instructions, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 🕮).

Related topics

HowTos

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_XDATA_WRITE_BL)........243
FPGA_XDATA_WRITE_BL........45
FPGA_XDATA_WRITE_BL (DS6602 FPGA Base Board Settings)........230

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6602 (KU15P) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_XDATA_WRITE_BL.. 45
FPGA_XDATA_WRITE_BL (DS6602 FPGA Base Board Settings).. 230
Parameters Page (FPGA_XDATA_WRITE_BL).. 230

FPGA_IO_READ_BL (DS6602 FPGA Base Board Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)........54
To implement read access to a physical input channel in the FPGA model.

Unit Page (FPGA_IO_READ_BL)........55
To specify the I/O type and channel to be used for read access.

Related RTI blocks

FPGA_IO_WRITE_BL........56
To provide write access to an external device via a physical output channel.

FPGA_IO_WRITE_BL (DS6602 FPGA Base Board Settings)........255
To configure write access to analog and digital output signals in the FPGA model when using the DS6602 (KU15P) FPGA Base Board framework.

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Others*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- APU Slave 1 … 6
- CN APP Status
- I-FPGA In 1 (IOCNET) ... I-FPGA In 32 (IOCNET)
- I-FPGA64 In 1 (IOCNET) ... I-FPGA64 In 32 (IOCNET)
- IOCNET Global Time
- Status In
- Temperature

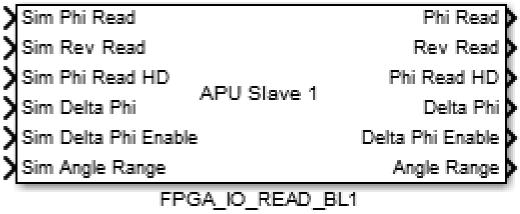
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to simulation data coming from a Simulink-based I/O environment model.

APU Slave 1 ... 6 description

Block display If you select the APU Slave channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the port of the block for digital input channels:

Port	Description
Input	
Sim Phi Read	Simulates the APU bus value for Phi Read. Available only if **Enable simulation Phi Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range depends on the angle range of the simulated APU bus: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns The range can be exceeded, and saturation is performed to a minimum or maximum value.
Sim Rev Read	Simulates the hardware input value for Rev Read. Available only if **Enable simulation Rev Read port** is set on the **Parameters** page. Data type: Double Data width: 1 The 37 Bit range is -2^{36} ... $+2^{36}$ - 1. APU bus clock cycle: 8 ns Range exceeding is not possible.
Sim Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Sim Angle Range	Simulates the angle range of the APU bus. Available only if **Enable simulation Angle Range port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range

Port	Description
Output	
Phi Read	Outputs the angle value that APU Slave reads from the APU bus. The angle value is independent from the angle range of the APU bus. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the angle range of the APU bus: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value for the APU bus. Data type: Double Data width: 1 The 37 Bit range is -2^{36} ... 2^{36} - 1. The APU bus clock cycle is 8 ns. Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle rangeThe angle range is 0° ... 720° ▪ 1 (high): 360° angle range

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Slave 1 ... 6 settings

The Parameters page provide the following dialog setting:

Enable Phi Read HD port The Phi Read HD port is available in the block only if you enable it.

Enable Delta Phi port The Delta Phi port is available in the block only if you enable it.

Enable Delta Phi Enable port The Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Phi Read port The Sim Phi Read port is available in the block only if you enable it.

Enable simulation Rev Read port The Sim Rev Read port is available in the block only if you enable it.

Enable simulation Phi Read HD port The Sim Phi Read HD port is available in the block only if you enable it.

Enable simulation Delta Phi port The Sim Delta Phi port is available in the block only if you enable it.

Enable simulation Delta Phi Enable port The Sim Delta Phi Enable port is available in the block only if you enable it.

Enable simulation Angle Range port The Sim Angle Range port is available in the block only if you enable it.

Angle range For the slave APU, you can inherit the angle range of the APU bus or you specify a local angle range independent from the APU bus. The following table shows you the possible combinations of angle range settings.

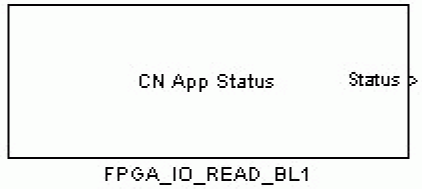
APU Bus Setting	Slave APU Setting	Resulting Angle Range of the Slave APU
360°	Inherit	360°
720°		720°
360°	360°	360°
	720°	720° [1]
720°	360°	360°[2]
	720°	720°

[1] Two engine cycles are required to run through the 720° angle range. If you simulate a four-stroke piston engine, for example, the angle-values of the function block are not clearly related to the camshaft position.

[2] One engine cycle runs twice through the 360° angle range.

CN App Status description

Block display If you select the CN App Status channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Status	Outputs the state of the application that is running on the computation node. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The application on the computation node is stopped.

Port	Description
	▪ 1: The application on the computation node is running.

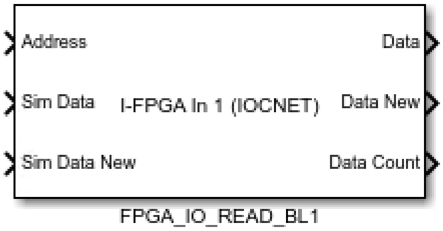
I/O mapping No external connection.

CN App Status settings

None

I-FPGA In (IOCNET) description

Block display If you select the I-FPGA In (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the **Data** outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the **Buffer size** on the **Parameters** page. The address range with valid data values can be derived from the value of the **Data Count** port.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the **Data** port and written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024 Range: Single precision value range
Sim Data New	Simulates the reception of new data values. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated. ▪ 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.

Port	Description
Output	
Data	Outputs a 32-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_32_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1 The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA In (IOCNET) settings

The following dialog settings are specific for the **I-FPGA In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 244.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

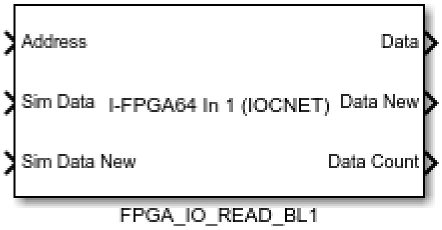
The **Sim Data** inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the **Data** outport.

The **Sim Data New** inport is added to the block.

I-FPGA64 In (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Address	Specifies a data value in the IOCNET buffer to be read. The block requires one clock cycle to update the value of the **Data** outport with the data value of the specified address. Data type: UFix_16_0 Data width: 1 The maximum address range depends on the **Buffer size** on the **Parameters** page. The address range with valid data values can be derived from the value of the **Data Count** port.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The provided data values are converted to the data format of the **Data** port and written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 512 Range: Double precision value range
Sim Data New	Simulates the reception of new data values. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 Values: ▪ 0: The reception of new data values is not simulated.

Port	Description
	- 1: The **Data New** port changes from 0 to 1 for one clock cycle to indicate new data values.
Output	
Data	Outputs a 64-bit raw data value from the specified address of the IOCNET buffer. Data type: UFix_64_0 Data width: 1
Data New	Outputs a flag that indicates the update of the **Data** port. Data type: UFix_1_0 Data width: 1 If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 within only one clock cycle.
Data Count	Outputs the number of elements in the current IOCNET buffer. You can use the value to define the valid range for the **Address** port from 0 to (Data Count -1). Data type: UFix_16_0 Data width: 1 The maximum value range depends on the **Buffer size** on the **Parameters** page.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.

The range can be exceeded for the **Data** outport. The outport's value is then cast to UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 In (IOCNET) settings The following dialog settings are specific for the **I-FPGA64 In (IOCNET)** I/O function. For common dialog settings, refer to Common settings on page 244.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 512. The maximum range of the Address inport depends on the buffer size.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an inport for offline simulation data.

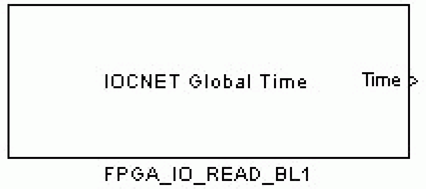
The Sim Data inport is added to the block.

Enable simulation data new port Lets you enable an inport for the offline simulating of new data values at the Data outport.

The Sim Data New inport is added to the block.

IOCNET Global Time description

Block display If you select the IOCNET Global Time channel from the channel list, the block display changes.



I/O characteristics The following table describes the port of the block:

Port	Description
Output	
Time	Outputs the number of hardware ticks that occurred since the SCALEXIO system power was switched to on. If you use a multiprocessor system, the value is set to zero each time an application is reloaded and restarted. Data type: 56 bit Tick step-width: 8.5 ns

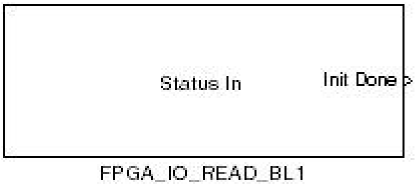
I/O mapping No external connection.

IOCNET Global Time settings

None

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

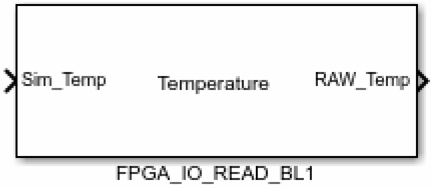
I/O mapping No external connection.

Status In settings

None

Temperature description

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C

Port	Description
	The range can be exceeded. The values are then saturated to the minimum or maximum values.
Output	

RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature.
	Equation for calculating the die temperature:
	`Temperature [°C] = (float)(Temperature[hex] & 0xFFF0)`
	`· 503.975 / 65536 - 273.15`
	Data type: UFix_16_0
	Data width: 1
	Value range: 0 ... 65536

I/O mapping No external connection.

Temperature settings

Only common dialog settings. Refer to Common settings on page 244.

Related topics

References

Description Page (FPGA_IO_READ_BL) 255
FPGA_IO_READ_BL 50
FPGA_IO_READ_BL (DS6602 FPGA Base Board Settings) 243

Scaling Page (FPGA_IO_READ_BL)

Description

The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6602 (KU15P) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

Description Page (FPGA_IO_READ_BL)............255
FPGA_IO_READ_BL............50
Parameters Page (FPGA_IO_READ_BL)............244

FPGA_IO_WRITE_BL (DS6602 FPGA Base Board Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model when using the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *APU*, *IFPGA32*, *IFPGA64*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

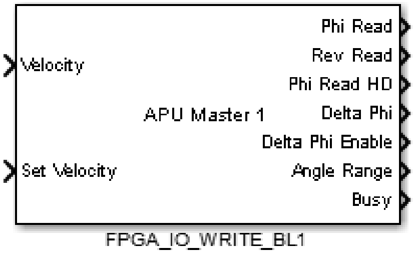
- APU Master 1 … APU Master 6
- DDR4 32 Mode 1
- DDR4 64 Mode 1
- DDR4 32 Mode 2
- DDR4 64 Mode 2
- I-FPGA Out (IOCNET)
- I-FPGA64 Out (IOCNET)
- LED Out

Common settings

None

APU Master 1 … 6 description

Block display If you select the APU Master channel from the channel list, the block display changes. For example:



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Velocity	Specifies a velocity value in degree/second to be applied as APU Master speed. The value will be applied if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: Fix_32_10 Data width: 1 Range: -1,200,000 °/s ... +1,200,000 °/s Range exceeding is not possible. The port is saturated at the higher or lower limit.
Set Velocity	Specifies the current value of **Velocity** as new velocity value. The new value is set only if **Set Velocity** is 1 (high) and **Busy** is 0 (low). Data type: UFix_1_0 Data width: 1 To distribute and execute a new velocity value takes at least 10 µs, depending on the IOCNET structure.
Output	
Phi Read	Outputs the angle counter value of the APU that the **APU Master** writes to the APU bus. The step size of the angle counter is approximately 0.011°. The step size is independent from the angle range. Formula for angle calculation: `alpha[°] = Phi Read * 720°/2`^{16} Data type: UFix_16_0 Data width: 1 The value range depends on the **Angle range** on the **Parameters** page: ▪ 360° angle range: 0 ... 32767 (2^{15}-1) ▪ 720° angle range: 0 ... 65535 (2^{16}-1) APU bus clock cycle: 8 ns Range exceeding is not possible.
Rev Read	Outputs the 37 bit total revolution (rev) value that **APU Master** reads from the APU bus. Data type: Fix_37_0

Port	Description
	Data width: 1 The 37 Bit range is -2^{36} ... 2^{36} - 1. APU bus clock cycle: 8 ns Range exceeding is not possible.
Phi Read HD	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Delta Phi Enable	Internal port - only for use in the dSPACE Electric Motor HIL Solution.
Angle Range	Outputs the angle range of the APU bus. You set the angle range on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0 (low): 720° angle range ▪ 1 (high): 360° angle range
Busy	Outputs whether **APU Master** is busy to set the last velocity value. Data type: UFix_1_0 Data width: 1 If the **Busy** port is 1 (high), new velocity values cannot be set. The **Busy** port stays active for at least 10 µs depending on the IOCNET structure.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

APU Master 1 … 6 settings

The **Parameters** page provides the following dialog setting:

Angle range Lets you select the angle range of the APU in degree.

- 360: The angle range is 360° and cannot be changed in ConfigurationDesk.
- 720: The angle range is 720° and cannot be changed in ConfigurationDesk.
- Individual: The **Angle range** property of the FPGA custom function block in ConfigurationDesk lets you set the angle range of the APU. The default value is 720°.

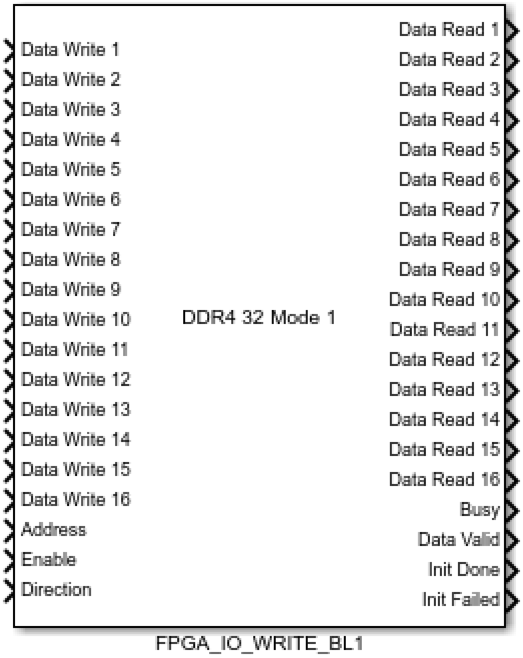
Initial position Lets you set the initial APU master position in degree.

- Value range: -1440° ... +1440°

Enable advanced ports The internal ports for the dSPACE Electric Motor HIL Solution are available in the block only if you enable it.

DDR 4 32 Mode 1 description

Block display If you select the DDR 4 32 Mode 1 channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	**Description**
Input	
Data Write 1 … 16	Specifies a 32-bit data value to be written to the RAM. 16 ports specify 16 data values. The Address port specifies the memory address to write the data values. Data type: UFix_32_0 Data width: 1 per port
Address	Specifies the first element in the RAM for the read/write access. The memory is addressed 512 bit-wise to read/write 16 x 32-bit data values with the same address. Data type: UFix_26_0 Data width: 1 Range: 0 … 67,108,863 (2^{26}-1)
Enable	Enables the RAM access. Data type: UFix_1_0 Data width: 1 Values: ▪ 1: Data values are written to the RAM or read from the RAM. ▪ 0: No read/write access.
Direction	Controls the direction of data access. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Write access

Port	Description
	▪ 1: Read access
Output	
Data Read 1 … 16	Outputs a 32-bit data value to be read from the RAM. 16 ports output 16 data values. The **Address** port specifies the memory address to read the data values. Data type: UFix_32_0 Data width: 1 per port
Busy	Outputs a flag that indicates the state of the DDR4 RAM. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The DDR4 module is ready for new read/write operations. ▪ 1: The DDR4 module is busy.
Data valid	Outputs a flag that indicates that the data values of the **Data Read** ports are valid. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be read by the FPAG application within the same clock cycle.
Init Done	Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Modeling DDR4 RAM Access (RTI FPGA Programming Blockset Guide 📖). Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized. ▪ 1: The RAM is initialized.
Init Failed	Outputs a flag that the initializing of the RAM with initial values failed. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized or no failure occurs. ▪ 1: A failure occurs during the initialization of the RAM.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

DDR 4 32 Mode 1 settings

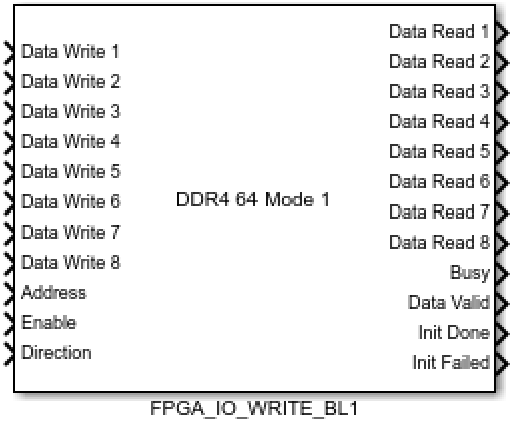
The **Parameters** page provides the following dialog setting:

Simulation init variable Lets you enter a workspace variable that provides values to initialize the RAM memory during offline simulation.

If the workspace variable does not provide a data value for each element of the RAM memory, the remaining elements are initialized with 0. For more information, refer to Initializing the DDR4 RAM of the DS6602 for Offline Simulation (RTI FPGA Programming Blockset Guide 🕮).

DDR 4 64 Mode 1 description

Block display If you select the DDR 4 64 Mode 1 channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Write 1 … 8	Specifies a 64-bit data value to be written to the RAM. 8 ports specify 8 data values at the same time. The Address port specifies the memory address to write the data values. Data type: UFix_64_0 Data width: 1 per port
Address	Specifies the first element in the RAM for the read/write access. The memory is addressed 512 bit-wise to read/write 8 x 64-bit data values with the same address. Data type: UFix_26_0 Data width: 1 Range: 0 … 67,108,863 (2^{26}-1)
Enable	Enables the RAM access. Data type: UFix_1_0 Data width: 1 Values: ▪ 1: Data values are written to the RAM or read from the RAM. ▪ 0: No read/write access.
Direction	Controls the direction of data access. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Write access ▪ 1: Read access

Port	Description
Output	
Data Read 1 … 16	Outputs a 32-bit data value to be read from the RAM. 8 ports output 8 data values at the same time. The **Address** port specifies the memory address to read the data values. Data type: UFix_64_0 Data width: 1 per port
Busy	Outputs a flag that indicates the state of the DDR4 RAM Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The DDR4 module is ready for new read/write operations. ▪ 1: The DDR4 module is busy.
Data valid	Outputs a flag that indicates that the data values of the **Data Read** ports are valid. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be read by the FPAG application within the same clock cycle.
Init Done	Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Modeling DDR4 RAM Access (RTI FPGA Programming Blockset Guide 🕮). Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized. ▪ 1: The RAM is initialized.
Init Failed	Outputs a flag that the initializing of the RAM with initial values failed. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized or no failure occurs. ▪ 1: A failure occurs during the initialization of the RAM.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

DDR 4 64 Mode 1 settings

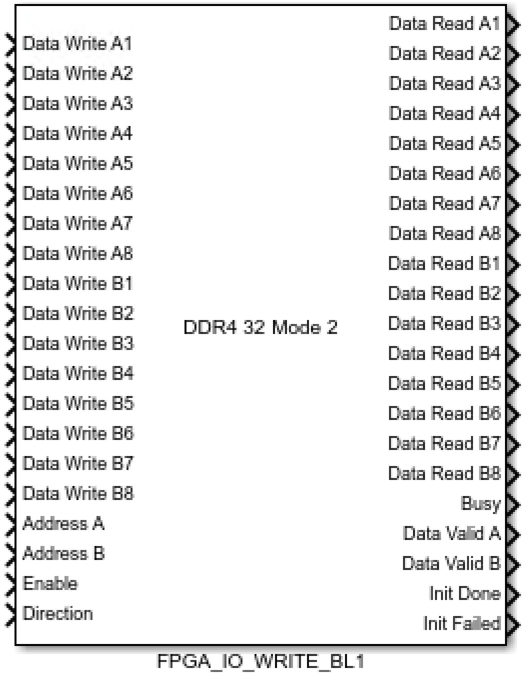
The **Parameters** page provides the following dialog setting:

Simulation init variable Lets you enter a workspace variable that provides values to initialize the RAM memory during offline simulation.

If the workspace variable does not provide a data value for each element of the RAM memory, the remaining elements are initialized with 0. For more information, refer to Initializing the DDR4 RAM of the DS6602 for Offline Simulation (RTI FPGA Programming Blockset Guide 🕮).

DDR 4 32 Mode 2 description

Block display If you select the DDR 4 32 Mode 2 channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Write A1 … A8	Specifies a 32-bit data value to be written to the RAM. 8 ports specify 8 data values at the same time. The **Address A** port specifies the memory address to write the data values. Data type: UFix_32_0 Data width: 1 per port
Data Write B1 … B8	Specifies a 32-bit data value to be written to the RAM. 8 ports specify 8 data values at the same time. The **Address B** port specifies the memory address to write the data values. Data type: UFix_32_0 Data width: 1 per port
Address A	Specifies the first element in the RAM for the read/write access of the **Data Write A**/**Data Read A** ports. The memory is addressed 256 bit-wise to read/write 8 x 32-bit data values with the same address. Data type: UFix_27_0 Data width: 1 Range: 0 … 134,217,727 (2^{27}-1)

Port	Description
Address B	Specifies the first element in the RAM for the read/write access of the **Data Write B**/**Data Read B** ports. The memory is addressed 256 bit-wise to read/write 8 x 32-bit data values with the same address. Data type: UFix_27_0 Data width: 1 Range: 0 … 134,217,727 (2^{27}-1)
Enable	Enables the RAM access. Data type: UFix_1_0 Data width: 1 Values: ▪ 1: Data values are written to the RAM or read from the RAM. ▪ 0: No read/write access.
Direction	Controls the direction of data access. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Write access ▪ 1: Read access
Output	
Data Read A1 … A8	Outputs a 32-bit data value to be read from the RAM. 8 ports output 8 data values at the same time. The **Address A** port specifies the memory address to read the data values. Data type: UFix_32_0 Data width: 1 per port
Data Read B1 … B8	Outputs a 32-bit data value to be read from the RAM. 8 ports output 8 data values at the same time. The **Address B** port specifies the memory address to read the data values. Data type: UFix_32_0 Data width: 1 per port
Busy	Outputs a flag that indicates the state of the DDR4 RAM. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The DDR4 module is ready for new read/write operations. ▪ 1: The DDR4 module is busy.
Data valid A	Outputs a flag that indicates that the data values of the **Data Read A** ports are valid Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be written within the same clock cycle.
Data valid B	Outputs a flag that indicates that the data values of the **Data Read B** ports are valid Data type: UFix_1_0

Port	Description
	Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be read by the FPAG application within the same clock cycle.
Init Done	Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Modeling DDR4 RAM Access (RTI FPGA Programming Blockset Guide 🕮). Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized. ▪ 1: The RAM is initialized.
Init Failed	Outputs a flag that the initializing of the RAM with initial values failed. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized or no failure occurs. ▪ 1: A failure occurs during the initialization of the RAM.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

DDR 4 32 Mode 2 settings

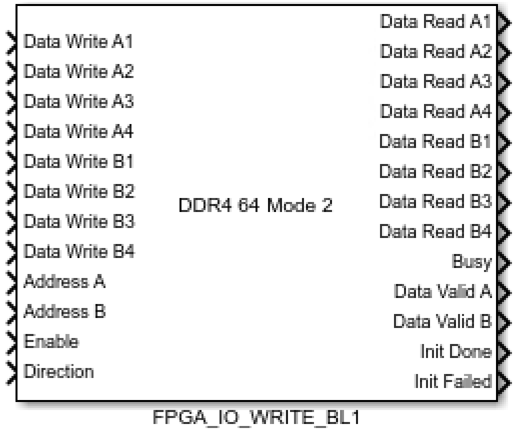
The Parameters page provides the following dialog setting:

Simulation init variable Lets you enter a workspace variable that provides values to initialize the RAM memory during offline simulation.

If the workspace variable does not provide a data value for each element of the RAM memory, the remaining elements are initialized with 0. For more information, refer to Initializing the DDR4 RAM of the DS6602 for Offline Simulation (RTI FPGA Programming Blockset Guide 🕮).

DDR 4 64 Mode 2 description

Block display If you select the DDR 4 32 Mode 2 channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Write A1 … A4	Specifies a 64-bit data value to be written to the RAM. 4 ports specify 4 data values at the same time. The **Address A** port specifies the memory address to write the data values. Data type: UFix_64_0 Data width: 1 per port
Data Write B1 … B4	Specifies a 64-bit data value to be written to the RAM. 4 ports specify 4 data values at the same time. The **Address B** port specifies the memory address to write the data values. Data type: UFix_64_0 Data width: 1 per port
Address A	Specifies the first element in the RAM for the read/write access of the **Data Write A**/**Data Read A** ports. The memory is addressed 256 bit-wise to read/write 4 x 64-bit data values with the same address. Data type: UFix_27_0 Data width: 1 Range: 0 … 134,217,727 (2^{27}-1)
Address B	Specifies the first element in the RAM for the read/write access of the **Data Write B**/**Data Read B** ports. The memory is addressed 256 bit-wise to read/write 4 x 64-bit data values with the same address. Data type: UFix_27_0 Data width: 1 Range: 0 … 134,217,727 (2^{27}-1)
Enable	Enables the RAM access. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 1: Data values are written to the RAM or read from the RAM. ▪ 0: No read/write access.
Direction	Controls the direction of data access. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Write access ▪ 1: Read access
Output	
Data Read A1 … A4	Outputs a 64-bit data value to be read from the RAM. 4 ports output 4 data values at the same time. The **Address A** port specifies the memory address to read the data values. Data type: UFix_64_0 Data width: 1 per port
Data Read B1 … B4	Outputs a 64-bit data value to be read from the RAM. 4 ports output 4 data values at the same time. The **Address B** port specifies the memory address to read the data values. Data type: UFix_64_0 Data width: 1 per port
Busy	Outputs a flag that indicates the state of the DDR4 RAM. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The DDR4 module is ready for new read/write operations. ▪ 1: The DDR4 module is busy.
Data valid A	Outputs a flag that indicates that the data values of the **Data Read A** ports are valid. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be read by the FPAG application within the same clock cycle.
Data valid B	Outputs a flag that indicates that the data values of the **Data Read B** ports are valid. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The values are not valid. ▪ 1: The values are valid. This value is set for one clock cycle. The data values must be read by the FPAG application within the same clock cycle.
Init Done	Outputs a flag that indicates that the RAM is initialized with specified data values. For more information, refer to Modeling DDR4 RAM Access (RTI FPGA Programming Blockset Guide 📖).

Port	Description
	Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized. ▪ 1: The RAM is initialized.
Init Failed	Outputs a flag that the initializing of the RAM with initial values failed. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The RAM is not initialized or no failure occurs. ▪ 1: A failure occurs during the initialization of the RAM.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping No external connection.

DDR 4 64 Mode 2 settings

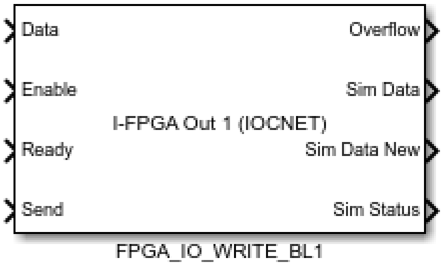
The Parameters page provides the following dialog setting:

Simulation init variable Lets you enter a workspace variable that provides values to initialize the RAM memory during offline simulation.

If the workspace variable does not provide a data value for each element of the RAM memory, the remaining elements are initialized with 0. For more information, refer to Initializing the DDR4 RAM of the DS6602 for Offline Simulation (RTI FPGA Programming Blockset Guide).

I-FPGA Out (IOCNET) description

Block display If you select the I-FPGA Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit raw data value to be written to an IOCNET buffer. Data type: UFix_32_0

Port	Description
	Data width: 1
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a

Port	Description
	simulated IOCNET buffer, including fixed-point to floating-point data conversion. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 1024, depends on the **Buffer size** parameter. Range: Single precision value range
Sim Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. ▪ Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 32 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_32_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.

The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_32_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA Out (IOCNET) settings The following dialog settings are specific for the **I-FPGA Out** I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 … 1024.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The Sim Data inport is added to the block.

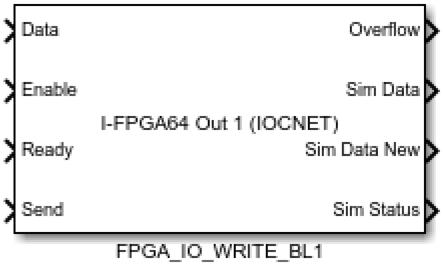
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the Sim Data inport.

The Sim Data New outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The Sim Status outport is added to the block.

I-FPGA64 Out (IOCNET) description

Block display If you select the I-FPGA64 Out (IOCNET) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit raw data value to be written to an IOCNET buffer. Data type: UFix_64_0 Data width: 1
Enable	Specifies the current valid Data port value. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: The **Data** value to be written is not stored in the IOCNET buffer. ▪ 1: The **Data** value to be written is stored in the IOCNET buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Send	Triggers a data transmission via IOCNET. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: **Data** values are not acknowledged for transmission. ▪ 1: Current **Data** values are acknowledged and will be transmitted via IOCNET.
Output	
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs when the new buffer is triggered for transmission and the old buffer was not sent completely. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Data	Simulates a data exchange between two FPGA boards via IOCNET. The port provides the data values that are written to a simulated IOCNET buffer. Available only if the **Enable simulation data port** is set on the **Parameters** page. Data type: Double Data width: 1 … 512, depends on the **Buffer size** parameter. Range: Double precision value range

Port	Description
Sim_Data New	Simulates the update of data values provided by the **Sim Data** outport. Available only if the **Enable simulation data new port** is set on the **Parameters** page. Data type: Double Data width: 1 A transition from 0 to 1 indicates that the **Sim Data** outport provides new data.
Sim Status	Outputs information about the simulated data exchange on the **Sim Data** outport. Available only if the **Enable simulation status port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim Status[0]: Contains the number of valid elements in the **Sim Data** vector. ▪ Sim Status[1]: Indicates whether the current buffer contains new or old values. The status is 1 if the buffer contains new values. ▪ Sim Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Note

You can transfer any data type with a bit width of up to 64 Bit via inter-FPGA over IOCNET. Use the **Reinterpret** Xilinx block to change your data type to UFix_64_0 and vice versa. Reinterpreting data types does not cost any hardware or latency.
The range of the **Data** inport can be exceeded. The value of the inport is then cast to the raw data format UFix_64_0.

Multiple clock domain support This block does not support multiple clock domains.

I-FPGA64 Out (IOCNET) settings

The following dialog settings are specific for the **I-FPGA Out** I/O function.

Buffer size Lets you specify the size of the IOCNET buffer in the range 1 ... 512.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation data port Lets you enable an outport for offline simulation data.

The Sim Data inport is added to the block.

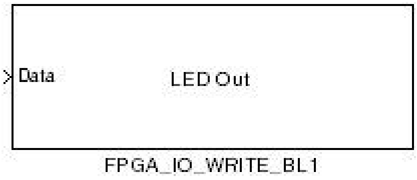
Enable simulation data new port Lets you enable an outport for the offline simulating of new data values at the Sim Data inport.

The Sim Data New outport is added to the block.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enabled a simulation port. The Sim Status outport is added to the block.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the LED on the DS6602 FPGA Base Board. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: LED is lit in green. ▪ 1: LED is lit in orange. If the value of the Data inport exceeds the specified data width, only the lowest bit is used (=1).

I/O mapping No external connection.

LED Out settings

None

Related topics

References

Description Page (FPGA_IO_WRITE_BL)....275
FPGA_IO_WRITE_BL....56
FPGA_IO_WRITE_BL (DS6602 FPGA Base Board Settings)....255

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6602 (KU15P) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (DS6602 FPGA Base Board Settings)... 255
Parameters Page (FPGA_IO_WRITE_BL).. 256

FPGA_INT_BL (DS6602 FPGA Base Board Settings)

Purpose

To configure the FPGA interrupt channel when using the *DS6602 (KU15P) FPGA Base Board* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL).. 277
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL).. 278
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL).. 66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

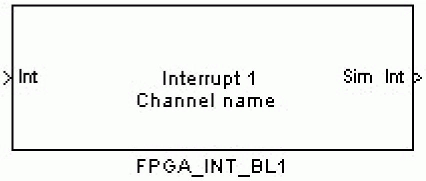
Description

The DS6602 FPGA Base Board provides 16 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The Sim_Int outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the *DS6602 (KU15P) FPGA Base Board* standard framework is included in this user documentation.

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

Description Page (FPGA_INT_BL)........278
FPGA_INT_BL........62
Parameters Page (FPGA_INT_BL)........277

RTI Block Settings for the DS2655M1 I/O Module Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks together with at least one *DS2655M1 I/O Module* framework:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework
- *DS6601 (KU035) FPGA Base Board* framework
- *DS6602 (KU15P) FPGA Base Board* framework

Where to go from here

Information in this section

Information in other sections

Information on selecting the framework to be used for each I/O module slot of a SCALEXIO FPGA base board:

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

FPGA_IO_READ_BL (DS2655M1 I/O Module Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using the *DS2655M1 I/O Module* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Digital_Mod_x* and *Analog_Mod_x*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available physical connections on the DS2655M1 I/O Module determines the I/O functions that you can select:

- Analog In - Ch: 11 … Analog In - Ch: 15
- Digital In - Ch: 1 … Digital In - Ch: 10

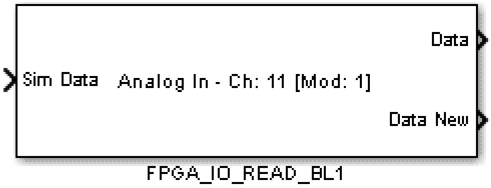
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim Data** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

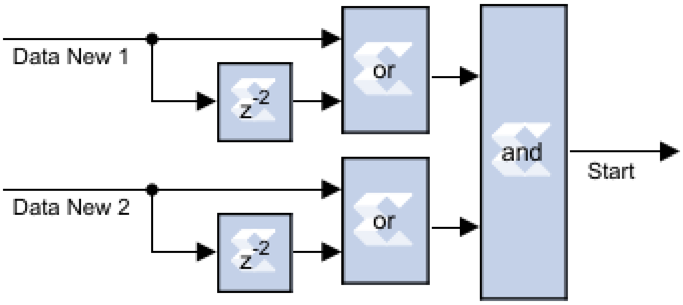
Analog In description

Block display If you select an Analog In channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable simulation port** is set on the **Parameters** page.

Port	Description
	Data type: Double Data width: 1 The value range corresponds to the settings of the **Input range** parameter in V. Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the current result of the analog input channel. Data type: Fix_16_0 [1] Data width: 1 Update rate: 2 Msps
Data New	Outputs a flag that indicates the current status of the **Data** port. The port is set to 1 for one clock cycle if the **Data** port provides new values. New measured values from analog input channels of the same I/O module are always provided synchronously. If analog inputs are read from different I/O modules, the measured values are provided either synchronously or offset by two clock cycles (16 ns). However, the sample time of the analog measurements is synchronous on different I/O modules except for 8 ns. If synchronous measured values from analog inputs of different I/O modules are required, you can implement a logic to wait with the further processing of analog values until the **Data New** ports flag new data within two clock cycles. The following example shows a logic for the **Data New** ports to use synchronous analog input signals of different I/O modules.  Data New 1 z^{-2} or and Start Data New 2 z^{-2} or Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No new values are available at the **Data** port. ▪ 1: New values are available at the **Data** port.

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).

Mapping for the **Data** output:

Input Voltage	Simulink Data Output
-5 V ... +5 V	The output port range depends on the setting of the **Scaling** parameter: ▪ -5000 ... +5000 mV ▪ -8192 ... +8191

Input Voltage	Simulink Data Output
-30 V ... +30 V	The output port range depends on the setting of the **Scaling** parameter: ▪ -30000 ... +30000 mV ▪ -8192 ... +8191

If the hardware signal or the value of the **Sim Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for analog input channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 I/O Module.

Outport	Channel	Connector Pin	Signal
Data	11	10	Analog In - Ch: 11 [Mod: x]
		26	Reference
	12	27	Analog In - Ch: 12 [Mod: x]
		43	Reference
	13	44	Analog In - Ch: 13 [Mod: x]
		11	Reference
	14	12	Analog In - Ch: 14 [Mod: x]
		28	Reference
	15	29	Analog In - Ch: 15 [Mod: x]
		45	Reference

Analog In settings

The following settings on the **Parameters** page are specific to the **Analog In** I/O function. For common dialog settings, refer to Common settings on page 281.

The **Parameters** page provide the following dialog setting:

Input range Lets you specify the input voltage range that can be converted from analog to digital for the chosen ADC channel. Input voltages outside the specified range are ignored.
This electrical interface setting can be changed in ConfigurationDesk.

Scaling Lets you select whether the I/O function scales the measuring results of the A/D converter to mV.

- mV

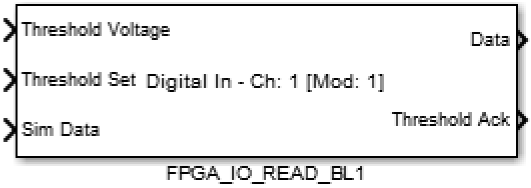
  To output the measuring results in mV. The valid value range corresponds to the settings of the **Input range** parameter in mV.
- Bit

  To output the raw measuring results as a signed Bit value.

  The value range is -8192 … +8191.

Digital In description

Block display If you select a Digital In channel from the channel list, the block display changes. The **Sim Data** port and the threshold voltage configuration ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Threshold Voltage	Specifies a trigger level in mV. A new threshold voltage level takes effect only after the **Threshold Set** port rises from 0 to 1. Available only if **Enable Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_14_0 Data width: 1 Range: 0 ... 10500 mV Update rate: FPGA clock frequency The range can be exceeded, and saturation is performed to a minimum or maximum value.
Threshold Set	Lets you set the trigger level as specified by the **Threshold Voltage** port. A new setting overwrites the settings of **Threshold init voltage** on the **Parameters** page. Available only if **Enable Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Range: ▪ No transition, or a 1 to 0 transition: A new voltage setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the output channel. The **Threshold Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 0 mV ... 10500 mV
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0

Port	Description
	Data width: 1 Values: ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency If the hardware signal or the value of the **Sim Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value. For information on the electrical characteristics of the DS2655M1 I/O Module, refer to Data Sheet of the DS2655M1 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Threshold Ack	Outputs a flag that indicates whether the threshold voltage level is up-to-date. Available only if **Enable Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The channel currently updates its threshold voltage. ▪ 1: The channel is up-to-date and a new configuration can be set.

> **Note**
>
> Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for digital input channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	2	Digital In - Ch: 1 [Mod: x]
		18	Reference
	2	19	Digital In - Ch: 2 [Mod: x]
		35	Reference
	3	36	Digital In - Ch: 3 [Mod: x]
		3	Reference
	4	4	Digital In - Ch: 4 [Mod: x]
		20	Reference
	5	21	Digital In - Ch: 5 [Mod: x]
		37	Reference
	6	6	Digital In - Ch: 6 [Mod: x]
		22	Reference
	7	23	Digital In - Ch: 7 [Mod: x]
		39	Reference
	8	40	Digital In - Ch: 8 [Mod: x]
		7	Reference
	9	8	Digital In - Ch: 9 [Mod: x]
		24	Reference
	10	25	Digital In - Ch: 10 [Mod: x]
		41	Reference

Digital In settings

The following settings on the Parameters page are specific to the Digital In I/O function. For common dialog settings, refer to Common settings on page 281.

Threshold init voltage Lets you specify the voltage value that is used for the threshold in mV. Range: 0 mV ... 10500 mV in 100 mV steps. This electrical interface setting can be changed in ConfigurationDesk.

Enable Threshold Voltage configuration ports Lets you enable ports to set the threshold voltage. The ports can overwrite the value of Threshold init voltage.

The following ports are added to the block:

- Threshold Voltage
- Threshold Set
- Threshold Ack

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital In settings

The following settings on the **Scaling** page are specific to the **Digital In** I/O function.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

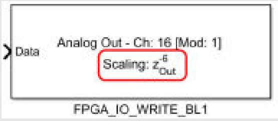
Analog In settings

The following settings on the **Scaling** page are specific to the **Analog In** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.

- 1 ... 20: The adding will be implemented with the specified latency.
  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide)

References

Description Page (FPGA_IO_READ_BL)....................290
FPGA_IO_READ_BL....................50
FPGA_IO_READ_BL (DS2655M1 I/O Module Settings)....................280
Parameters Page (FPGA_IO_READ_BL)....................281

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS2655M1 I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL....................50
FPGA_IO_READ_BL (DS2655M1 I/O Module Settings)....................280
Parameters Page (FPGA_IO_READ_BL)....................281
Scaling Page (FPGA_IO_READ_BL)....................287

FPGA_IO_WRITE_BL (DS2655M1 I/O Module Settings)

Purpose

To configure write access to analog and digital input signals in the FPGA model when using the *DS2655M1 I/O Module* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Digital_Mod_x* and *Analog_Mod_x*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available physical connections on the DS2655M1 Multi-I/O Module determines the I/O functions that you can select:

- Analog Out - Ch: 16 … Analog Out - Ch: 20
- Digital InOut - Ch: 1 … Digital InOut - Ch: 10
- Digital Out - Ch: 1 … Digital Out - Ch: 10

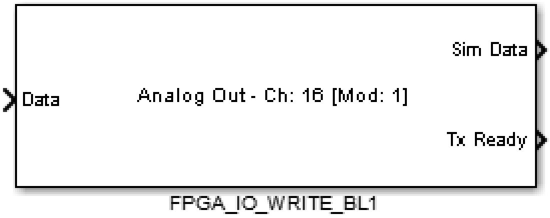
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL**.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

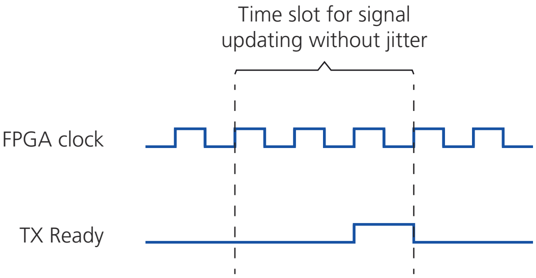
Analog Out description

Block display If you select an Analog Out channel from the channel list, the block display changes. The simulation port and the Tx Ready port are displayed optionally.



I/O characteristics The following table describes the ports of the block for analog output channels:

Port	Description
Input	
Data	Outputs a voltage signal in the specified range. Data type: Fix_15_0 [1)] Data width: 1 Mapping: see table below Update rate: 7.8125 MSPS
Output	
Sim Data	Analog output data signal. The output is fixed to -10 V … +10 V. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: ▪ -10000 … +10000 mV ▪ -8192 … +8191
TX Ready	Outputs a flag that indicates that the I/O channel of the DS2655M1 Multi-I/O Module is ready to be updated. The minimum update period is 64 ns. When you update data values only within the time slot for updating the output signal, the

Port	Description
	output signal has no jitter. The time slot begins two FPGA clock cycles before the flag is set to high and ends after three clock cycles.  Time slot for signal updating without jitter FPGA clock TX Ready **Note** **Parallel output without jitter** The **Analog Out** channels are synchronously updated if you do not use **Digital InOut** channels 5 ... 10. This lets you output parallel signals without jitter. **Analog Out** channels 16 ... 20 and **Digital InOut** channels 5 ... 10 share resources. If you use **Analog Out** channels, the **Analog Out** channels and the related **Digital InOut** channels use their own update process. This update process also lets you output parallel signals without jitter, but not synchronously to the other **Digital InOut** channels. For example, if you use **Digital InOut** channels 8 ... 10, **Analog Out** channels 16 ... 17 are synchronously updated. **Analog Out** channels 18 ... 20 are updated at another point in time, but these channels are also synchronously updated. Available only if **Enable Tx Ready port** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for analog output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	16	14	Analog Out - Ch: 16 [Mod: x]
		30	Reference
	17	31	Analog Out - Ch: 17 [Mod: x]
		47	Reference
	18	48	Analog Out - Ch: 18 [Mod: x]
		15	Reference
	19	16	Analog Out - Ch: 19 [Mod: x]
		32	Reference
	20	33	Analog Out - Ch: 20 [Mod: x]
		49	Reference

Analog Out settings

The following settings on the **Parameters** page are specific to the **Analog Out** I/O function. For common dialog settings, refer to Common settings on page 292.

Scaling Lets you select the scaling of the **Data** inport.

- mV

  To specify the output voltage in mV.

  The value range is -10,000 mV … +10,000 mV.
- Bit

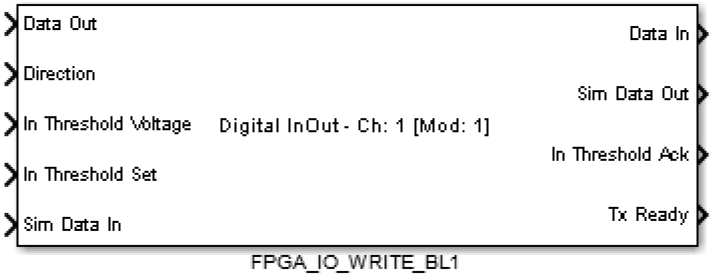
  To specify the output voltage with a signed bit value.

  The value range is -8,192 ... +8,191 (14-bit converter).

Enable Tx Ready port Lets you enable an outport to indicate that the analog output channel is ready to be updated. The **Tx Ready** port is added to the block.

Digital InOut description

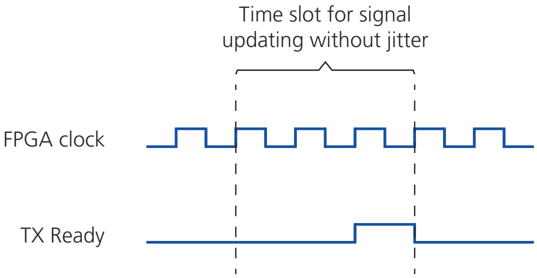
Block display If you select a Digital InOut channel from the channel list, the block display changes. The simulation ports, the threshold voltage configuration ports, and the **Tx Ready** port are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Out	Outputs a signal in the specified range. Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal at the I/O connector. If driven with 1, the hardware outputs a high-level signal. Update rate: 15.625 MHz For information on the electrical characteristics of the DS2655M1 Multi-I/O Module, refer to Data Sheet of the DS2655M1 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration).
Direction	Specifies the direction of the digital signal. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: digital in ▪ 1: digital out Update rate: 125 MHz
In Threshold Voltage	Specifies a trigger level in mV. A new threshold voltage level takes effect only after the **In Threshold Set** port rises from 0 to 1. Available only if **Enable In Threshold Voltage configuration ports** is set on the **Parameters** page. Data Type: UFix_14_0 Data width: 1 Range: 0 mV … +10500 mV in 100 mV steps Update rate: 125 MHz The range can be exceeded, and saturation is performed to a minimum or maximum value.
In Threshold Set	Lets you set the trigger level as specified by the **In Threshold Voltage** port. A new setting overwrites the settings of **Threshold init voltage** on the **Parameters** page. Available only if **Enable In Threshold Voltage ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ No transition, or a 1 to 0 transition: A new voltage setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the output channel. The **Threshold Ack** port outputs a flag if the channel is up-to-date. Update rate: 125 MHz Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data In	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable digital in simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 0 mV … 10500 mV
Output	
Data In	Outputs a signal that is 0 if the hardware input is driven with a voltage lower than the threshold voltage. Data Type: UFix_1_0 Data width: 1 For information on the electrical characteristics of the DS2655M1 Multi-I/O Module, refer to Data Sheet of the DS2655M1 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Sim Data Out	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable digital out simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Output voltage: 0 V … 5 V or 0 V … 3.3 V Update rate: 125 MHz
Threshold Ack	Outputs a flag that indicates changes of the threshold voltage level configuration. Available only if **Enable In Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Tx Ready	Outputs a flag that indicates that the I/O channel of the DS2655M1 Multi-I/O Module is ready to be updated. The minimum update period is 64 ns. When you update data values only within the time slot for updating the output signal, the output signal has no jitter. The time slot begins two

Port	Description
	FPGA clock cycles before the flag is set to high and ends after three clock cycles.  **Note** **Parallel output without jitter** The **Digital InOut** channels are synchronously updated if you do not use **Analog Out** channels. This lets you output parallel signals without jitter. **Digital InOut** channels 5 ... 10 and **Analog Out** channels 16 ... 20 share resources. If you use **Analog Out** channels, the **Analog Out** channels and the related **Digital InOut** channels use their own update process. This update process also lets you output parallel signals without jitter, but not synchronously to the other **Digital InOut** channels. For example, if you use **Analog Out** channels 18 ... 20, **Digital InOut** channels 1 ... 7 are synchronously updated. **Digital InOut** channels 8 ... 10 are updated at another point in time, but these channels are also synchronously updated. Available only if **Enable Tx Ready port** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1

Note

If you use a **Digital InOut** channel, the applicable threshold voltage for the digital input channel is less than or equal to the specified high supply.

To apply the maximum input voltage range, you have to use a **Digital In** channel.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for digital input/output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 Multi-I/O Module.

Port	Channel	Connector Pin	Signal
Data In and Data Out	1	2	Digital InOut - Ch: 1 [Mod: x]
		18	Reference
	2	19	Digital InOut - Ch: 2 [Mod: x]
		35	Reference
	3	36	Digital InOut - Ch: 3 [Mod: x]
		3	Reference
	4	4	Digital InOut - Ch: 4 [Mod: x]
		20	Reference
	5	21	Digital InOut - Ch: 5 [Mod: x]
		37	Reference
	6	6	Digital InOut - Ch: 6 [Mod: x]
		22	Reference
	7	23	Digital InOut - Ch: 7 [Mod: x]
		39	Reference
	8	40	Digital InOut - Ch: 8 [Mod: x]
		7	Reference
	9	8	Digital InOut - Ch: 9 [Mod: x]
		24	Reference
	10	25	Digital InOut - Ch: 10 [Mod: x]
		41	Reference

Digital InOut settings

The following settings on the Parameters page are specific to the Digital InOut I/O function.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Digital In threshold init voltage (0 mV ... +10500 mV) Lets you set the initial threshold voltage for a digital input signal in 100 mV steps. This electrical interface setting can be changed in ConfigurationDesk.

Enable In Threshold Voltage configuration ports Lets you enable ports to set the threshold voltage. The ports can overwrite the value of the Digital In threshold init voltage.

The following ports are added to the block:

- In Threshold Voltage
- In Threshold Set
- In Threshold Ack

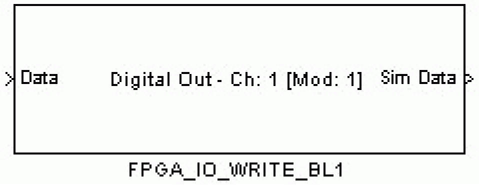
Enable digital out simulation port Lets you enable an outport for offline simulation data. The Sim Data Out outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable digital in simulation port Lets you enable an inport for offline simulation data. The Sim Data In inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model. This port is relevant only if the signal direction is in.

Enable Tx Ready port Lets you enable an outport to indicate that the analog output channel is ready to be updated. The Tx Ready port is added to the block.

Digital Out description

Block display If you select a Digital Out channel from the channel list, the block display changes. The Sim Data port is displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal at the I/O connector. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used. Note: The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS2655M1 Multi-I/O Module, refer to Data Sheet of the DS2655M1 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Output	
Sim Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data Type: Double Data width: 1 Output voltage: 0 V … 5 V or 0 V … 3.3 V Update rate: FPGA clock frequency

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M1 I/O Module* framework for digital output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M1 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	2	Digital Out - Ch: 1 [Mod: x]
		18	Reference
	2	19	Digital Out - Ch: 2 [Mod: x]
		35	Reference
	3	36	Digital Out - Ch: 3 [Mod: x]
		3	Reference
	4	4	Digital Out - Ch: 4 [Mod: x]
		20	Reference
	5	21	Digital Out - Ch: 5 [Mod: x]
		37	Reference
	6	6	Digital Out - Ch: 6 [Mod: x]
		22	Reference
	7	23	Digital Out - Ch: 7 [Mod: x]
		39	Reference
	8	40	Digital Out - Ch: 8 [Mod: x]
		7	Reference
	9	8	Digital Out - Ch: 9 [Mod: x]
		24	Reference
	10	25	Digital Out - Ch: 10 [Mod: x]
		41	Reference

Digital Out settings

The following settings on the **Parameters** page are specific to the **Digital Out** I/O function. For common dialog settings, refer to Common settings on page 292.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Description Page (FPGA_IO_WRITE_BL).. 306
FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (DS2655M1 I/O Module Settings).. 291
Scaling Page (FPGA_IO_WRITE_BL).. 302

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital Out settings

The following settings on the Scaling page are specific to the Digital Out I/O function.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- Enabled:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

Digital InOut settings

The following settings on the Scaling page are specific to the Digital InOut I/O function.

Invert input polarity Lets you invert the measured values of the electrical input signal:

- Disabled:

  The Data port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- Enabled:

  The output of the Data port is inverted: A low-high transition results in a 0 and vice versa.

Invert output polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- Enabled:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

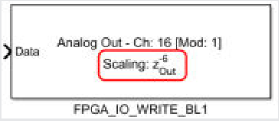
Analog Out settings

The following settings on the **Scaling** page are specific to the **Analog Out** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling factor** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling offset** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation minimum value Lets you specify the minimum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.

- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_IO_WRITE_BL)	306
FPGA_IO_WRITE_BL	56
FPGA_IO_WRITE_BL (DS2655M1 I/O Module Settings)	291
Parameters Page (FPGA_IO_WRITE_BL)	291

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS2655M1 I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL	56
FPGA_IO_WRITE_BL (DS2655M1 I/O Module Settings)	291
Parameters Page (FPGA_IO_WRITE_BL)	291
Scaling Page (FPGA_IO_WRITE_BL)	302

RTI Block Settings for the DS2655M2 I/O Module Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks together with at least one *DS2655M2 I/O Module* framework:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework
- *DS6601 (KU035) FPGA Base Board* framework
- *DS6602 (KU15P) FPGA Base Board* framework

Where to go from here

Information in this section

FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)............................308
To configure read access to the selected I/O function when using the DS2655M2 I/O Module framework.

FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)..........................320
To configure write access to analog and digital input signals in the FPGA model when using the DS2655M2 I/O Module framework.

Information in other sections

Information on selecting the framework to be used for each I/O module slot of a SCALEXIO FPGA base board:

Unit Page (FPGA_SETUP_BL)..29
To specify the FPGA framework.

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

RTI Block Settings for the DS2655 FPGA Base Board Framework............121
The block dialogs provide hardware-specific settings after you load one of the DS2655 FPGA Base Board frameworks.

RTI Block Settings for the DS6601 FPGA Base Board Framework............169
The block dialogs provide hardware-specific settings after you load the DS6601 (KU035) FPGA Base Board framework.

RTI Block Settings for the DS6602 FPGA Base Board Framework............219
The block dialogs provide hardware-specific settings after you load the DS6602 (KU15P) FPGA Base Board framework.

RTI Block Settings for the DS2655M1 I/O Module Framework................279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)

Purpose

To configure read access to the selected I/O function when using the *DS2655M2 I/O Module* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O type *Digital_Mod_<x>*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available channels on the selected DS2655M2 Digital I/O Module determines the I/O functions that you can select:

- Digital In - Ch: 1, … , Digital In - Ch: 32
- RS232 Rx - Ch: 2, … , RS232 Rx - Ch: 30
- RS485 Rx - Ch: 1-2, … , RS485 Rx - Ch: 29-30

Channel dependencies

The I/O functions of the *DS2655M2 I/O Module* framework share the I/O channels that provide the I/O functionality. The DS2655M2 Digital I/O Module provides 32 I/O channels. Some channels provide only specific I/O functionalities and some I/O functions use more than one I/O channel. These channel dependencies limit the number of available I/O functions.

For an overview of the DS2655M2 Digital I/O Module, refer to DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

For details on the signal mapping, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

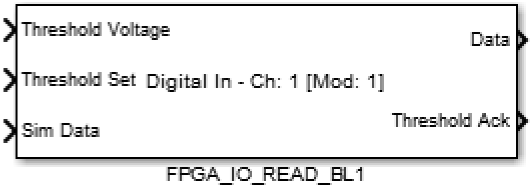
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim Data** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Digital In description

Block display If you select a Digital In channel from the channel list, the block display changes. Except for the outport **Data**, all the ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Threshold Voltage	Specifies a trigger level in mV. A new threshold voltage level takes effect only after the **Threshold Set** port rises from 0 to 1. Available only if **Enable Threshold Voltage ports** is set on the **Parameters** page. Data type: UFix_14_0 Data width: 1 Range: 0 mV ... 10500 mV Update rate: FPGA clock frequency, max 250 MHz. The range can be exceeded, and saturation is performed to a minimum or maximum value.
Threshold Set	Lets you set the trigger level as specified by the **Threshold Voltage** port. A new setting overwrites the settings of **Threshold init voltage** on the **Parameters** page. Available only if **Enable Threshold Voltage ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new voltage setting does not take effect. ▪ 0 to 1 transition: If the **Threshold Set** port rises from 0 to 1, the new settings are sent to the output channel. The **Threshold Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. The threshold voltage determines a logical 0 or 1 as output of the **Data** port. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V ... 10.5 V Range exceeding is possible and will be saturated to the minimum or maximum values.

Port	Description
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency, max 250 MHz For information on the electrical characteristics of the DS2655M2 Digital I/O Module, refer to Data Sheet of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Threshold Ack	Outputs a flag that acknowledges a change of the threshold voltage level configuration. Available only if **Enable Threshold Voltage ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for Digital In channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	18	Dig In (Ch. 1)
	2	2	Dig In (Ch. 2)
	3	35	Dig In (Ch. 3)
	4	19	Dig In (Ch. 4)
	5	20	Dig In (Ch. 5)
	6	4	Dig In (Ch. 6)
	7	37	Dig In (Ch. 7)
	8	21	Dig In (Ch. 8)
	9	22	Dig In (Ch. 9)
	10	6	Dig In (Ch. 10)
	11	39	Dig In (Ch. 11)
	12	23	Dig In (Ch. 12)
	13	24	Dig In (Ch. 13)
	14	8	Dig In (Ch. 14)
	15	41	Dig In (Ch. 15)
	16	25	Dig In (Ch. 16)
	17	26	Dig In (Ch. 17)
	18	10	Dig In (Ch. 18)
	19	43	Dig In (Ch. 19)
	20	27	Dig In (Ch. 20)
	21	28	Dig In (Ch. 21)
	22	12	Dig In (Ch. 22)
	23	45	Dig In (Ch. 23)
	24	29	Dig In (Ch. 24)
	25	30	Dig In (Ch. 25)
	26	14	Dig In (Ch. 26)
	27	47	Dig In (Ch. 27)
	28	31	Dig In (Ch. 28)
	29	32	Dig In (Ch. 29)
	30	16	Dig In (Ch. 30)
	31	49	Dig In (Ch. 31)
	32	33	Dig In (Ch. 32)

Digital In settings

The following settings on the **Parameters** page are specific to the **Digital In** I/O function. For common dialog settings, refer to Common settings on page 309.

Threshold init voltage Lets you specify the voltage value that is used for the threshold in mV. Range: 0 mV ... 10500 mV in 100 mV steps. This electrical interface setting can be changed in ConfigurationDesk.

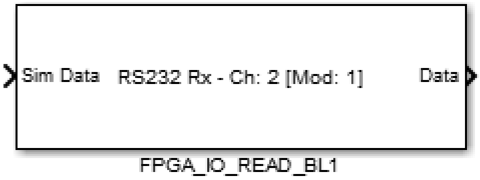
Enable Threshold Voltage configuration ports Lets you enable ports to set the threshold voltage. The ports can overwrite the value of Threshold init voltage.

The following ports are added to the block:

- Threshold Voltage
- Threshold Set
- Threshold Ack

RS232 Rx description

Block display If you select an RS232 Rx channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1 Range: –5.5 V ... +5.5 V Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the data received from the RS232 network. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The input voltage level is positive (≥ 0 V). ▪ 1: The input voltage level is negative (< 0 V). For information on the electrical characteristics of the DS2655M2 Digital I/O Module, refer to Data Sheet of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

Note

To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for RS232 Rx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Channel	Connector Pin	Signal
Data	2	2	RX (Ch. 2)
	6	4	RX (Ch. 6)
	10	6	RX (Ch. 10)
	14	8	RX (Ch. 14)
	18	10	RX (Ch. 18)
	22	12	RX (Ch. 22)
	26	14	RX (Ch. 26)
	30	16	RX (Ch. 30)

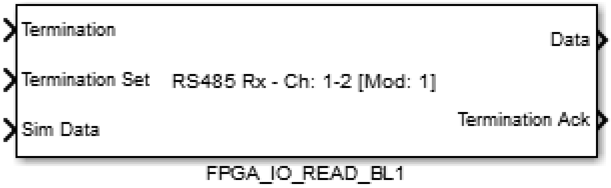
The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration).

RS232 Rx settings

Only common dialog settings. Refer to Common settings on page 309.

RS485 Rx description

Block display If you select an RS485 Rx channel from the channel list, the block display changes. Except for the **Data** outport, all the ports are displayed optionally.



Network mode With the RS485 Rx channel you can receive RS485 data in simplex mode. To connect to RS485 networks in full-duplex mode, use the corresponding RS485 Tx I/O function. For details on the RS485 Tx I/O function, refer to RS485 Tx description on page 338.

To connect to a RS485 network in half-duplex mode, use the RS485 Rx/Tx I/O function. For details, refer to RS485 Rx/Tx description on page 334.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Termination	Specifies whether the termination of the signal lines is enabled or disabled. A new termination configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No internal termination. ▪ 1: A 120 Ω resistor between the signal lines terminates the RS485 signal. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new termination setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: -5.5 V ... +5.5 V Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the data received from the RS485 network. Data type: UFix_1_0

Port	Description
	Data width: 1 Values: - 0: The input voltage level is negative (< 0 V). - 1: The input voltage level is positive (≥ 0 V).
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.

Note

To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 🕮).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for RS485 Rx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	18	Rx– (Ch. 1-2)
	2	2	Rx+ (Ch. 1-2)
	5	20	Rx– (Ch. 5-6)
	6	4	Rx+ (Ch. 5-6)
	9	22	Rx– (Ch. 9-10)
	10	6	Rx+ (Ch. 9-10)
	13	24	Rx– (Ch. 13-14)
	14	8	Rx+ (Ch. 13-14)
	17	26	Rx– (Ch. 17-18)
	18	10	Rx+ (Ch. 17-18)
	21	28	Rx– (Ch. 21-22)
	22	12	Rx+ (Ch. 21-22)
	25	30	Rx– (Ch. 25-26)
	26	14	Rx+ (Ch. 25-26)
	29	32	Rx– (Ch. 29-30)
	30	16	Rx+ (Ch. 29-30)

The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

RS485 Rx settings

The following settings on the **Parameters** page are specific to the **RS485 Rx** I/O function. For common dialog settings, refer to Common settings on page 309.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω resistor terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

- Termination
- Termination Set
- Termination Ack

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

Description Page (FPGA_IO_READ_BL)..319
FPGA_IO_READ_BL..50
FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)...308
Scaling Page (FPGA_IO_READ_BL)..318

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting settings for the selected I/O function.

Description

You can invert digital I/O signals and UART signals of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital In settings

The following settings on the **Scaling** page are specific to the **Digital In** I/O function.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

RS232 Rx settings

The following settings on the **Scaling** page are specific to the **RS232 Rx** I/O function.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:
  If the input voltage is negative (<0 V), the **Data** port outputs a 1.
  If the input voltage is positive (≥0 V), the **Data** port outputs a 0.
- **Enabled**:
  If the input voltage is negative (<0 V), the **Data** port outputs a 0.
  If the input voltage is positive (≥0 V), the **Data** port outputs a 1.

RS485 Rx settings

The following settings on the **Scaling** page are specific to the **RS485 Rx** I/O function.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:
  If the input voltage is negative (<0 V), the **Data** port outputs a 0.
  If the input voltage is positive (≥0 V), the **Data** port outputs a 1.
- **Enabled**:
  If the input voltage is negative (<0 V), the **Data** port outputs a 1.
  If the input voltage is positive (≥0 V), the **Data** port outputs a 0.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

References

Description Page (FPGA_IO_READ_BL)........319
FPGA_IO_READ_BL........50
FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)........308
Parameters Page (FPGA_IO_READ_BL)........309

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS2655M2 I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL..50
FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)...................308
Parameters Page (FPGA_IO_READ_BL)................................309
Scaling Page (FPGA_IO_READ_BL)...................................318

FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)

Purpose

To configure write access to analog and digital input signals in the FPGA model when using the *DS2655M2 I/O Module* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)...............................54
To implement read access to a physical input channel in the FPGA model.

Unit Page (FPGA_IO_READ_BL).......................................55
To specify the I/O type and channel to be used for read access.

Related RTI blocks

FPGA_IO_READ_BL...50
To provide read access to an external device via a physical input channel.

FPGA_IO_READ_BL (DS2655M2 I/O Module Settings)....................308
To configure read access to the selected I/O function when using the DS2655M2 I/O Module framework.

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O type *Digital_Mod_<x>*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available channels on the selected DS2655M2 Digital I/O Module determines the I/O functions that you can select:

- Digital Out - Ch: 1 … Digital Out - Ch: 32
- Digital Out-Z - Ch: 1-2 … Digital Out-Z - Ch: 31-32
- RS232 Tx - Ch: 1 ... RS232 Tx - Ch: 29
- RS485 Rx/Tx - Ch: 1-3 ... RS485 Rx/Tx - Ch: 29-31
- RS485 Tx - Ch: 1-2 ... RS485 Tx - Ch: 29-30

Channel dependencies

The I/O functions of the *DS2655M2 I/O Module* framework share the I/O channels that provide the I/O functionality. The DS2655M2 Digital I/O Module provides 32 I/O channels. Some channels provide only specific I/O functionalities and some I/O functions use more than one I/O channel. These channel dependencies limit the number of available I/O functions.

For an overview of the DS2655M2 Digital I/O Module, refer to DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 📖).

For details on the signal mapping, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 📖).

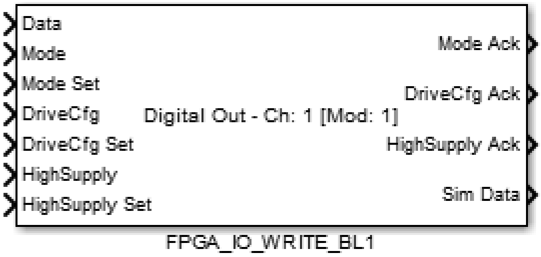
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the FPGA_IO_WRITE_BL.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

Digital Out description

Block display If you select an Digital Out channel from the channel list, the block display changes. Except for the inport **Data**, all the ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. To set the voltage level, use the **High supply** on the **Parameters** page or the **HighSupply Set** inport. Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS2655M2 Digital I/O Module, refer to Data Sheet of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).
Mode	Specifies the digital output mode. A new output mode takes effect only after the **Mode Set** port rises from 0 to 1. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data Type: UFix_3_0 Data width: 1 Values: ▪ 5: Enables the low-side switch that drives a load connected to VCC. ▪ 6: Enables the high-side switch that drives a load connected to GND. ▪ 7: Enables a push-pull switch that drives a load with VCC and GND.
Mode Set	Sets the output mode as specified by the **Mode** port. A new setting overwrites the settings of the **Output Mode** on the **Parameters** page. Available only if the **Enable Output Mode configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new mode does not take effect.

Port	Description
	▪ 0 to 1 transition: New settings are sent to the output channel. The **Mode Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg	Specifies the driver configuration. A new configuration takes effect only after the **DriveCfg Set** port rises from 0 to 1. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: An internal 68 Ω resistor to GND terminates the output signal. ▪ 1: The output signal is driven directly without a termination. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg Set	Sets the configuration as specified by the **DriveCfg** port. A new setting overwrites the settings of the **Drive Config** on the **Parameters** page. Available only if the **Enable Drive Config configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new configuration does not take effect. ▪ 0 to 1 transition: The output channel begins to update its channel. The **DriveCfg Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply	Specifies the supply voltage to output high-level signals. A new supply voltage takes effect only after the **HighSupply Set** port rises from 0 to 1. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The supply voltage for high signals is 5 V. ▪ 1: The supply voltage for high signals is 3.3 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply Set	Sets the supply voltage to the output high-level signal as specified by the **HighSupply** port. A new setting overwrites the setting of the **High Supply** parameter on the **Parameters** page. Available only if **Enable High Supply configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ No transition, or a 1 to 0 transition: A new supply voltage does not take effect. ▪ 0 to 1 transition: The new setting is sent to the output channel. The **HighSupply Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output Mode Ack	Outputs a flag that acknowledges a change of the output mode. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
DriveCfg Ack	Outputs a flag that acknowledges a change of the driver configuration. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
HighSupply Ack	Outputs a flag that acknowledges a change of the high supply configuration. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable simulation port** property is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V … 5 V or 0 V … 3.3 V Update rate: FPGA clock frequency

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for digital output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Dig Outport	Channel	Connector Pin	Signal
Data	1	18	Dig Out (Ch. 1)
	2	2	Dig Out (Ch. 2)
	3	35	Dig Out (Ch. 3)
	4	19	Dig Out (Ch. 4)
	5	20	Dig Out (Ch. 5)
	6	4	Dig Out (Ch. 6)
	7	37	Dig Out (Ch. 7)
	8	21	Dig Out (Ch. 8)
	9	22	Dig Out (Ch. 9)
	10	6	Dig Out (Ch. 10)
	11	39	Dig Out (Ch. 11)
	12	23	Dig Out (Ch. 12)
	13	24	Dig Out (Ch. 13)
	14	8	Dig Out (Ch. 14)
	15	41	Dig Out (Ch. 15)
	16	25	Dig Out (Ch. 16)
	17	26	Dig Out (Ch. 17)
	18	10	Dig Out (Ch. 18)
	19	43	Dig Out (Ch. 19)
	20	27	Dig Out (Ch. 20)
	21	28	Dig Out (Ch. 21)
	22	12	Dig Out (Ch. 22)
	23	45	Dig Out (Ch. 23)
	24	29	Dig Out (Ch. 24)
	25	30	Dig Out (Ch. 25)
	26	14	Dig Out (Ch. 26)
	27	47	Dig Out (Ch. 27)
	28	31	Dig Out (Ch. 28)
	29	32	Dig Out (Ch. 29)
	30	16	Dig Out (Ch. 30)
	31	49	Dig Out (Ch. 31)
	32	33	Dig Out (Ch. 32)

Digital Out settings

The following settings on the Parameters page are specific to the Digital Out I/O function. For common dialog settings, refer to Common settings on page 321.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Enable Output Mode configuration ports Lets you enable ports to set the digital output mode. The ports can overwrite the settings of the **Output Mode** parameter.

The following ports are added to the block:

- **Mode**
- **Mode Set**
- **Mode Ack**

Enable Drive Config configuration ports Lets you enable ports to terminate the digital output. The ports can overwrite the settings of the **Drive Config** parameter.

The following ports are added to the block:

- **DriveCfg**
- **DriveCfg Set**
- **DriveCfg Ack**

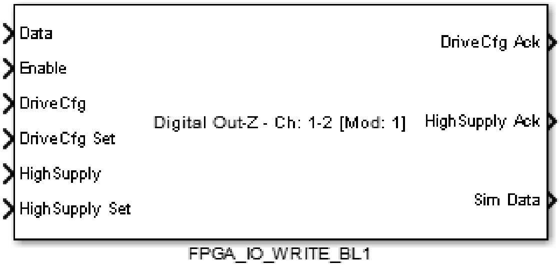
Enable High Supply configuration ports Lets you enable ports to set the voltage level of the high output signal. The ports can overwrite the settings of the **High supply** parameter.

The following ports are added to the block:

- **HighSupply Data**
- **HighSupply Set**
- **HighSupply Ack**

Digital Out-Z description

Block display If you select an Digital Out-Z channel from the channel list, the block display changes. Except for the **Data** and **Enable** inports, all the other ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range if the **Enable** port is set to 1. To set the voltage level, use the **High supply** parameter on the **Parameters** page or the **High Supply** inport . Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS2655M2 Digital I/O Module, refer to Data Sheet of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Enable	Enables the output of data values and disables the high-impedance state. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is set to the high-impedance state. ▪ 1: The output is enabled and outputs the data values of the **Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg	Specifies the driver configuration. A new configuration takes effect only after the **DriveCfg Set** port rises from 0 to 1. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: An internal 68 Ω resistor to GND terminates the output signal. ▪ 1: The output signal is driven directly without a termination. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg Set	Sets the configuration as specified by the **DriveCfg** port. A new setting overwrites the settings of the **Drive Config** on the **Parameters** page. Available only if the **Enable Drive Config configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new configuration does not take effect. ▪ 0 to 1 transition: The output channel begins to update its channel. The **DriveCfg Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply	Specifies the supply voltage to output high-level signals. A new supply voltage takes effect only after the **HighSupply Set** port rises from 0 to 1. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The supply voltage for high signals is 5 V. ▪ 1: The supply voltage for high signals is 3.3 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply Set	Sets the supply voltage to the output high-level signal as specified by the **HighSupply** port. A new setting overwrites the setting of the **High Supply** parameter on the **Parameters** page. Available only if **Enable High Supply configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new supply voltage does not take effect. ▪ 0 to 1 transition: The new setting is sent to the output channel. The **HighSupply Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
DriveCfg Ack	Outputs a flag that acknowledges a change of the driver configuration. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
HighSupply Ack	Outputs a flag that acknowledges a change of the high supply configuration. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable simulation port** property is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V … 5 V or 0 V … 3.3 V Update rate: FPGA clock frequency

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for digital output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Inport	Channel	Connector Pin	Signal
Data	1	18	Dig Out-Z (Ch. 1-2)
	2	2	No signal
	3	35	Dig Out-Z (Ch. 3-4)
	4	19	No signal
	5	20	Dig Out-Z (Ch. 5-6)
	6	4	No signal
	7	37	Dig Out-Z (Ch. 7-8)
	8	21	No signal
	9	22	Dig Out-Z (Ch. 9-10)
	10	6	No signal
	11	39	Dig Out-Z (Ch. 11-12)
	12	23	No signal
	13	24	Dig Out-Z (Ch. 13-14)
	14	8	No signal
	15	41	Dig Out-Z (Ch. 15-16)
	16	25	No signal
	17	26	Dig Out-Z (Ch. 17-18)
	18	10	No signal
	19	43	Dig Out-Z (Ch. 19-20)
	20	27	No signal
	21	28	Dig Out-Z (Ch. 21-22)
	22	12	No signal
	23	45	Dig Out-Z (Ch. 23-24)
	24	29	No signal
	25	30	Dig Out-Z (Ch. 25-26)
	26	14	No signal
	27	47	Dig Out-Z (Ch. 27-28)
	28	31	No signal
	29	32	Dig Out-Z (Ch. 29-30)
	30	16	No signal
	31	49	Dig Out-Z (Ch. 31-32)
	32	33	No signal

The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

Digital Out-Z settings

The following settings on the Parameters page are specific to the Digital Out-Z I/O function. For common dialog settings, refer to Common settings on page 321.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Enable Drive Config configuration ports Lets you enable ports to terminate the digital output. The ports can overwrite the settings of the Drive Config parameter.

The following ports are added to the block:

- DriveCfg
- DriveCfg Set
- DriveCfg Ack

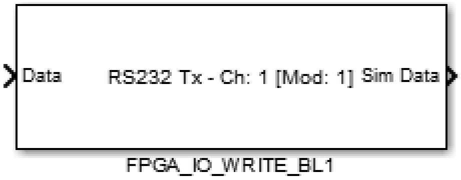
Enable High Supply configuration ports Lets you enable ports to set the voltage level of the high output signal. The ports can overwrite the settings of the High supply parameter.

The following ports are added to the block:

- HighSupply Data
- HighSupply Set
- HighSupply Ack

RS232 Tx description

Block display If you select an RS232 Tx channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs the data to be send to the RS232 Tx channel. Data type: UFix_1_0

Port	Description
	Data width: 1 Values: ▪ 0: The output voltage level is +5.5 V. ▪ 1: The output voltage level is –5.5 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. For information on the electrical characteristics of the DS2655M2 Digital I/O Module, refer to Data Sheet of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).
Output	
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. Available only if the **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: –5.5 V ... +5.5 V

> **Note**
>
> To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 🕮).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for RS232 Rx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	18	TX (Ch. 1)
	5	20	TX (Ch. 5)
	9	22	TX (Ch. 9)
	13	24	TX (Ch. 13)
	17	26	TX (Ch. 17)
	21	28	TX (Ch. 21)
	25	30	TX (Ch. 25)
	29	32	TX (Ch. 29)

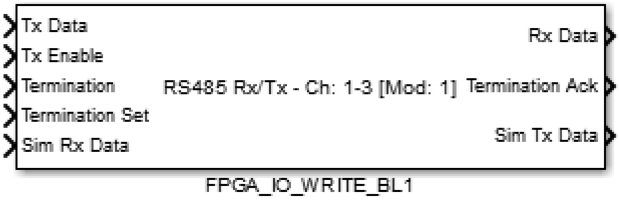
The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

RS232 Tx settings

Only common dialog settings. Refer to Common settings on page 321.

RS485 Rx/Tx description

Block display If you select an RS485 Rx/Tx channel from the channel list, the block display changes. The simulation and termination ports are displayed optionally.



Network mode With the RS485 Rx/Tx channel you can receive data from a RS485 network in half-duplex mode.

To connect to a RS485 network in simplex or full-duplex mode, use the RS485 Tx and RS485 Rx I/O functions. For details, refer to RS485 Tx description on page 338 and RS485 Rx description on page 314.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Tx Data	Outputs the data to be send to the RS485 network if the **Tx Enable** port is set to 1. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output voltage level is –5.5 V. ▪ 1: The output voltage level is +5.5 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Tx Enable	Enables the output of data values to the RS485 network and disables the high-impedance state. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is set to the high-impedance state (tri-state). The **Rx Data** outport can output received data from the RS485 network. ▪ 1: The output is enabled and transmits the data values of the **Tx Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Port	Description
Termination	Specifies whether the signals lines are terminated. A new configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if the **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The signal lines are driven directly without a termination. ▪ 1: An internal 120 Ω resistor terminates the signal lines. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new termination setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Rx Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if the **Enable Rx Data simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: –5.5 V ... +5.5 V Range exceeding is possible (using input bit widths > 1) and will be saturated to the minimum or maximum values.
Output	
Rx Data	Outputs the data that is received from the RS485 network if the **Tx Enable** inport is set to 0. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The input voltage level is negative (< 0 V). ▪ 1: The input voltage level is positive (≥ 0 V).
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0

Port	Description
	Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Tx Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable Tx Data simulation configuration port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: –5.5 V ... +5.5 V

> **Note**
>
> To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 🕮).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for RS485 Rx/Tx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Port	Channel	Connector Pin	Signal
Tx Data inport and Rx Data outport [1]	1	18	RxTx– (Ch. 1-3)
	2	2	RxTx+ (Ch. 1-3)
	3	35	No signal
	5	20	RxTx– (Ch. 5-7)
	6	4	RxTx+ (Ch. 5-7)
	7	37	No signal
	9	22	RxTx– (Ch. 9-11)
	10	6	RxTx+ (Ch. 9-11)
	11	39	No signal
	13	24	RxTx– (Ch. 13-15)
	14	8	RxTx+ (Ch. 13-15)
	15	41	No signal
	17	26	RxTx– (Ch. 17-19)
	18	10	RxTx+ (Ch. 17-19)
	19	43	No signal
	21	28	RxTx– (Ch. 21-23)
	22	12	RxTx+ (Ch. 21-23)
	23	45	No signal
	25	30	RxTx– (Ch. 25-27)
	26	14	RxTx+ (Ch. 25-27)
	27	47	No signal
	29	32	RxTx– (Ch. 29-31)
	30	16	RxTx+ (Ch. 29-31)
	31	49	No signal

[1] The RS485 Rx/Tx network is a half-duplex network.

The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

RS485 Rx/Tx settings

The following settings on the **Parameters** page are specific to the **RS485 Rx/Tx** I/O function.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω resistor terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

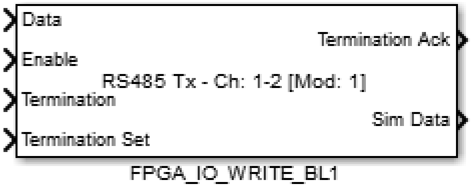
- Termination
- Termination Set
- Termination Ack

Enable Rx Data simulation port Lets you enable an inport for offline simulation data. The Sim Rx Data inport is added to the block to connect it to a Simulink-based I/O environment model.

Enable Tx Data simulation port Lets you enable an outport for offline simulation data. The Sim Tx Data outport is added to the block to connect it to a Simulink-based I/O environment model.

RS485 Tx description

Block display If you select an RS485 Tx channel from the channel list, the block display changes. The simulation and termination ports are displayed optionally.



Network mode With the RS485 Tx channel you can transmit data to a RS485 network in simplex mode. To connect to a RS485 network in full-duplex mode, also use the corresponding RS485 Rx I/O function. For details on the RS485 Rx I/O function, refer to RS485 Rx description on page 314.

To connect to a RS485 network in half-duplex mode, use the RS485 Rx/Tx I/O function. For details, refer to RS485 Rx/Tx description on page 334.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs the data to the RS485 network if the Enable port is set to 1. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output voltage level is –5.5 V. ▪ 1: The output voltage level is +5.5 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Enable	Enables the output of data values to the RS485 network. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: - 0: The output is disabled. The output voltage level is 0 V. The output does not support an high-impedance state (tri-state). - 1: The output is enabled and transmits the data values of the **Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination	Specifies whether the signals lines are terminated. A new configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if the **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: - 0: The signal lines are driven directly without a termination. - 1: An internal 120 Ω resistor terminates the signal lines. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: - No transition, or a 1 to 0 transition: A new termination setting does not take effect. - 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. Available only if the **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: –5.5 V ... +5.5 V

Note

To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the *DS2655M2 I/O Module* framework for RS485 Tx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	18	Tx– (1-2)
	2	2	Tx+ (1-2)
	5	20	Tx– (5-6)
	6	4	Tx+ (5-6)
	9	22	Tx– (9-10)
	10	6	Tx+ (9-10)
	13	24	Tx– (13-14)
	14	8	Tx+ (13-14)
	17	26	Tx– (17-18)
	18	10	Tx+ (17-18)
	21	28	Tx– (21-22)
	22	12	Tx+ (21-22)
	25	30	Tx– (25-26)
	26	14	Tx+ (25-26)
	29	32	Tx– (29-30)
	30	16	Tx+ (29-30)

The I/O functions of the *DS2655M2 I/O Module* framework share the 32 I/O channels of the DS2655M2 Digital I/O Module. For details on the signal mapping to optimize channel usage, refer to Signal Mapping of the DS2655M2 Digital I/O Module (SCALEXIO Hardware Installation and Configuration 📖).

RS485 Tx settings

The following settings on the **Parameters** page are specific to the **RS485 Tx** I/O function. For common dialog settings, refer to Common settings on page 321.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω resistor terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

- Termination
- Termination Set
- Termination Ack

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

Description Page (FPGA_IO_WRITE_BL)	343
FPGA_IO_WRITE_BL	56
FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)	320
Scaling Page (FPGA_IO_WRITE_BL)	341

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting settings for the selected I/O function.

Description

You can invert digital I/O signals and UART signals of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital Out settings

The following settings on the Scaling page are specific to the Digital Out I/O function.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- Enabled:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

Digital Out-Z settings

The following settings on the Scaling page are specific to the Digital Out-Z I/O function.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- Enabled:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

RS232 TX settings

The following settings on the Scaling page are specific to the RS232 TX I/O function.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the I/O function sets the output to –5.5 V.

  If driven with 0, the I/O function sets the output to +5.5 V.
- Enabled:

  If driven with 1, the I/O function sets the output to +5.5 V.

  If driven with 0, the I/O function sets the output to –5.5 V.

RS485 Rx/Tx settings

The following settings on the Scaling page are specific to the RS485 Rx/Tx I/O function.

Invert input polarity Lets you adapt the electrical output signal:

- Disabled:

  If the input voltage is negative (<0 V), the Rx Data port outputs a 0.

  If the input voltage is positive (≥0 V), the Rx Data port outputs a 1.
- Enabled:

  If the input voltage is negative (<0 V), the Rx Data port outputs a 1.

  If the input voltage is positive (≥0 V), the Rx Data port outputs a 0.

Invert output polarity Lets you adapt the electrical output signal:

- Disabled:

  If the Tx Data port is driven with 1, the I/O function sets the output to the high voltage level.

  If the Tx Data port is driven with 0, the I/O function sets the output to the low voltage level.
- Enabled:

  If the Tx Data port is driven with 1, the I/O function sets the output to the low voltage level.

  If the Tx Data port is driven with 0, the I/O function sets the output to the high voltage level.

RS485 Tx settings

The following settings on the Scaling page are specific to the RS485 Tx I/O function.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the I/O function sets the output to the high voltage level.

  If driven with 0, the I/O function sets the output to the low voltage level.
- Enabled:

  If driven with 1, the I/O function sets the output to the low voltage level.

  If driven with 0, the I/O function sets the output to the high voltage level.

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

References

Description Page (FPGA_IO_WRITE_BL)........343
FPGA_IO_WRITE_BL........56
FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)........320
Parameters Page (FPGA_IO_WRITE_BL)........321

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The Description page provides detailed information about the I/O type that you selected on the corresponding Unit page. The information is either in the text

field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS2655M2 I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (DS2655M2 I/O Module Settings)........................ 320
Parameters Page (FPGA_IO_WRITE_BL).. 321
Scaling Page (FPGA_IO_WRITE_BL).. 341

RTI Block Settings for the DS6651 Multi-I/O Module Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks together with at least one *DS6651 Multi-I/O Module* framework:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework
- *DS6601 (KU035) FPGA Base Board* framework
- *DS6602 (KU15P) FPGA Base Board* framework

Where to go from here

Information in this section

FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings)........................346
To configure read access to the selected I/O function when using the DS6651 Multi-I/O Module framework.

FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)........................364
To configure write access to analog and digital input signals in the FPGA model when using the DS6651 Multi-I/O Module framework.

Information in other sections

Information on selecting the framework to be used for each I/O module slot of a SCALEXIO FPGA base board:

Unit Page (FPGA_SETUP_BL)..29
To specify the FPGA framework.

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

RTI Block Settings for the DS2655 FPGA Base Board Framework............121
The block dialogs provide hardware-specific settings after you load one of the DS2655 FPGA Base Board frameworks.

RTI Block Settings for the DS6601 FPGA Base Board Framework............169
The block dialogs provide hardware-specific settings after you load the DS6601 (KU035) FPGA Base Board framework.

RTI Block Settings for the DS6602 FPGA Base Board Framework............219
The block dialogs provide hardware-specific settings after you load the DS6602 (KU15P) FPGA Base Board framework.

RTI Block Settings for the DS2655M1 I/O Module Framework................279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings)

Purpose

To configure read access to the selected I/O function when using the *DS6651 Multi-I/O Module* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Digital_Mod_<x>* and *Analog_Mod_<x>*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available channels on the selected DS6651 Multi-I/O Module determines the I/O functions that you can select:

- Analog In - Ch: 23, … , Analog In - Ch: 26
- Analog In-L - Ch: 27 and Analog In-L - Ch: 28
- Digital In - Ch: 1, … , Digital In - Ch: 16
- RS485 Rx - Ch: 1-2, … , RS485 Rx - Ch: 15-16

Digital channel dependencies

The I/O functions of the *DS6651 Multi-I/O Module* framework share the digital I/O channels that provide the digital I/O functionality. The DS6651 Multi-I/O Module provides 16 digital I/O channels. Some I/O channels provide only specific I/O functionalities, and some I/O functions use more than one I/O channel. These channel dependencies and I/O channel sharing limit the number of I/O functions that can be implemented.

For the data sheet of the DS6651 Multi-I/O Module, refer to DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

For details on the signal mapping to optimize channel usage, refer to Supported Digital Functions and Related I/O Channels (SCALEXIO Hardware Installation and Configuration 🕮).

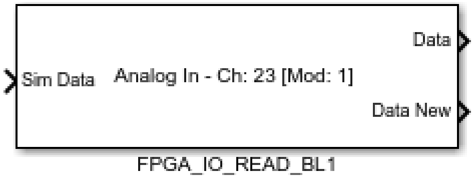
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim Data** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

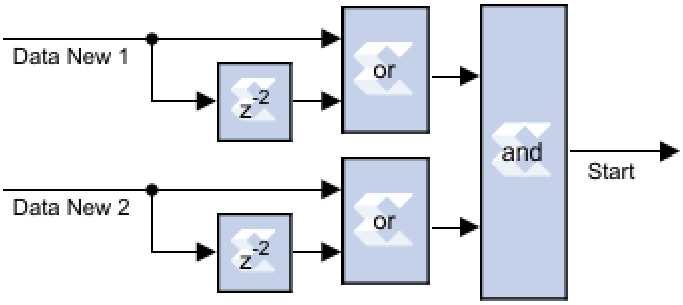
Analog In description

Block display If you select an Analog In channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range corresponds to the settings of the **Input range** parameter in V. Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the measured values of the 16-bit AD converter. Data type: Fix_22_5 1) Data width: 1 The value range of the **Data** outport depends on the setting of the **Scaling** parameter on the **Parameters** page. Refer to Scaling on page 350.
Data New	Outputs a flag that indicates the current status of the **Data** port. New measured values from analog input channels of the same I/O module are always provided synchronously. If analog inputs are read from different I/O modules, the measured values are provided either synchronously or offset by two clock cycles (16 ns). However, the sample time of the analog measurements is synchronous on different I/O modules except for 8 ns. If synchronous measured values from analog inputs of different I/O modules are required, you can implement a logic to wait with the further processing of analog values until the **Data New** ports flag new data within two clock cycles. The

<table>
<tr><th>Port</th><th>Description</th></tr>
<tr><td></td><td>following example shows a logic for the <b>Data New</b> ports to use synchronous analog input signals of different I/O modules.


Data New 1
z-2
or
Data New 2
z-2
or
and
Start

Data type: UFix_1_0
Data width: 1
Values:
▪ 0: No new value is available at the <b>Data</b> port.
▪ 1: A new valid value is available at the <b>Data</b> port.
The port is set to 1 only for one clock cycle.</td></tr>
</table>

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for analog input channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	23	29	Analog In - Ch: 23 [Mod: x]
		45	Reference
	24	14	Analog In - Ch: 24 [Mod: x]
		30	Reference
	25	31	Analog In - Ch: 25 [Mod: x]
		47	Reference
	26	48	Analog In - Ch: 26 [Mod: x]
		15	Reference

Analog In settings

The following settings on the **Parameters** page are specific to the **Analog In** I/O function. For common dialog settings, refer to Common settings on page 347.

The **Parameters** page provide the following dialog setting:

Input range Lets you specify the input voltage range that can be converted from analog to digital for the chosen ADC channel. Input voltages outside the specified range are ignored.

This electrical interface setting can be changed in ConfigurationDesk.

Scaling Lets you select whether the I/O function scales the measuring results of the A/D converter to mV.

- mV

  To output the measuring results in mV.

  The valid value range corresponds to the settings of the **Input range** parameter in mV.

  The default data type is Fix_22_5 to provide the precision of the A/D converter when using the ±1 V input voltage range.
- Bit

  To output the raw measuring results as a signed Bit value.

  The value range is -32,768 … +32,767.

  Data type: Fix_22_5

  Tip

  If you select **Bit**, you can reduce the complexity of the logic by using only 16 bits of the raw measurement result due to the 16-bit resolution of the A/D converter.
  - Eliminate the fraction bits and the most significant bit with the Xilinx **Slice** block.
  - Reinterpret the UFix_16_0 value of the **Slice** block to Fix_16_0 with the Xilinx **Reinterpret** block.

  In hardware this reduction costs nothing.

Trigger mode Lets you select the trigger mode and source for sampling the analog input voltage.

- Free running

  The ADC samples the input voltage with a fixed sample period that is set by the **Sample period** parameter.
- Trigger 1 or Trigger 2

  The ADC samples the input voltage with each trigger impulse provided by a **Trigger** I/O function. Refer to Trigger description on page 395.
- Digital In <x> (<edge type>, <filter type>)

  The selected digital input channel triggers the sampling of the analog input signal:
  - <x>: Indicates the channel number of the digital input channel.
  - <edge type>: Indicates which edge of a digital input signal triggers the ADC. A rising edge is a low to high transition, a falling edge is a high to low transition.
  - <filter type>: Indicates whether the digital signal is filtered. 8 ns is the time constant of the digital low-pass filter.

Sample period Lets you specify the sample period of the ADC in the free running mode.

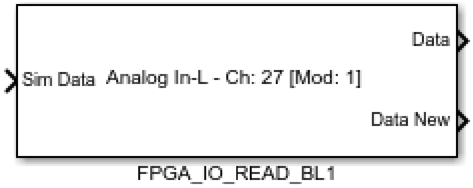
$\texttt{Sample period} = n_{\texttt{selected}} \cdot 8\ \texttt{ns}$

With the value range $25 \leq n_{\texttt{selected}} \leq 3{,}750{,}000{,}000$.

The resulting sample period is in the range 200 ns ... 30 s.

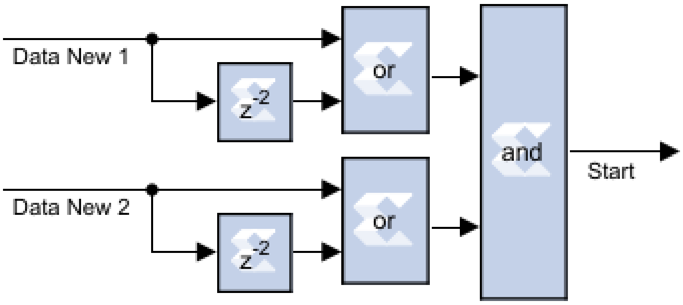
Analog In-L description

Block display If you select an Analog In-L channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range corresponds to the settings of the **Input range** parameter in V. Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the measured values of the 16-bit AD converter. Data type: Fix_22_5 1) Data width: 1 The value range of the Data output depends on the setting of the **Scaling** parameter on the **Parameters** page. Refer to Scaling on page 353.
Data New	Outputs a flag that indicates the current status of the **Data** port. New measured values from analog input channels of the same I/O module are always provided synchronously. If analog inputs are read from different I/O modules, the measured values are provided either synchronously or offset by two clock cycles (16 ns). However, the sample time of the analog measurements is synchronous on different I/O modules except for 8 ns. If synchronous measured values from analog inputs of different I/O modules are required, you can implement a logic to wait with the further processing of analog values until the **Data New** ports flag new data within two clock cycles. The

Port	Description
	following example shows a logic for the **Data New** ports to use synchronous analog input signals of different I/O modules.  Data New 1 z^{-2} or and Start Data New 2 z^{-2} or Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No new value is available at the **Data** port. ▪ 1: A new valid value is available at the **Data** port. The port is set to 1 only for one clock cycle.

[1] You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for analog input channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	27	16	Analog In-L - Ch: 27 [Mod: x]
		32	Reference
	28	33	Analog In-L - Ch: 28 [Mod: x]
		49	Reference

Analog In-L settings

The following settings on the **Parameters** page are specific to the **Analog In-L** I/O function. For common dialog settings, refer to Common settings on page 347.

The **Parameters** page provide the following dialog settings:

Input range Lets you specify the input voltage range that can be converted from analog to digital for the chosen ADC channel. Input voltages outside the specified range are ignored.

This electrical interface setting can be changed in ConfigurationDesk.

Scaling Lets you select whether the I/O function scales the measuring results of the A/D converter to mV.

- mV

  To output the measuring results in mV.

  The valid value range corresponds to the settings of the **Input range** parameter in mV.

  The default data type is Fix_22_5 to provide the precision of the A/D converter when using the ±1 V input voltage range.
- Bit

  To output the raw measuring results as a signed Bit value.

  The value range is -32,768 … +32,767.

  Data type: Fix_22_5

> **Tip**
>
> If you select **Bit**, you can reduce the complexity of the logic by using only 16 bits of the raw measurement result due to the 16-bit resolution of the A/D converter.
> - Eliminate the fraction bits and the most significant bit with the Xilinx **Slice** block.
> - Reinterpret the UFix_16_0 value of the **Slice** block to Fix_16_0 with the Xilinx **Reinterpret** block.
>
> In hardware this reduction costs nothing.

Load Config Lets you enable a 220 Ω resistor between the analog signal and the signal reference.

Trigger mode Lets you select the trigger mode and source for sampling the analog input voltage.

- Free running

  The ADC samples the input voltage with a fixed sample period that is set by the **Sample period** parameter.
- Trigger 1 or Trigger 2

  The ADC samples the input voltage with each trigger impulse provided by a **Trigger** I/O function. Refer to Trigger description on page 395.
- Digital In <x> (<edge type>, <filter type>)

  The selected digital input channel triggers the sampling of the analog input signal:
  - <x>: Indicates the channel number of the digital input channel.
  - <edge type>: Indicates which edge of a digital input signal triggers the ADC. A rising edge is a low to high transition, a falling edge is a high to low transition.
  - <filter type>: Indicates whether the digital signal is filtered. 8 ns is the time constant of the digital low-pass filter.

Sample period Lets you specify the sample period of the ADC in the free running mode.

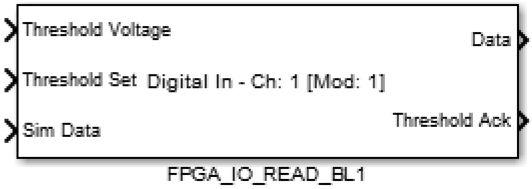
$\text{Sample period} = n_{selected} \cdot 8\ \text{ns}$

With the value range $25 \leq n_{selected} \leq 3{,}750{,}000{,}000$.

The resulting sample period is in the range 200 ns ... 30 s.

Digital In description

Block display If you select a Digital In channel from the channel list, the block display changes. Except for the outport **Data**, all the ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Threshold Voltage	Specifies the threshold voltage level in mV. A new threshold voltage level takes effect only after the **Threshold Set** port rises from 0 to 1. Available only if the **Enable Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_14_0 Data width: 1 Range: 0 mV ... 12,000 mV Update rate: FPGA clock frequency The range can be exceeded, and saturation is performed to a minimum or maximum value.
Threshold Set	Lets you set the trigger level as specified by the **Threshold Voltage** port. A new setting overwrites the settings of the **Threshold init voltage** on the **Parameters** page. Available only if **Enable Threshold Voltage configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new voltage setting does not take effect. ▪ 0 to 1 transition: If the **Threshold Set** port rises from 0 to 1, the new settings are sent to the output channel. The **Threshold Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. The threshold voltage determines a logical 0 or 1 as output of the **Data** port.

Port	Description
	Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V ... 30 V Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the current results of the digital input channel. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency For information on the electrical characteristics of the DS6651 Multi-I/O Module, refer to DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).
Threshold Ack	Outputs a flag that indicates whether the threshold voltage level is up-to-date. Available only if the **Enable Threshold Voltage configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for Digital In channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	1	18	Digital In (Ch. 1)
	2	2	Digital In (Ch. 2)
	3	35	Digital In (Ch. 3)
	4	19	Digital In (Ch. 4)
	5	3	Digital In (Ch. 5)
	6	36	Digital In (Ch. 6)
	7	20	Digital In (Ch. 7)
	8	4	Digital In (Ch. 8)
	9	37	Digital In (Ch. 9)
	10	21	Digital In (Ch. 10)
	11	22	Digital In (Ch. 11)
	12	6	Digital In (Ch. 12)
	13	39	Digital In (Ch. 13)
	14	23	Digital In (Ch. 14)
	15	7	Digital In (Ch. 15)
	16	40	Digital In (Ch. 16)

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 347.

Digital In settings

The following settings on the Parameters page are specific to the Digital In I/O function. For common dialog settings, refer to Common settings on page 347.

Threshold init voltage Lets you specify the voltage value that is used for the threshold in mV.

Range: 0 mV ... 12,000 mV in 100 mV steps.

This electrical interface setting can be changed in ConfigurationDesk.

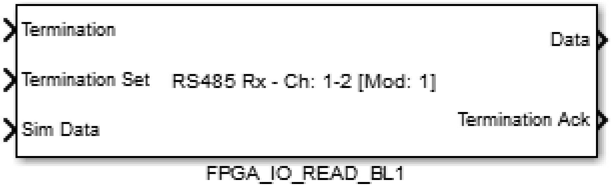
Enable Threshold Voltage configuration ports Lets you enable ports to set the threshold voltage. The ports can overwrite the value of Threshold init voltage.

The following ports are added to the block:

- Threshold Voltage
- Threshold Set
- Threshold Ack

RS485 Rx description

Block display If you select an RS485 Rx channel from the channel list, the block display changes. Except for the **Data** outport, all the ports are displayed optionally.



Network mode With the RS485 Rx channel you can receive RS485 data in simplex mode. To connect to RS485 networks in full-duplex mode, use the corresponding RS485 Tx I/O function. For details on the RS485 Tx I/O function, refer to RS485 Tx description on page 391.

To connect to a RS485 network in half-duplex mode, use the RS485 Rx/Tx I/O function. For details, refer to RS485 Rx/Tx description on page 387.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Termination	Specifies whether the signals lines are terminated. A new configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if the **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The signal lines are driven directly without a termination. ▪ 1: The signal lines are terminated via an internal RC termination with 120 Ω/5 nF. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new termination setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model.

Port	Description
	Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: -5.5 V ... +5.5 V Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data	Outputs the data received from the RS485 network. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The input voltage level is negative (< 0 V). ▪ 1: The input voltage level is positive (≥ 0 V).
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.

Note

To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 🕮).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for RS485 Rx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Outport	Digital Channel	Connector Pin	Signal
Data	1	18	RS485 Rx– (Ch. 1-2)
	2	2	RS485 Rx+ (Ch. 1-2)
	3	35	RS485 Rx– (Ch. 3-4)
	4	19	RS485 Rx+ (Ch. 3-4)
	5	3	RS485 Rx– (Ch. 5-6)
	6	36	RS485 Rx+ (Ch. 5-6)
	7	20	RS485 Rx– (Ch. 7-8)
	8	4	RS485 Rx+ (Ch. 7-8)
	9	37	RS485 Rx– (Ch. 9-10)
	10	21	RS485 Rx+ (Ch. 9-10)
	11	22	RS485 Rx– (Ch. 11-12)
	12	6	RS485 Rx+ (Ch. 11-12)
	13	39	RS485 Rx– (Ch. 13-14)
	14	23	RS485 Rx+ (Ch. 13-14)
	15	7	RS485 Rx– (Ch. 15-16)
	16	40	RS485 Rx+ (Ch. 15-16)

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 347.

RS485 Rx settings

The following settings on the **Parameters** page are specific to the **RS485 Rx** I/O function. For common dialog settings, refer to Common settings on page 347.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

- Termination
- Termination Set
- Termination Ack

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Description Page (FPGA_IO_READ_BL).......... 363
FPGA_IO_READ_BL.......... 50
FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings).......... 346
Scaling Page (FPGA_IO_READ_BL).......... 360

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting settings for the selected I/O function.

Description

You can invert digital I/O signals and UART signals of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

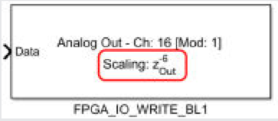
Analog In/Analog In-L settings

The following settings on the **Scaling** page are specific to the **Analog In** and **Analog In-L** I/O functions.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.

- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital In settings

The following settings on the **Scaling** page are specific to the **Digital In** I/O function.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

RS485 Rx settings

The following settings on the **Scaling** page are specific to the **RS485 Rx** I/O function.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If the input voltage is negative (<0 V), the **Data** port outputs a 0.

  If the input voltage is positive (≥0 V), the **Data** port outputs a 1.
- **Enabled**:

  If the input voltage is negative (<0 V), the **Data** port outputs a 1.

  If the input voltage is positive (≥0 V), the **Data** port outputs a 0.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Description Page (FPGA_IO_READ_BL)........363
FPGA_IO_READ_BL........50
FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings)........346
Parameters Page (FPGA_IO_READ_BL)........347

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS6651 Multi-I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

FPGA_IO_READ_BL.......... 50
FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings).......... 346
Parameters Page (FPGA_IO_READ_BL).......... 347
Scaling Page (FPGA_IO_READ_BL).......... 360

FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)

Purpose

To configure write access to analog and digital input signals in the FPGA model when using the *DS6651 Multi-I/O Module* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_IO_WRITE_BL).......... 365
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_WRITE_BL).......... 396
To specify the inverting settings for the selected I/O function.

Description Page (FPGA_IO_WRITE_BL).......... 400
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)..54
To implement read access to a physical input channel in the FPGA model.

Unit Page (FPGA_IO_READ_BL)..55
To specify the I/O type and channel to be used for read access.

Related RTI blocks

FPGA_IO_READ_BL...50
To provide read access to an external device via a physical input channel.

FPGA_IO_READ_BL (DS6651 Multi-I/O Module Settings)........................346
To configure read access to the selected I/O function when using the DS6651 Multi-I/O Module framework.

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Digital_Mod_<x>* and *Analog_Mod_<x>*, which you can select on the **Unit** page of the block. The module number <x> depends on the slot that the I/O module is connected to.

The number of the available channels on the selected DS6651 Multi-I/O Module determines the I/O functions that you can select:

- Analog Out - Ch: 17, … , Analog Out - Ch: 20
- Analog Out-T - Ch: 21 and Analog Out-T - Ch: 22
- Digital Out - Ch: 1 … Digital Out - Ch: 16
- Digital Out-Z - Ch: 1-2 … Digital Out-Z - Ch: 15-16
- Digital In/Out-Z - Ch: 1-3 ... Digital In/Out-Z - Ch: 13-15
- RS485 Rx/Tx - Ch: 1-3 ... RS485 Rx/Tx - Ch: 13-15
- RS485 Tx - Ch: 1-2 ... RS485 Tx - Ch: 15-16
- Trigger 1 and Trigger 2

Digital channel dependencies

The I/O functions of the *DS6651 Multi-I/O Module* framework share the digital I/O channels that provide the digital I/O functionality. The DS6651 Multi-I/O Module provides 16 digital I/O channels. Some I/O channels provide only specific I/O functionalities, and some I/O functions use more than one I/O channel. These channel dependencies and I/O channel sharing limit the number of I/O functions that can be implemented.

For the data sheet of the DS6651 Multi-I/O Module, refer to DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).

For details on the signal mapping to optimize channel usage, refer to Supported Digital Functions and Related I/O Channels (SCALEXIO Hardware Installation and Configuration 📖).

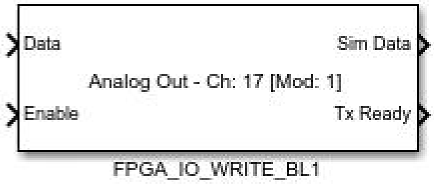
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the FPGA_IO_WRITE_BL.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

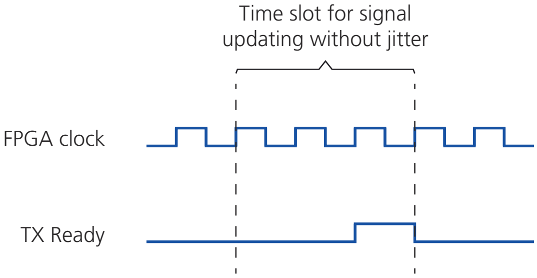
Analog Out description

Block display If you select an Analog Out channel from the channel list, the block display changes. The Enable port, the simulation port, and the Tx Ready port are displayed optionally.



I/O characteristics The following table describes the ports of the block for analog output channels:

Port	Description
Input	
Data	Outputs a voltage signal in the specified range. Data type: Fix_18_2 1) Data width: 1 The **Scaling** parameter on the **Parameters** pags specifies the value range: ▪ -10,000 mV … +10,000 mV ▪ -32,768 … +32,776 Update rate: 10.417 MSPS
Enable	Lets you enable on-demand conversion of the **Data** port value to save up to 64 ns of latency compared to free-running mode. It takes 96 ns … 120 ns until a new

Port	Description
	value can be converted. When a new value can be converted, the **Tx Ready** port outputs 1 for one clock cycle. The Tx Ready port has an internal time pattern of 32 ns. The port changes its value to 1 only within this time pattern. Therefore, the output values can jitter up to 24 ns if you do not follow the 32 ns time pattern when setting the **Enable** port. Tipp: You can ignore the Tx Ready signal if you are sure that you will never enable the **Data** port in periods shorter than 96 ns. If you enable the **Data** port in periods shorter than 96 ns, you can implement the following best practise for a jitter free output signal: Set the **Enable** port on the 32ns time pattern by comparing a free running 2-bit counter output value to 3 and resetting the counter by the signal from the **Tx Ready** port. Note: If you connect the **Tx Ready** port directly to the **Enable** port, the analog output channel works in the same way as in the free running mode. Available only if the **Enable Enable port** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1
Output	
Sim Data	Simulates the analog output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Value range: -10 V … +10 V
Tx Ready	Outputs a flag that indicates that the analog output channel of the DS6651 Multi-I/O Module is ready to be updated. The minimum update period is 96 ns. When you update data values only within the time slot for updating the output signal, the output signal has no jitter. The time slot begins two FPGA clock cycles before the flag is set to high and ends after three clock cycles.  Time slot for signal updating without jitter FPGA clock TX Ready Available only if **Enable Tx Ready port** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1

1) You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for analog output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module Module.

Outport	Channel	Connector Pin	Signal
Data	17	8	Analog Out - Ch: 17 [Mod: x]
		24	Reference
	18	25	Analog Out - Ch: 18 [Mod: x]
		41	Reference
	19	10	Analog Out - Ch: 19 [Mod: x]
		26	Reference
	20	27	Analog Out - Ch: 20 [Mod: x]
		43	Reference

Analog Out settings

The following settings on the **Parameters** page are specific to the **Analog Out** I/O function. For common dialog settings, refer to Common settings on page 366.

Scaling Lets you select the scaling of the **Data** inport.

- mV

  To specify the output voltage in mV.

  The value range is -10,000 mV … +10,000 mV.

- Bit

  To specify the output voltage with a signed bit value.

  The value range is -32,768 … +32,776 (16-bit converter).

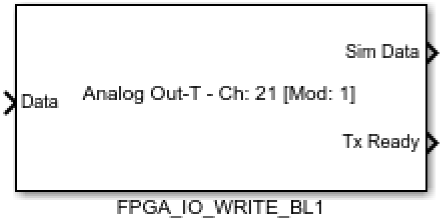
Enable Tx Ready port Lets you enable an outport to indicate that the analog output channel is ready to be updated. The **Tx Ready** port is added to the block.

Enable Enable port Lets you enable an inport to enable the conversion of the **Data** port value. The **Enable** port is added to the block.

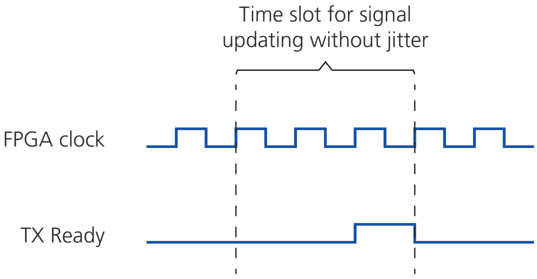
If you enable the **Enable** port, enable the **Tx Ready** port, too.

Analog Out-T description

Block display If you select an Analog Out-T channel from the channel list, the block display changes. The simulation port and the Tx Ready port are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Data	Outputs a voltage signal in the specified range. Data type: Fix_18_2 1) Data width: 1 The value range depends on the setting of the **Scaling** parameter. Refer to Scaling on page 370. Update rate: 10.417 MSPS
Output	
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 The value range depends on the setting of the **Mode** parameter. Refer to Mode on page 370.
TX Ready	Outputs a flag that indicates that the analog output channel of the DS6651 Multi-I/O Module is ready to be updated. The minimum update period is 96 ns. When you update data values only within the time slot for updating the output signal, the output signal has no jitter. The time slot begins two FPGA clock cycles before the flag is set to high and ends after three clock cycles.  Time slot for signal updating without jitter FPGA clock TX Ready Available only if **Enable Tx Ready port** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1

1) You can change the data type of the **Data** port with the **Scaling format** parameter on the **Scaling** page. Refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for analog output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Outport	Channel	Connector Pin	Signal
Data	21	44	Analog Out-T - Ch: 21 [Mod: x]
		11	Reference
	22	12	Analog Out-T - Ch: 22 [Mod: x]
		28	Reference

Analog Out-T settings

The following settings on the **Parameters** page are specific to the **Analog Out-T** I/O function. For common dialog settings, refer to Common settings on page 366.

Scaling Lets you select the scaling of the **Data** inport to specify the output voltage.

- mV

  To specify the output voltage in mV.

  The value range corresponds to the settings of the **Mode** parameter.
- Bit

  To specify the output voltage with a signed bit value.

  The value range is -32,768 … +32,776 (16-bit converter).

Mode Lets you select the converter mode of the analog output channel:

- ±10 VDC:

  The DA converter directly outputs the voltage signal without using a transformer. The output voltage range is -10 VDC … +10 VDC.
- ±20 V transformer coupled AC:

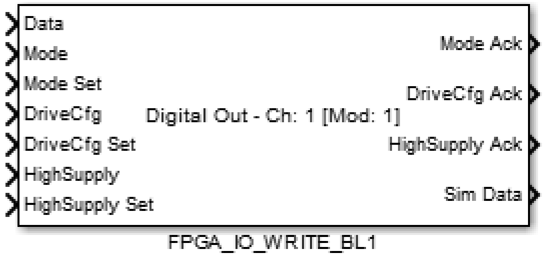
  The DA converter outputs the voltage signal via a transformer. The output voltage range is -20 VAC … +20 VAC.

  For the AC characteristics, refer to Data Sheet of the DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).

Enable Tx Ready port Lets you enable an outport to indicate that the analog output channel is ready to be updated. The **Tx Ready** port is added to the block.

Digital Out description

Block display If you select an Digital Out channel from the channel list, the block display changes. Except for the **Data** inport, all the ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. To set the voltage level, use the **High supply** on the **Parameters** page or the **HighSupply Set** inport. Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS6651 Multi-I/O Module, refer to Data Sheet of the DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Mode	Specifies the digital output mode. A new output mode takes effect only after the **Mode Set** port rises from 0 to 1. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data Type: UFix_5_0 Data width: 1 Values: ▪ 17: Enables the low-side switch that drives a load connected to VCC. ▪ 18: Enables the high-side switch that drives a load connected to GND. ▪ 19: Enables a push-pull switch that drives a load with VCC and GND.
Mode Set	Sets the output mode as specified by the **Mode** port. A new setting overwrites the settings of the **Output Mode** on the **Parameters** page. Available only if the **Enable Output Mode configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new mode does not take effect.

Port	Description
	▪ 0 to 1 transition: New settings are sent to the output channel. The **Mode Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg	Specifies the driver configuration. A new configuration takes effect only after the **DriveCfg Set** port rises from 0 to 1. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: An internal 68 Ω resistor to GND terminates the output signal. ▪ 1: The output signal is driven directly without a termination. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg Set	Sets the configuration as specified by the **DriveCfg** port. A new setting overwrites the settings of the **Drive Config** on the **Parameters** page. Available only if the **Enable Drive Config configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new configuration does not take effect. ▪ 0 to 1 transition: The output channel begins to update its channel. The **DriveCfg Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply	Specifies the supply voltage to output high-level signals. A new supply voltage takes effect only after the **HighSupply Set** port rises from 0 to 1. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The supply voltage for high signals is 5 V. ▪ 1: The supply voltage for high signals is 3.3 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply Set	Sets the supply voltage to the output high-level signal as specified by the **HighSupply** port. A new setting overwrites the setting of the **High Supply** parameter on the **Parameters** page. Available only if **Enable High Supply configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ No transition, or a 1 to 0 transition: A new supply voltage does not take effect. ▪ 0 to 1 transition: The new setting is sent to the output channel. The **HighSupply Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output Mode Ack	Outputs a flag that acknowledges a change of the output mode. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
DriveCfg Ack	Outputs a flag that acknowledges a change of the driver configuration. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
HighSupply Ack	Outputs a flag that acknowledges a change of the high supply configuration. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable simulation port** property is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V … 5 V or 0 V … 3.3 V Update rate: FPGA clock frequency

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 365.

Digital Outport	Channel	Connector Pin	Signal
Data	1	18	Digital Out (Ch. 1)
	2	2	Digital Out (Ch. 2)
	3	35	Digital Out (Ch. 3)
	4	19	Digital Out (Ch. 4)
	5	3	Digital Out (Ch. 5)
	6	36	Digital Out (Ch. 6)
	7	20	Digital Out (Ch. 7)
	8	4	Digital Out (Ch. 8)
	9	37	Digital Out (Ch. 9)
	10	21	Digital Out (Ch. 10)
	11	22	Digital Out (Ch. 11)
	12	6	Digital Out (Ch. 12)
	13	39	Digital Out (Ch. 13)
	14	23	Digital Out (Ch. 14)
	15	7	Digital Out (Ch. 15)
	16	40	Digital Out (Ch. 16)

Digital Out settings

The following settings on the **Parameters** page are specific to the **Digital Out** I/O function. For common dialog settings, refer to Common settings on page 366.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Enable Output Mode configuration ports Lets you enable ports to set the digital output mode. The ports can overwrite the settings of the Output Mode parameter.

The following ports are added to the block:

- Mode
- Mode Set
- Mode Ack

Enable Drive Config configuration ports Lets you enable ports to terminate the digital output. The ports can overwrite the settings of the Drive Config parameter.

The following ports are added to the block:

- DriveCfg
- DriveCfg Set
- DriveCfg Ack

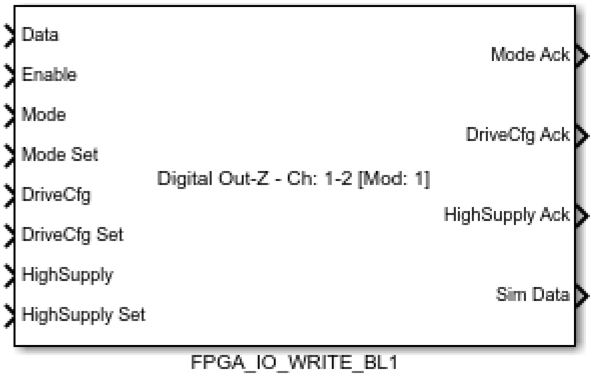
Enable High Supply configuration ports Lets you enable ports to set the voltage level of the high output signal. The ports can overwrite the settings of the High supply parameter.

The following ports are added to the block:

- HighSupply Data
- HighSupply Set
- HighSupply Ack

Digital Out-Z description

Block display If you select an Digital Out-Z channel from the channel list, the block display changes. Except for the Data and Enable inports, all the other ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. To set the voltage level, use the **High supply** on the **Parameters** page or the **HighSupply Set** inport. Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS6651 Multi-I/O Module, refer to Data Sheet of the DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 📖).
Enable	Enables the output of data values and disables the high-impedance state. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is set to the high-impedance state. ▪ 1: The output is enabled and outputs the data values of the **Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Mode	Specifies the digital output mode. A new output mode takes effect only after the **Mode Set** port rises from 0 to 1. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data Type: UFix_6_0 Data width: 1

Port	Description
	Values: ▪ 33: Enables the low-side switch that drives a load connected to VCC. ▪ 34: Enables the high-side switch that drives a load connected to GND. ▪ 35: Enables a push-pull switch that drives a load with VCC and GND.
Mode Set	Sets the output mode as specified by the **Mode** port. A new setting overwrites the settings of the **Output Mode** on the **Parameters** page. Available only if the **Enable Output Mode configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new mode does not take effect. ▪ 0 to 1 transition: New settings are sent to the output channel. The **Mode Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg	Specifies the driver configuration. A new configuration takes effect only after the **DriveCfg Set** port rises from 0 to 1. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: An internal 68 Ω resistor to GND terminates the output signal. ▪ 1: The output signal is driven directly without a termination. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg Set	Sets the configuration as specified by the **DriveCfg** port. A new setting overwrites the settings of the **Drive Config** on the **Parameters** page. Available only if the **Enable Drive Config configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new configuration does not take effect. ▪ 0 to 1 transition: The output channel begins to update its channel. The **DriveCfg Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply	Specifies the supply voltage to output high-level signals. A new supply voltage takes effect only after the **HighSupply Set** port rises from 0 to 1. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: The supply voltage for high signals is 5 V. ▪ 1: The supply voltage for high signals is 3.3 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply Set	Sets the supply voltage to the output high-level signal as specified by the **HighSupply** port. A new setting overwrites the setting of the **High Supply** parameter on the **Parameters** page. Available only if **Enable High Supply configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new supply voltage does not take effect. ▪ 0 to 1 transition: The new setting is sent to the output channel. The **HighSupply Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Mode Ack	Outputs a flag that acknowledges a change of the output mode. Available only if the **Enable Output Mode configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
DriveCfg Ack	Outputs a flag that acknowledges a change of the driver configuration. Available only if the **Enable Drive Config configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
HighSupply Ack	Outputs a flag that acknowledges a change of the high supply configuration. Available only if **Enable High Supply configuration ports** parameter is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable simulation port** property is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V … 5 V or 0 V … 3.3 V

Port	Description
	Update rate: FPGA clock frequency

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital output channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Inport	Digital Channel	Connector Pin	Signal
Data	1	18	Digital Out-Z (Ch. 1-2)
	2	2	No signal
	3	35	Digital Out-Z (Ch. 3-4)
	4	19	No signal
	5	3	Digital Out-Z (Ch. 5-6)
	6	36	No signal
	7	20	Digital Out-Z (Ch. 7-8)
	8	4	No signal
	9	37	Digital Out-Z (Ch. 9-10)
	10	21	No signal
	11	22	Digital Out-Z (Ch. 11-12)
	12	6	No signal
	13	39	Digital Out-Z (Ch. 13-14)
	14	23	No signal
	15	7	Digital Out-Z (Ch. 15-16)
	16	40	No signal

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 365.

Digital Out-Z settings

The following settings on the Parameters page are specific to the Digital Out-Z I/O function. For common dialog settings, refer to Common settings on page 366.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Enable Output Mode configuration ports Lets you enable ports to set the digital output mode. The ports can overwrite the settings of the Output Mode parameter.

The following ports are added to the block:

- Mode
- Mode Set
- Mode Ack

Enable Drive Config configuration ports Lets you enable ports to terminate the digital output. The ports can overwrite the settings of the Drive Config parameter.

The following ports are added to the block:

- DriveCfg
- DriveCfg Set
- DriveCfg Ack

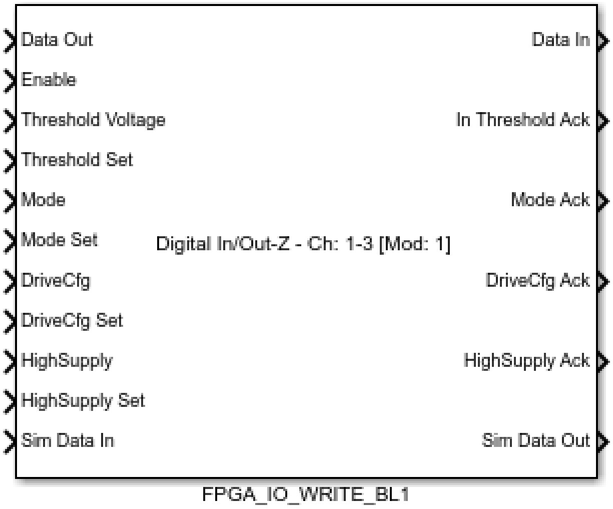
Enable High Supply configuration ports Lets you enable ports to set the voltage level of the high output signal. The ports can overwrite the settings of the High supply parameter.

The following ports are added to the block:

- HighSupply Data
- HighSupply Set
- HighSupply Ack

Digital In/Out-Z description

Block display If you select a Digital In/Out channel from the channel list, the block display changes. Except for the **Data In**, **Data Out**, and **Enable** ports, all the ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data Out	Outputs a signal in the specified range if the **Enable** port is set to 1. To set the voltage level, use the **High supply** parameter on the **Parameters** page or the **High Supply** inport . Data Type: UFix_1_0 Data width: 1 If driven with 0, the hardware outputs a low-level signal. If driven with 1, the hardware outputs a high-level signal. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS6651 Multi-I/O Module, refer to Data Sheet of the DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).
Enable	Enables the output of data values and disables the high-impedance state. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is set to the high-impedance state. ▪ 1: The output is enabled and outputs the data values of the **Data** inport.

Port	Description
	Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Threshold Voltage	Specifies a trigger level in mV. A new threshold voltage level takes effect only after the **Threshold Set** port rises from 0 to 1. Available only if **Enable digital in configuration ports** is set on the **Parameters** page. Data type: UFix_14_0 Data width: 1 Range: 0 mV ... 12,000 mV Update rate: FPGA clock frequency The range can be exceeded, and saturation is performed to a minimum or maximum value.
Threshold Set	Lets you set the trigger level as specified by the **Threshold Voltage** port. A new setting overwrites the settings of **Threshold init voltage** parameter on the **Parameters** page. Available only if the **Enable digital in configuration ports** is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new voltage setting does not take effect. ▪ 0 to 1 transition: If the **Threshold Set** port rises from 0 to 1, the new settings are sent to the output channel. The **Threshold Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Mode	Specifies the digital output mode. A new output mode takes effect only after the **Mode Set** port rises from 0 to 1. Available only if the **Enable digital out configuration ports** parameter is set on the **Parameters** page. Data Type: UFix_6_0 Data width: 1 Values: ▪ 49: Enables the low-side switch that drives a load connected to VCC. ▪ 50: Enables the high-side switch that drives a load connected to GND. ▪ 51: Enables a push-pull switch that drives a load with VCC and GND.
Mode Set	Sets the output mode as specified by the **Mode** port. A new setting overwrites the settings of the **Output Mode** on the **Parameters** page. Available only if the **Enable digital out configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new mode does not take effect. ▪ 0 to 1 transition: New settings are sent to the output channel. The **Mode Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency

Port	Description
	Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg	Specifies the driver configuration. A new configuration takes effect only after the **DriveCfg Set** port rises from 0 to 1. Available only if the **Enable digital out configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Value range: ▪ 0: An internal 68 Ω resistor to GND terminates the output signal. ▪ 1: The output signal is driven directly without a termination. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
DriveCfg Set	Sets the configuration as specified by the **DriveCfg** port. A new setting overwrites the settings of the **Drive Config** on the **Parameters** page. Available only if the **Enable digital out configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Value range: ▪ No transition, or a 1 to 0 transition: A new configuration does not take effect. ▪ 0 to 1 transition: The output channel begins to update its channel. The **DriveCfg Ack** port outputs a flag if the channel is up-to-date. Update rate: FPGA clock frequency Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply	Specifies the supply voltage to output high-level signals. A new supply voltage takes effect only after the **HighSupply Set** port rises from 0 to 1. Available only if the **Enable digital out configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The supply voltage for high signals is 5 V. ▪ 1: The supply voltage for high signals is 3.3 V. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
HighSupply Set	Sets the supply voltage to the output high-level signal as specified by the **HighSupply** port. A new setting overwrites the setting of the **High Supply** on the **Parameters** page. Available only if the **Enable digital out configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Range: ▪ No transition, or a 1 to 0 transition: A new supply voltage does not take effect. ▪ 0 to 1 transition: The new setting is sent to the output channel. The **HighSupply Ack** port outputs a flag if the channel is up-to-date.

Port	Description
	Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Data In	Simulates an input signal that you can connect to a Simulink-based I/O environment model. The threshold voltage determines a logical 0 or 1 as output of the **Data** port. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: 0 V ... 12 V Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Data In	Outputs the current results of the digital input channel. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency For information on the electrical characteristics of the DS6651 Multi-I/O Module, refer to DS6651 Multi-I/O Module (SCALEXIO Hardware Installation and Configuration 🕮).
In Threshold Ack	Outputs a flag that acknowledges a change of the threshold voltage level configuration. Available only if the **Enable digital in configuration ports** is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Mode Ack	Outputs a flag that acknowledges a change of the output mode. Available only if **Enable digital out configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
DriveCfg Ack	Outputs a flag that acknowledges a change of the driver configuration. Available only if **Enable digital out configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.

Port	Description
HighSupply Ack	Outputs a flag that acknowledges a change of the high supply configuration. Available only if **Enable digital out configuration ports** is set on the **Parameters** page. Data type: Double Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data Out	Simulates an output signal that you can connect to a Simulink-based I/O environment model. The signal is in the same range as specified for the hardware output signal. Available only if **Enable digital in simulation port** is set on the **Parameters** page. Data type: Double Output voltage: 0 V … 5 V or 0 V … 3.3 V Update rate: FPGA clock frequency

If the value of the **Data In** inport exceeds the specified data width, only the lowest bit is used. If the hardware signal or the value of the **Sim Data In** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

Note

Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for digital signals. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS2655M2 Digital I/O Module.

Ports	Digital Channel	Connector Pin	Signal
Data In and Data Out	1	18	Digital In/Out-Z (Ch. 1-3)
	2	2	No signal
	3	35	No signal
	4	19	Usable by other I/O functions
	5	3	Digital In/Out-Z (Ch. 5-7)
	6	36	No signal
	7	20	No signal
	8	4	Usable by other I/O functions
	9	37	Digital In/Out-Z (Ch. 9-11)
	10	21	No signal
	11	22	No signal
	12	6	Usable by other I/O functions
	13	39	Digital In/Out-Z (Ch. 13-15)
	14	23	No signal
	15	7	No signal
	16	40	Usable by other I/O functions

Digital In/Out-Z settings

The following settings on the Parameters page are specific to the Digital In/Out-Z I/O function.

Output Mode Lets you select the output mode.

- Low-side switch

  Lets you actively drive the output to GND to output a low-level signal.

  An external load to VCC is required to output a high-level signal.
- High-side switch

  Lets you actively drive the output to VCC to output a high-level signal.

  An external load to GND is necessary to output a low-level signal.
- Push-pull

  Lets you drive the output between VCC and GND.

  An external load is not required.

This electrical interface setting can be changed in ConfigurationDesk.

Drive Config Lets you enable/disable the termination of the signal line by an internal resistor.

- Direct Drive

  Lets you directly drive the I/O signal. The internal termination resistor is disabled.
- 68 Ohm Terminated

  Lets you terminate the I/O signal with an internal 68 Ω resistor.

This electrical interface setting can be changed in ConfigurationDesk.

High supply Lets you select the VCC voltage that determines the high-level voltage for the high-side switch.

This electrical interface setting can be changed in ConfigurationDesk.

Threshold init voltage Lets you specify the voltage value that is used for the threshold in mV.

Range: 0 mV ... 12,000 mV in 100 mV steps.

This electrical interface setting can be changed in ConfigurationDesk.

Enable digital in configuration ports Lets you enable ports to set the threshold voltage. The ports can overwrite the settings of In Threshold init voltage.

The following ports are added to the block:

- Threshold Voltage
- Threshold Set
- In Threshold Ack

Enable digital out simulation port Lets you enable ports to configure the digital output. The ports can overwrite the settings of the Output Mode, Drive Config, and High supply parameters.

The following ports are added to the block:

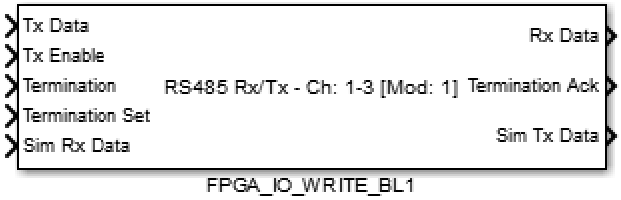
- Mode
- Mode set
- DriveCfg
- DriveCfg Set
- HighSupply
- HighSupply Set

Enable digital in simulation port Lets you enable an inport for offline simulation data. The Sim Data In inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model. This port is relevant only if the signal direction is in.

Enable digital out simulation port Lets you enable an outport for offline simulation data. The Sim Data Out port is added to the block to connect it to a Simulink-based I/O environment model.

RS485 Rx/Tx description

Block display If you select an RS485 Rx/Tx channel from the channel list, the block display changes. The simulation and termination ports are displayed optionally.



Network mode With the RS485 Rx/Tx channel you can receive data from a RS485 network in half-duplex mode.

To connect to a RS485 network in simplex or full-duplex mode, use the RS485 Tx and RS485 Rx I/O functions. For details, refer to RS485 Tx description on page 391 and RS485 Rx description on page 357.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Tx Data	Outputs the data to be send to the RS485 network if the **Tx Enable** port is set to 1. The differential output voltage level depends on the **High Supply** setting on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output voltage level is low. ▪ 1: The output voltage level is high. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Tx Enable	Enables the output of data values to the RS485 network and disables the high-impedance state. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is set to the high-impedance state (tri-state). The **Rx Data** outport can output received data from the RS485 network. ▪ 1: The output is enabled and transmits the data values of the **Tx Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination	Specifies whether the signals lines are terminated. A new configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if the **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The signal lines are driven directly without a termination. ▪ 1: The signal lines are terminated via an internal RC termination with 120 Ω/5 nF. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0

Port	Description
	Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new termination setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim Rx Data	Simulates an input signal that you can connect to a Simulink-based I/O environment model. Available only if **Enable Rx Data simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: ±3.3 V or ±5 V, depends on the **High Supply** parameter setting. Range exceeding is possible and will be saturated to the minimum or maximum values.
Output	
Rx Data	Outputs the data that is received from the RS485 network if the **Tx Enable** inport is set to 0. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The input voltage level is negative (< 0 V). ▪ 1: The input voltage level is positive (≥ 0 V).
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1 The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Tx Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. Available only if the **Enable Tx Data simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: ±3.3 V or ±5 V, depends on the **High Supply** parameter setting.

Note

To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide 📖).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for RS485 Rx/Tx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Port	Digital Channel	Connector Pin	Signal
Tx Data inport and Rx Data outport [1]	1	18	RS485 RxTx– (Ch. 1-3)
	2	2	RS485 RxTx+ (Ch. 1-3)
	3	35	No signal
	4	19	Usable by other I/O functions
	5	3	RS485 RxTx– (Ch. 5-7)
	6	36	RS485 RxTx+ (Ch. 5-7)
	7	20	No signal
	8	4	Usable by other I/O functions
	9	37	RS485 RxTx– (Ch. 9-11)
	10	21	RS485 RxTx+ (Ch. 9-11)
	11	22	No signal
	12	6	Usable by other I/O functions
	13	39	RS485 RxTx– (Ch. 13-15)
	14	23	RS485 RxTx+ (Ch. 13-15)
	15	7	No signal
	16	40	Usable by other I/O functions

[1] The RS485 RxTx network is a half-duplex network.

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 365.

RS485 Rx/Tx settings

The following settings on the **Parameters** page are specific to the **RS485 Rx/Tx** I/O function.

High supply Lets you set the differential output voltage to ±3.3 V or ±5 V.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

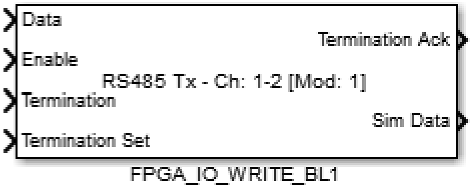
- Termination
- Termination Set
- Termination Ack

Enable Rx Data simulation port Lets you enable an inport for offline simulation data. The Sim Rx Data inport is added to the block to connect it to a Simulink-based I/O environment model.

Enable Tx Data simulation port Lets you enable an outport for offline simulation data. The Sim Tx Data outport is added to the block to connect it to a Simulink-based I/O environment model.

RS485 Tx description

Block display If you select an RS485 Tx channel from the channel list, the block display changes. The simulation and termination ports are displayed optionally.



Network mode With the RS485 Tx channel you can transmit data to a RS485 network in simplex mode. To connect to a RS485 network in full-duplex mode, also use the corresponding RS485 Rx I/O function. For details on the RS485 Rx I/O function, refer to RS485 Rx description on page 357.

To connect to a RS485 network in half-duplex mode, use the RS485 Rx/Tx I/O function. For details, refer to RS485 Rx/Tx description on page 387.

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs the data to the RS485 network if the Enable port is set to 1. The differential output voltage level depends on the High Supply setting on the Parameters page. Data type: UFix_1_0 Data width: 1

Port	Description
	Values: ▪ 0: The output voltage level is low. ▪ 1: The output voltage level is high. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Enable	Enables the output of data values to the RS485 network. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The output is disabled. The output voltage level is 0 V. The output does not support an high-impedance state (tri-state). ▪ 1: The output is enabled and transmits the data values of the **Data** inport. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination	Specifies whether the signals lines are terminated. A new configuration takes effect only after the **Termination Set** port rises from 0 to 1. Available only if the **Enable RS485 Termination configuration ports** is set on the **Parameters** page. Data Type: UFix_1_0 Data width: 1 Values: ▪ 0: The signal lines are driven directly without a termination. ▪ 1: The signal lines are terminated via an internal RC termination with 120 Ω/5 nF. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Termination Set	Lets you set the termination of the RS485 signal lines as specified by the **Termination** port. A new setting overwrites the settings of the **RS485 Termination** on the **Parameters** page. Available only if the **Enable RS485 Termination configuration ports** parameter is set. Data type: UFix_1_0 Data width: 1 Values: ▪ No transition, or a 1 to 0 transition: A new termination setting does not take effect. ▪ 0 to 1 transition: The new settings are sent to the channel. The **Termination Ack** port outputs a flag if the channel is up-to-date. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Termination Ack	Outputs a flag that acknowledges a change of the termination setting. Available only if **Enable RS485 Termination configuration ports** parameter is set on the **Parameters** page. Data type: UFix_1_0 Data width: 1

Port	Description
	The value 1 acknowledges the update. The flag is set high only within one clock cycle.
Sim Data	Simulates an output signal that you can connect to a Simulink-based I/O environment model. Available only if the **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Range: ±3.3 V or ±5 V, depends on the **High Supply** parameter setting.

> **Note**
>
> To set the baud rate for a serial transmission, refer to Using the UART Demo Model for SCALEXIO Systems (RTI FPGA Programming Blockset Guide).

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide).

I/O mapping The following I/O mapping is relevant if you use the *DS6651 Multi-I/O Module* framework for RS485 Tx channels. The signals are available at the female 50-pin Sub-D I/O connector of the respective DS6651 Multi-I/O Module.

Outport	Digital Channel	Connector Pin	Signal
Data	1	18	RS485 Tx– (Ch. 1-2)
	2	2	RS485 Tx+ (Ch. 1-2)
	3	35	RS485 Tx– (Ch. 3-4)
	4	19	RS485 Tx+ (Ch. 3-4)
	5	3	RS485 Tx– (Ch. 5-6)
	6	36	RS485 Tx+ (Ch. 5-6)
	7	20	RS485 Tx– (Ch. 7-8)
	8	4	RS485 Tx+ (Ch. 7-8)
	9	37	RS485 Tx– (Ch. 9-10)
	10	21	RS485 Tx+ (Ch. 9-10)
	11	22	RS485 Tx– (Ch. 11-12)
	12	6	RS485 Tx+ (Ch. 11-12)
	13	39	RS485 Tx– (Ch. 13-14)
	14	23	RS485 Tx+ (Ch. 13-14)
	15	7	RS485 Tx– (Ch. 15-16)
	16	40	RS485 Tx+ (Ch. 15-16)

The digital I/O functions of the *DS6651 Multi-I/O Module* framework share the channels that provide the digital I/O functionality. For more information, refer to Digital channel dependencies on page 365.

RS485 Tx settings

The following settings on the Parameters page are specific to the RS485 Tx I/O function. For common dialog settings, refer to Common settings on page 366.

High supply Lets you set the differential output voltage to ±3.3 V or ±5 V.

RS485 Termination Lets you enable an internal termination between the signal lines. The setting can be overwritten by the RS485 termination ports.

- Open

  The signal lines are not terminated.
- Terminated

  An internal 120 Ω/5 nF RC termination terminates the signal lines.

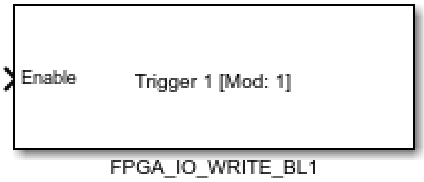
This electrical interface setting can be changed in ConfigurationDesk.

Enable RS485 Termination configuration ports Lets you enable ports to set the termination of the RS485 input. The following ports are added to the block:

- Termination
- Termination Set
- Termination Ack

Trigger description

Block display If you select a Trigger channel from the channel list, the block display changes. The simulation and termination ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Enable	Lets you trigger the analog measurement of the DS6651 Multi-I/O Module. Data Type: UFix_1_0 Data width: 1 A transition from 0 to 1 provides a trigger impulse that can be used by the Analog In/Analog In-L I/O functions. Refer to Analog In description on page 348 and Analog In-L description on page 351.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping No external connection.

Trigger settings

None

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 🕮)

References

Description Page (FPGA_IO_WRITE_BL)	400
FPGA_IO_WRITE_BL	56
FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)	364
Scaling Page (FPGA_IO_WRITE_BL)	396

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting settings for the selected I/O function.

Description

You can invert digital I/O signals and UART signals of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

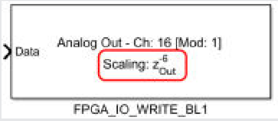
Analog Out/Analog Out-T settings

The following settings on the Scaling page are specific to the Analog Out and Analog Out-T I/O functions.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the Scaling page to scale analog signals, the data type of the Data port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- Signed/Unsigned:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the Number of bits and Binary point (fraction width) parameters.
- Single:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling factor** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling offset** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation minimum value Lets you specify the minimum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum**

value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the Scaling factor parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the Scaling offset parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital Out/Digital Out-Z settings

The following settings on the Scaling page are specific to the Digital Out and Digital Out-Z I/O functions.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- Enabled:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

Digital In/Out-Z settings

The following settings on the Scaling page are specific to the Digital In/Out-Z I/O function.

Invert input polarity Lets you invert the measured values of the electrical input signal:

- Disabled:

  The Data port outputs the signals as measured: A low-high transition results in a 1 and vice versa.

- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

Invert output polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- **Enabled**:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

RS485 Rx/Tx settings

The following settings on the **Scaling** page are specific to the **RS485 Rx/Tx** I/O function.

Invert input polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If the input voltage is negative (<0 V), the **Rx Data** port outputs a 0.

  If the input voltage is positive (≥0 V), the **Rx Data** port outputs a 1.
- **Enabled**:

  If the input voltage is negative (<0 V), the **Rx Data** port outputs a 1.

  If the input voltage is positive (≥0 V), the **Rx Data** port outputs a 0.

Invert output polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If the **Tx Data** port is driven with 1, the I/O function sets the output to the high voltage level.

  If the **Tx Data** port is driven with 0, the I/O function sets the output to the low voltage level.
- **Enabled**:

  If the **Tx Data** port is driven with 1, the I/O function sets the output to the low voltage level.

  If the **Tx Data** port is driven with 0, the I/O function sets the output to the high voltage level.

RS485 Tx settings

The following settings on the **Scaling** page are specific to the **RS485 Tx** I/O function.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If driven with 1, the I/O function sets the output to the high voltage level.

  If driven with 0, the I/O function sets the output to the low voltage level.
- **Enabled**:

  If driven with 1, the I/O function sets the output to the low voltage level.

  If driven with 0, the I/O function sets the output to the high voltage level.

Trigger settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

Description Page (FPGA_IO_WRITE_BL)..400
FPGA_IO_WRITE_BL..56
FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)..364
Parameters Page (FPGA_IO_WRITE_BL)..365

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *DS6651 Multi-I/O Module* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

Basics

Configuring the Basic Functionality (FPGA) (ConfigurationDesk I/O Function Implementation Guide 📖)

References

FPGA_IO_WRITE_BL..56
FPGA_IO_WRITE_BL (DS6651 Multi-I/O Module Settings)..364
Parameters Page (FPGA_IO_WRITE_BL)..365
Scaling Page (FPGA_IO_WRITE_BL)..396

RTI Block Settings for the DS660X_MGT Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks together with the *DS660X_MGT* framework:

- *DS6601 (KU035) FPGA Base Board* framework
- *DS6602 (KU15P) FPGA Base Board* framework

Where to go from here

Information in this section

FPGA_IO_READ_BL (MGT In Settings)..402
To configure read access to the MGT communication bus when using the DS660X_MGT framework.

FPGA_IO_WRITE_BL (MGT Out Settings)..409
To configure write access to the MGT communication bus when using the DS660X_MGT framework.

Information in other sections

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

RTI Block Settings for the DS2655 FPGA Base Board Framework............121
The block dialogs provide hardware-specific settings after you load one of the DS2655 FPGA Base Board frameworks.

RTI Block Settings for the DS6601 FPGA Base Board Framework............169
The block dialogs provide hardware-specific settings after you load the DS6601 (KU035) FPGA Base Board framework.

RTI Block Settings for the DS6602 FPGA Base Board Framework............219
The block dialogs provide hardware-specific settings after you load the DS6602 (KU15P) FPGA Base Board framework.

RTI Block Settings for the DS2655M1 I/O Module Framework................279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

RTI Block Settings for the DS2655M2 I/O Module Framework................307
The block dialogs provide hardware-specific settings after you load the DS2655M2 I/O Module framework.

RTI Block Settings for the DS6651 Multi-I/O Module Framework............345
The block dialogs provide hardware-specific settings after you load the DS6651 Multi-I/O Module framework.

RTI Block Settings for the Inter-FPGA Interface Framework.....................417
The block dialogs provide hardware-specific settings after you load the Inter-FPGA Interface framework for SCALEXIO.

FPGA_IO_READ_BL (MGT In Settings)

Purpose

To configure read access to the MGT communication bus when using the *DS660X_MGT* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O type *MGT*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- Aurora 64b66b In 1 … 4
- Aurora 64b66b 128 Bit In 1 … 4
- MGT In

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the FPGA_IO_READ_BL.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim Data** inport is added to the block to connect it to any Simulink signal used for simulation.

Aurora 64b66b In description

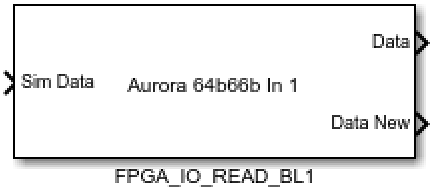
Purpose To read 64-bit data values from an MGT communication channel using the Aurora 64b66b link-layer protocol.

Used communication protocol setting The I/O function uses the Aurora 64B/66B protocol with the following settings:

- Transceiver: GTH
- Line Rate: 10.3125 Gbps
- Dataflow Mode: Duplex
- Interface: Framing
- Flow Control: NFC
- USER K: off
- Little Endian Support: off
- CRC: off

For more information on the protocol, refer to https://www.xilinx.com/support/documentation/ip_documentation/aurora_64b66b/v12_0/pg074-aurora-64b66b.pdf.

Block display If you select an Aurora 64b66b In channel from the channel list, the block display changes. The inport **Sim Data** is displayed optionally.



I/O characteristics The following table describes the ports of the block for the MGT module.

Port	Description
Input	
Sim Data	Simulates a data exchange via the MGT communication bus including floating-point to fixed-point data conversion. Available only if the **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Reads a 64-bit data value from the MGT communication bus. User data transfer rate: Max. 8 Gbit/s (125 MHz FPGA clock, 64 bits) MGT latency1): ▪ With maximum data rate: 456 ns ▪ Single words: Max. 472 ns, typ. 384 ns If you implement inter-FPGA communication via MGT modules, clock drifts can result in additional latencies. Refer to Implementing Inter-FPGA Communication via MGT Modules (RTI FPGA Programming Blockset Guide 🕮). Data type: UFix_64_0 Data width: 1
Data New	Indicates whether a new data value was received by the MGT module. Data type: UFix_1_0 Data width: 1 If the MGT module contains a new value, the flag changes from 0 to 1 for one clock cycle: ▪ 0: No new data available. ▪ 1: New data available.

1) Latency between sender and receiver via an optical loopback.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

Aurora 64b66b In settings Only common dialog settings. Refer to Common settings on page 403.

Aurora 64b66b 128 Bit In description

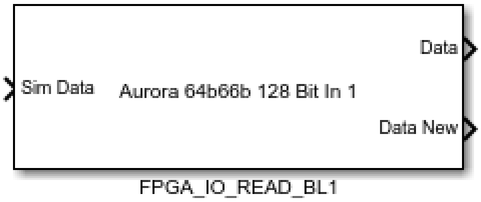
Purpose To read 128-bit data values from an MGT communication channel using the Aurora 64b66b link-layer protocol.

Used communication protocol setting The I/O function uses the Aurora 64B/66B protocol with the following settings:

- Transceiver: GTH
- Line Rate: 10.3125 Gbps
- Dataflow Mode: Duplex
- Interface: Framing
- Flow Control: NFC
- USER K: off
- Little Endian Support: off
- CRC: off

For more information on the protocol, refer to https://www.xilinx.com/support/documentation/ip_documentation/aurora_64b66b/v12_0/pg074-aurora-64b66b.pdf.

Block display If you select an Aurora 64b66b 128 Bit In channel from the channel list, the block display changes. The inport **Sim Data** is displayed optionally.



I/O characteristics The following table describes the ports of the block for the MGT module.

Port	Description
Input	
Sim Data	Simulates a data exchange via the MGT communication bus including floating-point to fixed-point data conversion. The converting can lead to inaccuracies. Available only if the **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Reads a 128-bit data value from the MGT communication bus. User data transfer rate: Max. 10.3125 Gbit/s, limited by the MGT module. MGT latency[1)]: ▪ With maximum data rate: Max. 6.272 µs, typ. 6.192 µs

Port	Description
	The latency increases, because the TX-FIFO buffer becomes full when the data stream fills the buffer with 16 Gbit/s (128 bits at 125 MHz). - Single words: Max. 472 ns, typ. 384 ns If you implement inter-FPGA communication via MGT modules, clock drifts can result in additional latencies. Refer to Implementing Inter-FPGA Communication via MGT Modules (RTI FPGA Programming Blockset Guide 📖). Data type: UFix_128_0 Data width: 1
Data New	Indicates whether a new data value was received by the MGT module. Data type: UFix_1_0 Data width: 1 If the MGT module contains a new value, the flag changes from 0 to 1 for one clock cycle: - 0: No new data available. - 1: New data available.

[1] Latency between sender and receiver via an optical loopback.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

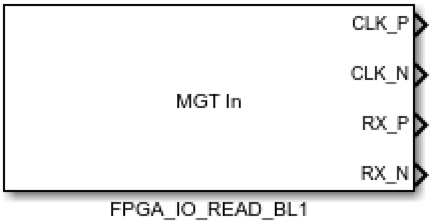
Aurora 64b66b 128 Bit In settings

Only common dialog settings. Refer to Common settings on page 403.

MGT In description

Purpose To provide the information about the connection between the GTH transceivers and the MGT module and to specify the reference clock frequency. The information is required for customer-specific protocol blocks that configure the GTH transceivers. A GTH transceiver is a configurable transceiver of the XILINX UltraScale FPGA architecture.

Block display If you select an MGT In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the MGT module.

Port	Description
Output	
CLK_P CLK_N	Provides the MGT reference clock frequency.

Port	Description
	The ports represent the differential signal of an internal clock. The ports must be connected to a block that provides the configuration for the GTH transceivers. The reference frequency is specified on the **Parameters** page. Data type: UFix_1_0 Data width: 1
RX_P RX_N	Reads the raw data from the MGT module. The ports represent the differential output signals of the MGT module. The ports must be connected to a block that provides the configurations for the GTH transceiver. Data type: UFix_4_0 Each bit represents the output of one MGT channel. Data width: 1 Update rate: Clock frequency of the GTH transceivers.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

Demo model For implementing an MGT communication with a customized protocol, refer to Modeling MGT Communication Using a Customized Protocol (RTI FPGA Programming Blockset Guide 🕮).

MGT In settings

The **Parameters** page provides the following dialog setting:

MGT reference clock frequency Lets you specify the reference clock frequency that is used to generate the MGT clock frequency.

The reference clock frequency depends on the protocol type, transfer rate, and internal scaling factors. In many cases, the reference clock frequency for the MGT module of the FPGA base board is 156.25 MHz.

For more information, refer to Modeling MGT Communication Using a Customized Protocol (RTI FPGA Programming Blockset Guide 🕮).

Related topics

References

Description Page (FPGA_IO_READ_BL)..408
FPGA_IO_READ_BL..50
FPGA_IO_READ_BL (MGT In Settings)..402

Scaling Page (FPGA_IO_READ_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard DS660X_MGT framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (MGT In Settings)... 402
Parameters Page (FPGA_IO_READ_BL)... 403

FPGA_IO_WRITE_BL (MGT Out Settings)

Purpose

To configure write access to the MGT communication bus when using the *DS660X_MGT* framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O type *MGT*, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- Aurora 64b66b Out 1 … 4
- Aurora 64b66b 128 Bit Out 1 … 4
- MGT Out 1 ... 4

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the FPGA_IO_WRITE_BL.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim Data** outport is added to the block to connect it to any Simulink signal used for simulation.

Aurora 64b66b Out description

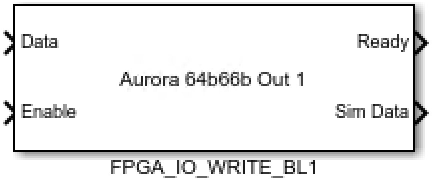
Purpose To write 64-bit data values to an MGT communication channel using the Aurora 64b66b link-layer protocol.

Used communication protocol setting The I/O function uses the Aurora 64B/66B protocol with the following settings:

- Transceiver: GTH
- Line Rate: 10.3125 Gbps
- Dataflow Mode: Duplex
- Interface: Framing
- Flow Control: NFC
- USER K: off
- Little Endian Support: off
- CRC: off

For more information on the protocol, refer to https://www.xilinx.com/support/documentation/ip_documentation/aurora_64b66b/v12_0/pg074-aurora-64b66b.pdf.

Block display If you select an Aurora 64b66b Out channel from the channel list, the block display changes. The outport **Sim Data** is displayed optionally.



I/O characteristics The following table describes the ports of the block for the MGT module:

Port	**Description**
Input	
Data	Writes a 64-bit data value to the MGT communication bus. User data transfer rate: Max. 8 Gbit/s (125 MHz FPGA clock, 64 bits) MGT latency[1]): ▪ With maximum data rate: 456 ns ▪ Single words: Max. 472 ns, typ. 384 ns If you implement inter-FPGA communication via MGT modules, clock drifts can result in additional latencies. Refer to Implementing Inter-FPGA Communication via MGT Modules (RTI FPGA Programming Blockset Guide 🕮). Data type: UFix_64_0 Data width: 1
Enable	Enables the write access to the MGT communication bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No write access. ▪ 1: The **Data** port value of the current clock cycle is written on the MGT communication bus.
Output	
Ready	Outputs a flag that indicates that the MGT module is ready to write new data on the MGT communication bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The MGT channel is busy. ▪ 1: New data values can be written to the MGT communication bus.
Sim Data	Simulates a data exchange via the MGT communication bus including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1

[1] Latency between sender and receiver via an optical loopback.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

Aurora 64b66b Out settings

Only common dialog settings. Refer to Common settings on page 410.

Aurora 64b66b 128 Bit Out description

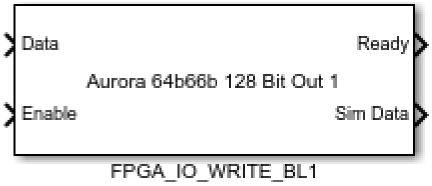
Purpose To write 128-bit data values to an MGT communication channel using the Aurora 64b66b link-layer protocol.

Used communication protocol setting The I/O function uses the Aurora 64B/66B protocol with the following settings:

- Transceiver: GTH
- Line Rate: 10.3125 Gbps
- Dataflow Mode: Duplex
- Interface: Framing
- Flow Control: NFC
- USER K: off
- Little Endian Support: off
- CRC: off

For more information on the protocol, refer to https://www.xilinx.com/support/documentation/ip_documentation/aurora_64b66b/v12_0/pg074-aurora-64b66b.pdf.

Block display If you select an MGT Out channel from the channel list, the block display changes. The outport **Sim Data** is displayed optionally.



I/O characteristics The following table describes the ports of the block for the MGT module:

Port	Description
Input	
Data	Writes a 128-bit data value to the MGT communication bus. User data transfer rate: Max. 10.3125 Gbit/s, limited by the MGT module. MGT latency[1]: ▪ With maximum data rate: Max. 6.272 µs, typ. 6.192 µs

Port	Description
	The latency increases, because the TX-FIFO buffer becomes full when the data stream fills the buffer with 16 Gbit/s (128 bits at 125 MHz). ▪ Single words: Max. 472 ns, typ. 384 ns If you implement inter-FPGA communication via MGT modules, clock drifts can result in additional latencies. Refer to Implementing Inter-FPGA Communication via MGT Modules (RTI FPGA Programming Blockset Guide 🕮). Data type: UFix_128_0 Data width: 1
Enable	Enables the write access to the MGT communication bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: No write access. ▪ 1: The **Data** port value of the current clock cycle is written on the MGT communication bus.
Output	
Ready	Outputs a flag that indicates that the MGT module is ready to write new data on the MGT communication bus. Data type: UFix_1_0 Data width: 1 Values: ▪ 0: The MGT channel is busy. ▪ 1: New data values can be written to the MGT communication bus.
Sim Data	Simulates a data exchange via the MGT communication bus including fixed-point to floating-point data conversion. The converting can lead to inaccuracies. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

[1] Latency between sender and receiver via an optical loopback.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

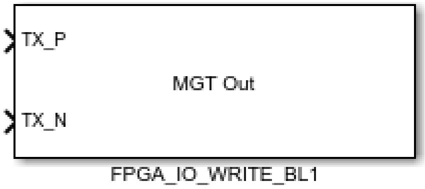
Aurora 64b66b 128 Bit Out settings

Only common dialog settings. Refer to Common settings on page 410.

MGT Out description

Purpose To provide the information about the connection between the GTH transceivers and the MGT module. The information is required for customer-specific protocol blocks that configure the GTH transceivers. A GTH transceiver is a configurable transceiver of the XILINX UltraScale FPGA architecture.

Block display If you select an MGT Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the MGT module:

Port	Description
Input	
TX_P TX_N	Writes data to the MGT module. The ports represent the differential signals of the GTH transceivers that are connected to the MGT module. The ports must be connected to a block that provides the configuration for the GTH transceivers. Data type: UFix_4_0 Data width: 1 Each bit represents the input for one MGT channel. Update rate: MGT reference clock frequency The **MGT reference clock frequency** parameter of the **MGT In** block lets you specify the reference frequency that is used to generate the MGT clock frequency. Refer to MGT In description on page 406.

I/O mapping The MGT communication bus uses the MPO connector of the FPGA Base Board.

Demo model For implementing an MGT communication with a customized protocol, refer to Modeling MGT Communication Using a Customized Protocol (RTI FPGA Programming Blockset Guide 🕮).

MGT Out settings None

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........415
FPGA_IO_WRITE_BL........56
FPGA_IO_WRITE_BL (MGT Out Settings)........409

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard DS660X_MGT framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (MGT Out Settings).. 409

RTI Block Settings for the Inter-FPGA Interface Framework

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks together with at least one *Inter-FPGA Interface* framework:

- *DS2655 (7K160) FPGA Base Board* framework
- *DS2655 (7K410) FPGA Base Board* framework
- *DS6601 (KU035) FPGA Base Board* framework
- *DS6602 (KU15P) FPGA Base Board* framework

Where to go from here

Information in this section

Information in other sections

Details on implementing an inter-FPGA communication bus:

Modeling Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 📖)
An inter-FPGA communication bus lets you exchange data directly between FPGA boards.

Method to select an I/O module slot of a SCALEXIO FPGA base board to be used as inter-FPGA interface:

How to Specify I/O Module Slots of SCALEXIO FPGA Base Boards as Inter-FPGA Interfaces (RTI FPGA Programming Blockset Guide 📖)
You have to specify the inter-FPGA connector only if you use the inter-FPGA communication on SCALEXIO FPGA base boards.

Other frameworks that provide access to the FPGA functionality of a SCALEXIO system:

RTI Block Settings for the DS2655 FPGA Base Board Framework............121
The block dialogs provide hardware-specific settings after you load one of the DS2655 FPGA Base Board frameworks.

RTI Block Settings for the DS6601 FPGA Base Board Framework............169
The block dialogs provide hardware-specific settings after you load the DS6601 (KU035) FPGA Base Board framework.

RTI Block Settings for the DS6602 FPGA Base Board Framework............219
The block dialogs provide hardware-specific settings after you load the DS6602 (KU15P) FPGA Base Board framework.

RTI Block Settings for the DS2655M1 I/O Module Framework................279
The block dialogs provide hardware-specific settings after you load the DS2655M1 I/O Module framework.

RTI Block Settings for the DS2655M2 I/O Module Framework................307
The block dialogs provide hardware-specific settings after you load the DS2655M2 I/O Module framework.

FPGA_IO_READ_BL (Inter-FPGA Interface Settings)

Purpose

To configure read access to the inter-FPGA communication bus when using the *Inter-FPGA Interface* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_IO_READ_BL)..419
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_READ_BL)..424
The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Description Page (FPGA_IO_READ_BL)..425
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)..54
To implement read access to a physical input channel in the FPGA model.

Unit Page (FPGA_IO_READ_BL)..55
To specify the I/O type and channel to be used for read access.

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose	To specify relevant settings for the selected I/O function.

NOTICE

The improper assembly of inter-FPGA communication buses will damage the FPGA boards

For inter-FPGA communication buses, special inter-FPGA communication cables must be used. Other cables, such as the cables used for connecting the I/O modules, will damage the FPGA boards. Furthermore, special rules for attaching the FPGA boards must be observed to ensure proper bus communication.

- Use the **SCLX_INT_FPGA_CAB1** inter-FPGA cables and observe the enclosed documentation for assembling.
- Do not connect FPGA boards via inter-FPGA cables if the FPGA boards are connected to different processors via IOCNET.

Description

The framework provides the I/O type *Other*_<IO module slot>, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- I-FPGA In 1 ... 8 [Mod: <1 ... 5>]

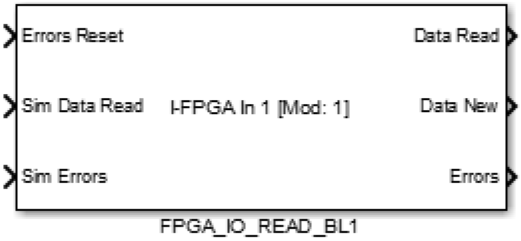
Common settings

None

I-FPGA In description

Purpose To provide read access to the inter-FPGA communication bus with bus synchronization.

Block display If you select an I-FPGA In channel from the channel list, the block display changes. Except for the outport **Data Read**, all ports are displayed optionally.



The module number **Mod: <x>** depends on the used I/O module slot for inter-FPGA communication.

I/O characteristics The following table describes the ports of the block for the inter-FPGA communication channels.

Port	Description
Input	
Errors Reset	Resets the **Errors** output. The counter of the **Errors** outport possibly increases until the corresponding **I-FPGA Out** function starts working. Available only if **Bit mode** is disabled on the **Parameters** page. Data type: UFix_1_0 Data width: 1 Value range: 0 ... 1
Sim Data Read	Simulates the data to be read from inter-FPGA communication bus. Available only if the **Enable simulation Data Read port** is set on the Parameters page. Data type: Double Data width: 1 Value range: 0 … 2^{27}-1
Sim Errors	Simulates the error counter. Available only if **Bit mode** is disabled on the **Parameters** page.
Output	
Data Read	Reads data from the inter-FPGA communication bus. I-FPGA In supports a *synchronous* data exchange. The maximum number of 27 bits is available only for one bus. If you have specified subbuses (up to 8), the maximum number of bits depends on their configurations. Bits that exceed the configured bus width are discarded. Data type: UFix_27_0 Data width: 1 Value range: 0 … 2^{27}-1
Data New	Indicates whether new data was written to the **Data Read** register. If the register contains new values, the flag changes from 0 to 1 for one clock cycle. If the transmission failed, the error counter increases. Available only if **Bit mode** is disabled on the **Parameters** page. Data type: UFix_1_0 Data width: 1

Port	Description
	▪ 0: No new data available in the **Data Read** register. Either the transmission is not yet finished, or the transmission failed (see **Errors** outport). ▪ 1: New data available in the **Data Read** register.
Errors	Provides the number of transmission errors. The counter is reset only at FPGA application start or if it reset by the **Errors Reset** port. If the range is exceeded, the counter restarts from 0. Available only if **Bit mode** is disabled on the **Parameters** page. Data type: UFix_32_0 Data width: 1 Value range: 0 … 2^{32}-1

I/O mapping No external connection to the I/O connector of the board. The SCALEXIO FPGA base board uses its I/O module slots inside the SCALEXIO system for inter-FPGA communication.

I-FPGA In settings

The following settings on the **Parameters** page are specific to the **I-FPGA In** I/O function.

Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 27.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For more information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Note

If you send and receive data with the same inter-FPGA interface, you have to consider limitations on the bit ranges for the subbuses. Refer to How to Determine the Bit Ranges for Inter-FPGA Subbuses Between SCALEXIO FPGA Base Boards (RTI FPGA Programming Blockset Guide 🕮).

Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 27.

The range of the end bit is automatically adapted to the specified start bit. The end bit must not be less than the corresponding start bit.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For more information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Note

If you send and receive data with the same inter-FPGA interface, you have to consider limitations on the bit ranges for the subbuses. Refer to How to Determine the Bit Ranges for Inter-FPGA Subbuses Between SCALEXIO FPGA Base Boards (RTI FPGA Programming Blockset Guide).

Expert mode Lets you enable the expert mode, where you can modify the bit length, the clock and the filter depth.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.

The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Bit length Displays the default bit length of 6 clock cycles.

If you have enabled **Expert mode** and disabled **Bit mode**, you can specify a bit length in the range 3 … 128 cycles. A reference value for the bit length can be calculated by $2 + 2 \cdot \mathtt{FilterDepth}_{\mathtt{inport}}$. If $\mathtt{FilterDepth}_{\mathtt{inport}}$ is 0, the minimum bit length is 3.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.

The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Clock Displays the default I-FPGA clock of 125 MHz. This value only affects the inter-FPGA communication and not the clock of the FPGA.

If you have enabled **Expert mode** and disabled **Bit mode**, you can specify an I-FPGA clock of 200 MHz or 250 MHz.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Filter depth Displays the default filter depth of 2 clock cycles.

At a filter depth > 0, a spike filter with the specified length is applied to the transmission to reduce transmission errors.

If you have enabled **Expert mode** and disabled **Bit mode**, you can specify a filter depth in the range 0 … 32 clock cycles.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Add internal pipeline register to relax timing Lets you enable an additional internal pipeline to relax timing especially for 250 MHz communication if otherwise a FPGA build process were not possible.

Configurable only if the **Bit mode** is disabled.

Bit mode Lets you enable the bit mode to read data with a lower latency (2 clock cycles), but the bits are not synchronous.

This setting is selectable only if you have enabled **Expert mode**.

Note

You should enable this mode only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.

Enable simulation Data Read port Lets you enable an inport for offline simulation data. The **Sim Data Read** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model instead of the inter-FPGA communication bus.

Enable simulation Errors port Lets you enable an inport for offline simulation data. The Sim Errors inport is added to the block to to provide the number of simulated transmission errors.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_IO_READ_BL)	425
FPGA_IO_READ_BL	50
FPGA_IO_READ_BL (Inter-FPGA Interface Settings)	418

Scaling Page (FPGA_IO_READ_BL)

Description

The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 🕮)

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *Inter-FPGA Interface* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL (Inter-FPGA Interface Settings)

Purpose

To configure write access to the inter-FPGA communication bus when using the *Inter-FPGA Interface* framework.

Where to go from here

Information in this section

Parameters Page (FPGA_IO_WRITE_BL)........ 426
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_WRITE_BL)........ 430
The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Description Page (FPGA_IO_WRITE_BL)........ 431
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Block Description (FPGA_IO_WRITE_BL)..60
To implement write access to a physical output channel in the FPGA model.

Unit Page (FPGA_IO_WRITE_BL)..61
To specify the I/O type and channel to be used for write access.

Related RTI blocks

FPGA_IO_READ_BL..50
To provide read access to an external device via a physical input channel.

FPGA_IO_READ_BL (Inter-FPGA Interface Settings)...................................418
To configure read access to the inter-FPGA communication bus when using the Inter-FPGA Interface framework.

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

NOTICE

The improper assembly of inter-FPGA communication buses will damage the FPGA boards

For inter-FPGA communication buses, special inter-FPGA communication cables must be used. Other cables, such as the cables used for connecting the I/O modules, will damage the FPGA boards. Furthermore, special rules for attaching the FPGA boards must be observed to ensure proper bus communication.

- Use the **SCLX_INT_FPGA_CAB1** inter-FPGA cables and observe the enclosed documentation for assembling.
- Do not connect FPGA boards via inter-FPGA cables if the FPGA boards are connected to different processors via IOCNET.

Description

The framework provides the I/O type *Other*_<IO module slot>, which you can select on the **Unit** page of the block. The number of the available physical connections determines the I/O functions that you can select:

- I-FPGA Out 1 ... 8 [Mod: <1 ... 5>]

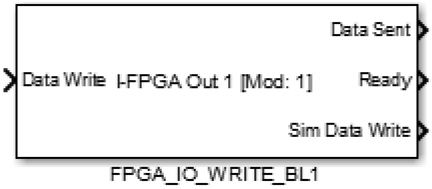
Common settings

None

I-FPGA Out description

Purpose To provide write access to the inter-FPGA communication bus with bus synchronization.

Block display If you select an I-FPGA Out channel from the channel list, the block display changes. Except for the inport **Data Write**, all ports are displayed optionally.



The module number **Mod: <x>** depends on the used I/O module slot for inter-FPGA communication.

I/O characteristics The following table describes the ports of the block for the inter-FPGA communication channels:

Port	Description
Input	
Data Write	Outputs data to the inter-FPGA communication bus. The maximum number of bits in the data buffer depends on the number of blocks connected to the bus and the blocks settings. Bits which exceed the configured bus width are discarded. Data type: UFix_27_0 Data width: 1 Value range: 0 … 2^{27}-1
Output	
Data Sent	Provides the last transmitted value. Available only if **Bit mode** is disabled on the **Parameters** page. Data type: UFix_27_0 Data width: 1 Value range: 0 … 2^{27}-1
Ready	Indicates the clock cycle in which the data is sampled for transmission. The port is high for one clock cycle with the periodicity of the bit length. Available only if **Bit mode** is disabled on the **Parameters** page. Data width: 1
Sim Data Write	Simulates the data to be written to the inter-FPGA communication bus. Available only if **Enable simulation Data Write port** is set on the **Parameters** page. Data type: Double Data width: 1

I/O mapping No external connection to the I/O connector of the board. The SCALEXIO FPGA base board uses its I/O module slots inside the SCALEXIO system for inter-FPGA communication.

I-FPGA Out settings

> ***NOTICE***
>
> **An incorrect configuration might damage the electrical interface.**
> If you configure both ends of an inter-FPGA connection bus to write on the bus, the connection results in a short circuit. This short circuit might damage the electrical interface of the used I/O module slots. In multiprocessor applications, an incorrect configuration cannot be detected automatically to beware hardware damage.
>
> - Make sure that the counterpart interface on the other FPGA board uses the same **Startbit** and **Endbit** to read the data. Refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

The following settings on the **Parameters** page are specific to the **I-FPGA Out** I/O function.

Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 27.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For more information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

> **Note**
>
> If you send and receive data with the same inter-FPGA interface, you have to consider limitations on the bit ranges for the subbuses. Refer to How to Determine the Bit Ranges for Inter-FPGA Subbuses Between SCALEXIO FPGA Base Boards (RTI FPGA Programming Blockset Guide 🕮).

Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 27.

The range of the end bit is automatically adapted to the specified start bit. The end bit must not be less than the corresponding start bit.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For more information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Note

If you send and receive data with the same inter-FPGA interface, you have to consider limitations on the bit ranges for the subbuses. Refer to How to Determine the Bit Ranges for Inter-FPGA Subbuses Between SCALEXIO FPGA Base Boards (RTI FPGA Programming Blockset Guide 📖).

Expert mode Lets you enable the expert mode, where you can modify the bit length, the clock and the filter depth.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Bit length Displays the default bit length of 6 clock cycles.

If you have enabled **Expert mode** and disabled **Bit mode**, you can specify a bit length in the range 3 ... 128 cycles. A reference value for the bit length can be calculated by $2 + 2 \cdot \mathtt{FilterDepth}_{inport}$. If $\mathtt{FilterDepth}_{inport}$ is 0, the minimum bit length is 3.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Clock Displays the default I-FPGA clock of 125 MHz. This value only affects the inter-FPGA communication and not the clock of the FPGA.

If you have enabled **Expert mode** and disabled **Bit mode**, you can specify an I-FPGA clock of 200 MHz or 250 MHz.

> **Note**
>
> You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
> The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Bit mode Lets you enable the bit mode to write data with a lower latency (2 clock cycles), but the bits are not synchronous.

This setting is selectable only if you have enabled **Expert mode**.

> **Note**
>
> You should enable this mode only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.

Enable simulation Data Write port The **Sim Data Write** simulation port is available in the block only if you enable it.

Related topics

Basics

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 📖)

References

Description Page (FPGA_IO_WRITE_BL)........431
FPGA_IO_WRITE_BL........56
FPGA_IO_WRITE_BL (Inter-FPGA Interface Settings)........425

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Frameworks with scaling support

The frameworks of the I/O modules support FPGA scaling:

- DS2655M1 I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 287
  - Scaling Page (FPGA_IO_WRITE_BL) on page 302
- DS2655M2 Digital I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 318
  - Scaling Page (FPGA_IO_WRITE_BL) on page 341
- DS6651 Multi-I/O Module
  - Scaling Page (FPGA_IO_READ_BL) on page 360
  - Scaling Page (FPGA_IO_WRITE_BL) on page 396

Related topics

Basics

Basics on FPGA Test Access and Scaling (RTI FPGA Programming Blockset Guide 📖)

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the standard *Inter-FPGA Interface* framework is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (Inter-FPGA Interface Settings).. 425
Parameters Page (FPGA_IO_WRITE_BL)... 426

RTI Block Settings for the DS5203 with onboard I/O Frameworks

---

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks:

- *DS5203 (7K325) with onboard I/O*
- *DS5203 (7K410) with onboard I/O*

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (DS5203 with Onboard I/O Settings)........................433
To configure read access to PHS-bus data in the FPGA model when using the onboard I/O frameworks of DS5203.

FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings)......................441
To configure write access to PHS-bus data in the FPGA model when using the onboard I/O frameworks of DS5203.

FPGA_IO_READ_BL (DS5203 with Onboard I/O Settings)................................450
To configure read access to analog and digital input signals in the FPGA model when using the onboard I/O frameworks of DS5203.

FPGA_IO_WRITE_BL (DS5203 with Onboard I/O Settings)...............................461
To configure write access to analog and digital output signals in the FPGA model when using the onboard I/O frameworks of DS5203.

FPGA_INT_BL (DS5203 with Onboard I/O Settings)...471
To configure the FPGA interrupt channel when using the onboard I/O frameworks of DS5203.

Information in other sections

Details on implementing an inter-FPGA communication bus:

Modeling Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮)
An inter-FPGA communication bus lets you exchange data directly between FPGA boards.

Other frameworks that provide access to the FPGA functionality of a PHS-bus-based system:

RTI Block Settings for the DS5203 with Multi-I/O Module (DS5203M1) Frameworks...474
The block dialogs provide hardware-specific settings after you load one of the DS5203 with Multi-I/O Module (DS5203M1) frameworks.

FPGA_XDATA_READ_BL (DS5203 with Onboard I/O Settings)

Purpose

To configure read access to PHS-bus data in the FPGA model when using one of the following frameworks:

- *DS5203 (7K325) with onboard I/O*
- *DS5203 (7K410) with onboard I/O*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)..434
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)..441
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To write data from the processor model to the FPGA model via the board-specific bus.

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings)..441
To configure write access to PHS-bus data in the FPGA model when using the onboard I/O frameworks of DS5203.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The DS5203 with onboard I/O frameworks provide the following access types, which you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from a PHS-bus register. There are 128 available registers with a data width of 32 bits each, and 128 registers with a data width of 64 bits each. The values are transmitted element-wise. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from a PHS-bus buffer. There are 32 available buffers that provide elements with a data width of 32 bits each, and 32 buffers that provide elements with a data width of 64 bits each.

  Each buffer has a variable buffer size of 1 ... 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position, respectively.
- floating-point

  The values of the **Data** outport are in floating-point format.

  The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

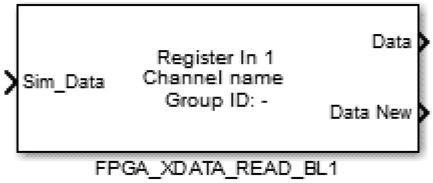
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates a PHS-bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from a PHS-bus register. The data format depends on the related dialog settings.

Port	Description
	▪ Fixed-point format UFix_32_<Binary point position> or Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

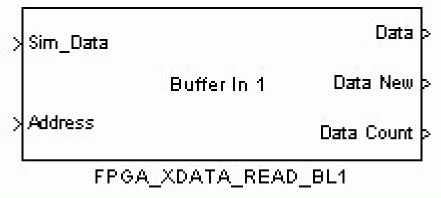
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 434.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are read from the PHS bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates a PHS-bus data exchange including floating-point to fixed-point data conversion.

Port	Description
	Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the Data outport according to the specified address. The maximum port range depends on the specified buffer size in the Parameters page. The valid port range depends on the number of elements currently in the buffer (see Data Count outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from a PHS-bus buffer. The data format depends on the related dialog settings. ▪ Fixed-point format UFix_32_<Binary point position> or Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the Address inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

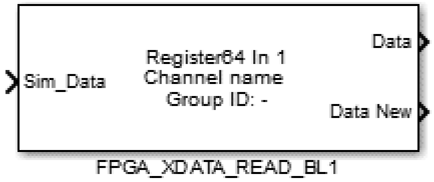
The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 434.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates a PHS-bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from a PHS-bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position> or Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a

Port	Description
	new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

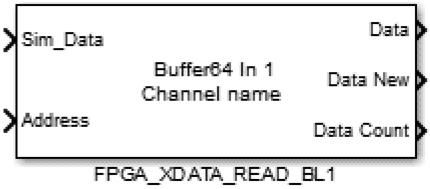
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 434.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the local bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer64 In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates the PHS-bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size on the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater

Port	Description
	than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from the PHS-bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position> or Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 434.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the onboard I/O frameworks of DS5203 (7K325) and DS5203 (7K410) is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings)

Purpose

To configure write access to PHS-bus data in the FPGA model after you load one of the following frameworks:

- DS5203 (7K325) with onboard I/O
- DS5203 (7K410) with onboard I/O

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The DS5203 with onboard I/O frameworks provide the following access types, which you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to a PHS-bus register. There are 128 available registers with a data width of 32 bits each, and 128 registers with a data width of 64 bits each. The values are transmitted element-wise. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to a PHS-bus buffer. There are 32 available buffers that provide elements with a data width of 32 bits each and 32 buffers that provide elements with a data width of 64 bits each.

  Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position, respectively.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

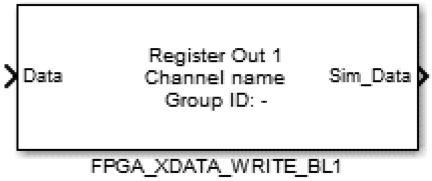
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The Sim_Data outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data and Sim_Status. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to a PHS-bus register. The data format depends on the related dialog settings. ▪ Fixed-point format UFix_32_<Binary point position> or Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates a PHS-bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1

Note

The range can be exceeded for the Data inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

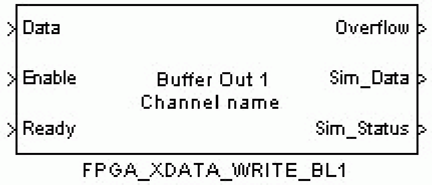
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 443.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. Registers which you specified with the same group ID are sampled simultaneously within the FPGA application. The values form therefore a consistent data group that is written to the PHS bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to a PHS-bus buffer. The data format depends on the related dialog settings. ▪ Fixed-point format UFix_32_<Binary point position> or Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the following clock cycle. The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.

Port	Description
Output	
Sim_Data	Simulates a PHS-bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the Parameters page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1, if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data is lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 443.

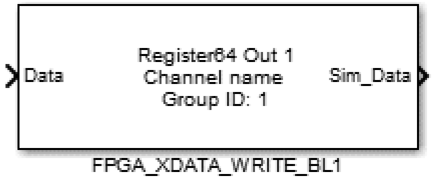
Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to a PHS-bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position> or Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates a PHS-bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

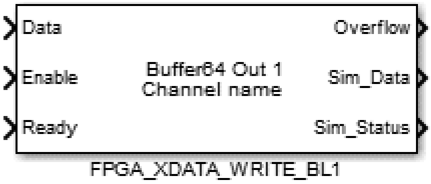
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 443.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously within the FPGA application. The values therefore form a consistent data group that is written to the PHS bus.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to a PHS-bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position> or Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via PHS bus in the following clock cycle. The ready flag must be set no later than the last data value. Otherwise the buffer switches twice.
Output	
Sim_Data	Simulates a PHS-bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.

Port	Description
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the Parameters page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 443.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enable the simulation data port. The **Sim_Status** outport is added to the block.

Related topics

References

- Description Page (FPGA_XDATA_WRITE_BL)........449
- FPGA_XDATA_WRITE_BL........45
- FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings)........441

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the onboard I/O frameworks of DS5203 (7K325) and DS5203 (7K410) is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_XDATA_WRITE_BL..45
FPGA_XDATA_WRITE_BL (DS5203 With Onboard I/O Settings)........................441
Parameters Page (FPGA_XDATA_WRITE_BL)..442

FPGA_IO_READ_BL (DS5203 with Onboard I/O Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model after you load one of the following frameworks:

- *DS5203 (7K325) with onboard I/O*
- *DS5203 (7K410) with onboard I/O*

Where to go from here

Information in this section

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)..54
To implement read access to a physical input channel in the FPGA model.

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Analog*, *Digital* and *Other*, which you can select on the Unit page of the block. The number of the available physical connections on the DS5203 FPGA Board determines the I/O functions that you can select:

- ADC 1 ... ADC 6
- Digital In 1 ... Digital In 16
- Status In
- APU Slave
- I-FPGA Slave 1 … 8

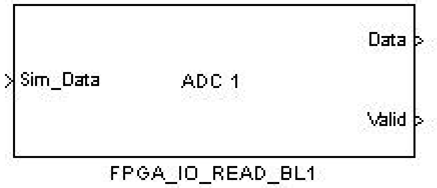
Common settings

The following settings on the Parameters page are common to the I/O functions that you can select on the Unit page of the FPGA_IO_READ_BL block, except for the Status In function.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

ADC description

Block display If you select an ADC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–5 V ... +5 V	Depending on the specified Scaling parameter, the output port range is: ▪ –5000 mV … +5000 mV or ▪ –8192 … +8191
–30 V ... +30 V	Depending on the specified Scaling parameter, the output port range is: ▪ –30000 mV … +30000 mV or ▪ –8192 … +8191

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Input voltage range: ▪ –5 V ... +5 V ▪ –30 V ... +30 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 Range: ▪ –5000 mV ... +5000 mV ▪ –30000 mV ... +30000 mV or ▪ –8192 … +8191 Update rate: 10 Msps

Port	Description
Valid	Represents the current status of the data output. Data type: UFix_1_0 ▪ 0: Converted value is out of range. ▪ 1: Converted value is within the specified input range.

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, it is treated as an invalid value:

- **Data** outport is saturated to minimum or maximum range value.
- **Valid** outport is set to 0.

I/O mapping The following I/O mapping is relevant if you use one of the DS5203 with onboard I/O frameworks for analog input channels.

Outport	Channel	Connector Pin	Signal
Data	Ch 1	P1 1	$\overline{\text{ADC1}}$
		P1 34	ADC1
	Ch 2	P1 18	$\overline{\text{ADC2}}$
		P1 2	ADC2
	Ch 3	P1 35	$\overline{\text{ADC3}}$
		P1 19	ADC3
	Ch 4	P1 3	$\overline{\text{ADC4}}$
		P1 36	ADC4
	Ch 5	P1 20	$\overline{\text{ADC5}}$
		P1 4	ADC5
	Ch 6	P1 37	$\overline{\text{ADC6}}$
		P1 21	ADC6

ADC settings

The following settings on the **Parameters** page are specific to the **ADC** I/O function. For common dialog settings, refer to Common settings on page 451.

Input range Lets you select the input voltage range for the analog input channel.

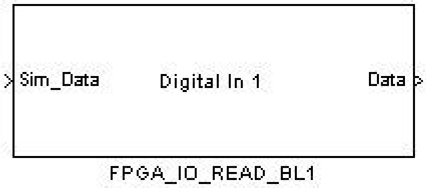
- –5 V ... +5 V
- –30 V ... +30 V

The selected input range is also valid for the enabled simulation data inport.

Scaling Lets you select the scaling of the output data. If you select **mV**, the valid output port range corresponds to the specified input range in mV (-5000 … +5000 mV or -30000 … +30000 mV). If you select the unscaled **Bit** value, the valid output port range is -8192 … +8191, independently from the specified input range.

Digital In description

Block display If you select a Digital In channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Threshold level: ▪ Up to board revision DS5203-03: 1.0 V ... 5.0 V (in steps of 0.1 V) ▪ Since board revision DS5203-04: 1.0 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage. Update rate: 100 MHz

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

Note

Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
To use the maximum input voltage range, you have to use a digital channel only as the input.

I/O mapping The following I/O mapping is relevant if you use one of the DS5203 with onboard I/O frameworks for digital input channels.

Outport	Channel	Connector Pin	Signal
Data	Ch 1	P1 22	DIG_IO1
	Ch 2	P1 6	DIG_IO2
	Ch 3	P1 23	DIG_IO3
	Ch 4	P1 7	DIG_IO4
	Ch 5	P1 24	DIG_IO5
	Ch 6	P1 8	DIG_IO6
	Ch 7	P1 25	DIG_IO7
	Ch 8	P1 9	DIG_IO8
	Ch 9	P1 26	DIG_IO9
	Ch 10	P1 10	DIG_IO10
	Ch 11	P1 27	DIG_IO11
	Ch 12	P1 11	DIG_IO12
	Ch 13	P1 28	DIG_IO13
	Ch 14	P1 12	DIG_IO14
	Ch 15	P1 29	DIG_IO15
	Ch 16	P1 13	DIG_IO16

You can use the same digital channel for input and output signals.

Digital In settings

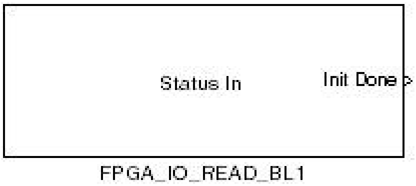
The following settings on the **Parameters** page are specific to the **Digital In** I/O function. For common dialog settings, refer to Common settings on page 451.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1,000 mV ... 7,500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

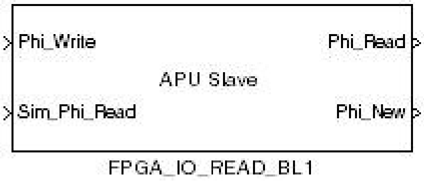
I/O mapping No external connection.

Status In settings

None

APU Slave description

Block display If you select the APU Slave channel from the channel list, the block display changes. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Phi_Read	Simulates the input value for the time-base connector. Available only if the **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Output	
Phi_Read	Reads a value from the time-base connector. Data type: UFix_16_0

Port	Description
	Data width: 1 Angle value range: 0 … 65535 (0 … 720°) Time-base clock cycle: 250 ns
Phi_New	Indicates that new data was written to the Phi_Read register. Data type: UFix_1_0 Data width: 1 Value range: 0, 1 If the flag changes from 0 to 1, the requested register contains new values. The flag is set to 1 for only one clock cycle.

I/O mapping No external connection to the I/O connector of the board.

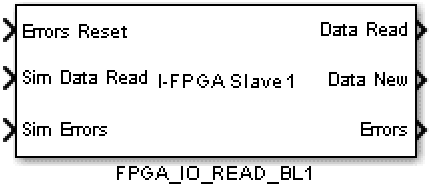
For further information on the time-base connector, refer to Board Overview (PHS Bus System Hardware Reference).

APU Slave settings

Only common dialog settings. Refer to Common settings on page 451.

I-FPGA Slave description

Block display If you select an I-FPGA Slave channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for the inter-FPGA communication channels:

Port	Description
Input	
Errors Reset	Resets the **Errors** output. The I-FPGA Slaves Errors counter possibly increases until the FPGA Board with the corresponding I-FPGA Master starts working. Available only if the **Enable simulation Data Read port** is set on the Parameters page. Data type: U_Fix1_0 Value range: 0 ... 1
Sim Data Read	Simulates the data to be read from the inter-FPGA communication bus. Available only if the **Enable simulation Data Read port** is set on the Parameters page. Data type: Double Data width: 1 Value range: 0 … 2^{31}-1
Sim Errors	Simulates the error counter.

Port	Description
Output	
Data Read	Reads data from the inter-FPGA communication bus. The maximum number of 31 bits is only available for one bus. If you have specified subbuses (up to 8), the maximum number of bits depends on their configurations. Bits which exceed the configured bus width are discarded. Data type: U_Fix31_0 Data width: 1 Value range: 0 … 2^{31}-1
Data New	Indicates whether new data was written to the **Data Read** register. If the register contains new values, the flag changes from 0 to 1 for one clock cycle. If the transmission failed, the error counter increases. Data type: U_Fix1_0 Data width: 1 ▪ 0: No new data available in the **Data Read** register. Either the transmission is not yet finished, or the transmission failed (see **Errors** outport). ▪ 1: New data available in the **Data Read** register.
Errors	Provides the number of transmission errors. The counter is reset only at FPGA application start. If the range is exceeded, the counter restarts from 0. Data type: U_Fix32_0 Data width: 1 Value range: 0 … 2^{31}-1

I/O mapping No external connection to the I/O connector of the board.

For further information on the inter-FPGA communication connector, refer to Board Overview (PHS Bus System Hardware Reference 🕮).

I-FPGA Slave settings

The following settings on the **Parameters** page are specific to the I_FPGA Slave I/O function.

Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 31.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For further information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 31.

The range of the end bit is automatically adapted to the specified start bit. The end bit must not be less than the corresponding start bit.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For further information, refer to

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 📖).

Expert mode Lets you enable the expert mode, where you can modify the bit length, the clock and the filter depth.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Bit length Displays the default bit length of 6 clock cycles.

If you have enabled **Expert mode**, you can specify a bit length in the range 3 … 128 cycles. A reference value for the bit length can be calculated by $\texttt{2 + 2} \cdot \texttt{FilterDepth}_{\texttt{inport}}$. If $\texttt{FilterDepth}_{\texttt{inport}}$ is 0, the minimum bit length is 3.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Clock Displays the default I-FPGA clock of 100 MHz. This value only affects the inter-FPGA communication and not the clock of the FPGA.

If you have enabled the **Expert mode**, you can specify an I-FPGA clock in the range 100 … 300.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Filter depth Displays the default filter depth of 2 clock cycles.

At a filter depth > 0, a spike filter with the specified length is applied to the transmission to reduce transmission errors.

If you have enabled Expert mode, you can specify a filter depth in the range 0 … 32 clock cycles.

Note

You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Add internal pipeline register to relax timing Lets you enable an additional internal pipeline to relax timing especially for 300 MHz communication if otherwise a FPGA build process were not possible.

Enable simulation Data Read port Lets you enable an inport for offline simulation data. The Sim Data Read inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model instead of the inter-FPGA communication bus.

Enable simulation Errors port Lets you enable an inport for offline simulation data. The Sim Errors inport is added to the block to to provide the number of simulated transmission errors.

Related topics

References

Scaling Page (FPGA_IO_READ_BL)

Description

The Scaling page is empty because the I/O functions of this framework do not support FPGA scaling.

Related topics

References

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O function that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the DS5203 onboard I/O frameworks is included in this user documentation. The description of the I/O functions of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (DS5203 with Onboard I/O Settings)........................ 450
Parameters Page (FPGA_IO_READ_BL).. 451

FPGA_IO_WRITE_BL (DS5203 with Onboard I/O Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model after you load one of the following frameworks:

- *DS5203 (7K325) with onboard I/O*
- *DS5203 (7K410) with onboard I/O*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the I/O types *Analog*, *Digital* and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS5203 FPGA Board determines the I/O functions that you can select:

- DAC 1 ... DAC 6
- Digital Out 1 ... Digital Out 16
- LED Out
- APU Master
- I-FPGA Master 1 … 8

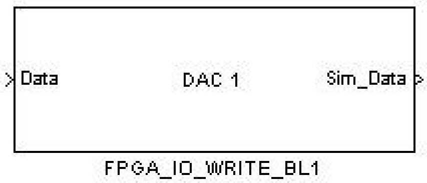
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out function.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

DAC description

Block display If you have select an DAC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog output voltage and the input of the block is:

Output Voltage Range	Simulink Input
–10 V ... +10 V	Depending on the specified Scaling parameter, the input port range is: ▪ –10000 mV … +10000 mV or ▪ –8192 … +8191

The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data type: Fix_15_0 Output voltage range: ▪ –10000 mV ... +10000 mV or ▪ –8192 … +8191 The range can be exceeded, and saturation is performed to a minimum or maximum value. Hardware update rate: 10 Msps If the values are updated at a higher FPGA model rate, intermediate values are not updated by the DAC.
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Output voltage range: -10.0 V ... +10.0 V

I/O mapping The following I/O mapping is relevant if you use one of the DS5203 with onboard I/O frameworks for analog output channels.

Inport	Channel	Connector Pin	Signal
Data	Ch 1	P1 30	$\overline{\text{DAC1}}$
		P1 14	DAC1
	Ch 2	P1 47	$\overline{\text{DAC2}}$
		P1 31	DAC2
	Ch 3	P1 15	$\overline{\text{DAC3}}$
		P1 48	DAC3
	Ch 4	P1 32	$\overline{\text{DAC4}}$
		P1 16	DAC4
	Ch 5	P1 49	$\overline{\text{DAC5}}$
		P1 33	DAC5
	Ch 6	P1 17	$\overline{\text{DAC6}}$
		P1 50	DAC6

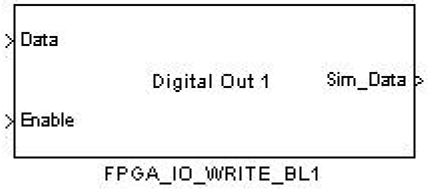
DAC settings

The following settings on the **Parameters** page are specific to the **DAC** I/O function. For common dialog settings, refer to Common settings on page 462.

Scaling Lets you select the scaling of the input data. If you select **mV**, the valid input port range is -10000 mV … +10000 mV. If you select the unscaled **Bit** value, the valid input port range is –8192 … +8191 (14-bit D/A converter).

Digital Out description

Block display If you select a Digital Out channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high supply voltage. The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: 100 MHz

Port	Description
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the Parameters page. Output voltage: 0 V ... 3.3 V or 0 V ... 5 V Update rate: 100 MHz

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

I/O mapping The following I/O mapping is relevant if you use one of the DS5203 with onboard I/O frameworks for digital output channels.

Inport	Channel	Connector Pin	Signal
Data	Ch 1	P1 22	DIG_IO1
	Ch 2	P1 6	DIG_IO2
	Ch 3	P1 23	DIG_IO3
	Ch 4	P1 7	DIG_IO4
	Ch 5	P1 24	DIG_IO5
	Ch 6	P1 8	DIG_IO6
	Ch 7	P1 25	DIG_IO7
	Ch 8	P1 9	DIG_IO8
	Ch 9	P1 26	DIG_IO9
	Ch 10	P1 10	DIG_IO10
	Ch 11	P1 27	DIG_IO11
	Ch 12	P1 11	DIG_IO12
	Ch 13	P1 28	DIG_IO13
	Ch 14	P1 12	DIG_IO14
	Ch 15	P1 29	DIG_IO15
	Ch 16	P1 13	DIG_IO16

You can use the same digital channel for input and output signals.

Digital Out settings

The following settings on the **Parameters** page are specific to the **Digital Out** I/O function. For common dialog settings, refer to Common settings on page 462.

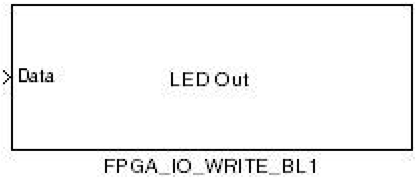
High supply Lets you select the voltage for the high side switch for all digital output channels.

Note

You can specify the high supply voltage value only globally for all digital output channels.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the LED on the board's bracket. Data type: UFix_1_0 ▪ 0: LED lights green. ▪ 1: LED lights orange.

If the value of the Data inport exceeds the specified data width, only the lowest bit is used (=1).

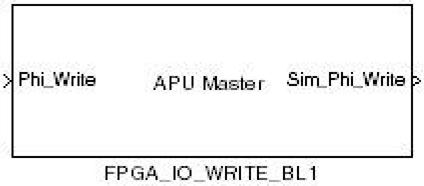
I/O mapping No external connection.

LED Out settings

None

APU Master description

Block display If you have select the APU Master channel from the channel list, the block display changes. The simulation port is displayed optionally.



Note

Only one APU master is allowed within a network of boards connected via the time-base connector.
If a DS2210 or DS2211 board is connected to your APU bus, you have to reconfigure the DS2210/DS2211 APU master to a DS2210/DS2211 APU slave. For further information, refer to DS2211APU_CRANK_Bx (DS2211 RTI Reference 📖).

I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Phi_Write	Outputs a value to the time-base connector. Data type: UFix_16_0 Data width: 1 Angle value range: 0 … 65535 (0 … 720°) Time-base clock cycle: 250 ns
Output	
Sim_Phi_Write	Simulates the output value of the time-base connector. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 The range can be exceeded, and saturation is performed to a minimum or maximum value.

I/O mapping No external connection to the I/O connector of the board.

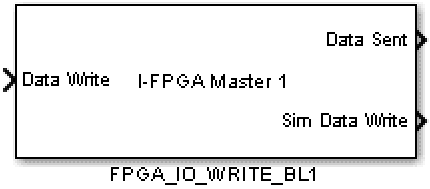
For further information on the time-base connector, refer to Board Overview (PHS Bus System Hardware Reference 📖).

APU Master settings

Only common dialog settings. Refer to Common settings on page 462.

I-FPGA Master description

Block display If you select an I-FPGA Master channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for the inter-FPGA communication channels:

Port	Description
Input	
Data Write	Outputs data to the inter-FPGA communication bus. The maximum number of bits in the data buffer depends on the number of blocks connected to the bus and the blocks settings. Bits which exceed the configured bus width are discarded. Data type: U_Fix31_0 Data width: 1 Value range: 0 … 2^{31}-1
Output	
Data Sent	Provides the last transmitted value. Data type: U_Fix31_0 Data width: 1 Value range: 0 … 2^{31}-1
Sim Data Write	Simulates the data to be written to the inter-FPGA communication bus. Available only if **Enable simulation Data Write port** is set on the Parameters page. Data type: Double Data width: 1

I/O mapping No external connection to the I/O connector of the board.

For further information on the inter-FPGA communication connector, refer to Board Overview (PHS Bus System Hardware Reference 🕮).

I-FPGA Master settings

The following settings on the **Parameters** page are specific to the **I-FPGA Master** I/O function.

Startbit Lets you specify the bit with which the transmission data starts in the range 0 … 31.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For further information, refer to Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Endbit Lets you specify the bit with which the transmission data ends in the range 0 … 31.

The range of the end bit is automatically adapted to the specified start bit. The end bit must not be less than the corresponding start bit.

If you have configured more than one inter-FPGA bus, you can access a specific subbus by specifying the related start and end bits. One bit has to be reserved for synchronization for each configured subbus. The maximum data width of a subbus is therefore `Endbit - Startbit`. For further information, refer to

Overview of Inter-FPGA Communication (RTI FPGA Programming Blockset Guide 🕮).

Expert mode Lets you enable the expert mode, where you can modify the bit length, the clock and the filter depth.

> **Note**
>
> You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
> The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Bit length Displays the default bit length of 6 clock cycles.

If you have enabled **Expert mode**, you can specify a bit length in the range 3 … 128 cycles. A reference value for the bit length can be calculated by $2 + 2 \cdot \texttt{FilterDepth}_{\texttt{inport}}$. If $\texttt{FilterDepth}_{\texttt{inport}}$ is 0, the minimum bit length is 3.

> **Note**
>
> You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
> The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Clock Displays the default I-FPGA clock of 100 MHz. This value only affects the inter-FPGA communication and not the clock of the FPGA.

If you have enabled the **Expert mode**, you can specify an I-FPGA clock in the range 100 … 300.

> **Note**
>
> You should change these values only if you have enough experience of configuring buses and knowledge of checking the correctness of the configured transmission with regard to the observed signal integrity at the applicable temperature range.
> The default values for the bit length, clock, and filter depth have been tested by dSPACE.

Enable simulation Data Write port Lets you enable an outport for offline simulation data. The **Sim_Data_Write** outport is added to the block to provide data to the Simulink-based I/O environment model instead of the inter-FPGA communication bus.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)..470
FPGA_IO_WRITE_BL..56
FPGA_IO_WRITE_BL (DS5203 with Onboard I/O Settings)..461

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Related topics

References

FPGA Access Page (FPGA_SETUP_BL)..35

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O function that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the DS5203 onboard I/O frameworks is included in this user documentation. The description of the I/O functions of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_IO_WRITE_BL..56
FPGA_IO_WRITE_BL (DS5203 with Onboard I/O Settings)..461
Parameters Page (FPGA_IO_WRITE_BL)...462

FPGA_INT_BL (DS5203 with Onboard I/O Settings)

Purpose

To configure the FPGA interrupt channel after you load one of the following frameworks:

- *DS5203 (7K325) with onboard I/O*
- *DS5203 (7K410) with onboard I/O*

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL)...471
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)...472
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL)...66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

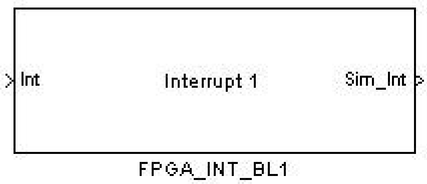
Description

The onboard I/O framework of DS5203 provides 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. This means, that an interrupt that was requested cannot be released, but it will be transferred to the processor. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	**Description**
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved. This means, that an interrupt that was requested cannot be released, but it will be transferred to the processor.
Output	
Sim_Int	Simulates an interrupt by performing a function call to enable a function call subsystem. Available only if Enable simulation port is set on the Parameters page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The Sim_Int outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)........472
FPGA_INT_BL........62
FPGA_INT_BL (DS5203 with Onboard I/O Settings)........471

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description	The **Description** page provides detailed information about the I/O function that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page. The description of the I/O function that is provided by the DS5203 onboard I/O frameworks is included in this user documentation. The description of the I/O functions of customized frameworks or mounted piggybacks will be available as separate documents.
Related topics	References FPGA_INT_BL........62 FPGA_INT_BL (DS5203 with Onboard I/O Settings)........471 Parameters Page (FPGA_INT_BL)........471

RTI Block Settings for the DS5203 with Multi-I/O Module (DS5203M1) Frameworks

Introduction

The block dialogs provide hardware-specific settings after you load one of the following frameworks:

- *DS5203 (7K325) with Multi-I/O Module (DS5203M1)*
- *DS5203 (7K410) with Multi-I/O Module (DS5203M1)*

Where to go from here

Information in this section

Information in other sections

Details on implementing an inter-FPGA communication bus:

Handcoding Inter-FPGA Communication (RTI FPGA Programming Blockset Handcode Interface Guide 📖)
The DS5203 FPGA Board and the SCALEXIO FPGA base boards support inter-FPGA communication.

Other frameworks that provide access to the FPGA functionality of a PHS-bus-based system:

RTI Block Settings for the DS5203 with onboard I/O Frameworks....432
The block dialogs provide hardware-specific settings after you load one of the DS5203 with onboard I/O frameworks.

FPGA_IO_READ_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model after you load one of the following frameworks:

- *DS5203 (7K325) with Multi-I/O Module (DS5203M1)*
- *DS5203 (7K410) with Multi-I/O Module (DS5203M1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the two I/O types *Analog* and *Digital*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS5203M1 Multi-I/O Module determines the I/O functions that you can select:

- ADC 1 (M1) ... ADC 6 (M1)
- Digital In 1 (M1) ... Digital In 16 (M1)

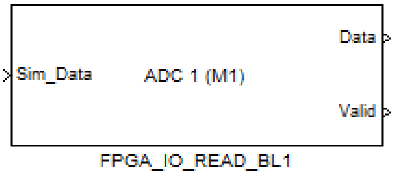
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

ADC (M1) description

Block display If you select an ADC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–5 V ... +5 V	Depending on the specified Scaling parameter, the output port range is: ▪ –5000 mV … +5000 mV or ▪ –8192 … +8191
–30 V ... +30 V	Depending on the specified Scaling parameter, the output port range is: ▪ –30000 mV … +30000 mV or ▪ –8192 … +8191

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Input voltage range: ▪ -5 V ... +5 V ▪ -30 V ... +30 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 Range: ▪ –5000 mV ... +5000 mV ▪ –30000 mV ... +30000 mV or ▪ –8192 … +8191 Update rate: 10 Msps
Valid	Represents the current status of the data output. Data type: UFix_1_0 ▪ 0: Converted value is out of range. ▪ 1: Converted value is within the specified input range.

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, it is treated as an invalid value:

- **Data** outport is saturated to minimum or maximum range value.
- **Valid** outport is set to 0.

I/O mapping The following I/O mapping is relevant if you use one of the *DS5203 with Multi-I/O Module (DS5203M1)* frameworks for analog input channels.

Outport	Channel	Connector Pin	Signal
Data	Ch 1	P2 1	$\overline{\text{ADC1}}$
		P2 34	ADC1
	Ch 2	P2 18	$\overline{\text{ADC2}}$
		P2 2	ADC2
	Ch 3	P2 35	$\overline{\text{ADC3}}$
		P2 19	ADC3
	Ch 4	P2 3	$\overline{\text{ADC4}}$
		P2 36	ADC4
	Ch 5	P2 20	$\overline{\text{ADC5}}$
		P2 4	ADC5
	Ch 6	P2 37	$\overline{\text{ADC6}}$
		P2 21	ADC6

For further information on the module's I/O connector, refer to DS5203 FPGA Board (PHS Bus System Hardware Reference).

ADC (M1) settings

The following settings on the **Parameters** page are specific to the **ADC (M1)** I/O function. For common dialog settings, refer to Common settings on page 476.

Input range Lets you select the input voltage range for the analog input channel.

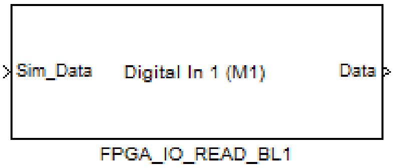
- –5 V ... +5 V
- –30 V ... +30 V

The selected input range is also valid for the enabled simulation data inport.

Scaling Lets you select the scaling of the output data. If you select **mV**, the valid output port range corresponds to the specified input range in mV (-5000 … +5000 mV or -30000 … +30000 mV). If you select the unscaled **Bit** value, the valid output port range is -8192 … +8191, independently from the specified input range.

Digital In (M1) description

Block display If you select a Digital In channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Threshold level: 1 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage. Update rate: 100 MHz

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
To use the maximum input voltage range, you have to use a digital channel only as the input.

I/O mapping The following I/O mapping is relevant if you use one of the *DS5203 with Multi-I/O Module (DS5203M1)* frameworks for digital input channels.

Outport	Channel	Connector Pin	Signal
Data	Ch 1	P2 22	DIG_IO1
	Ch 2	P2 6	DIG_IO2
	Ch 3	P2 23	DIG_IO3
	Ch 4	P2 7	DIG_IO4
	Ch 5	P2 24	DIG_IO5
	Ch 6	P2 8	DIG_IO6
	Ch 7	P2 25	DIG_IO7
	Ch 8	P2 9	DIG_IO8
	Ch 9	P2 26	DIG_IO9
	Ch 10	P2 10	DIG_IO10
	Ch 11	P2 27	DIG_IO11
	Ch 12	P2 11	DIG_IO12
	Ch 13	P2 28	DIG_IO13
	Ch 14	P2 12	DIG_IO14
	Ch 15	P2 29	DIG_IO15
	Ch 16	P2 13	DIG_IO16

You can use the same digital channel for input and output signals.

For further information on the module's I/O connector, refer to DS5203 FPGA Board (PHS Bus System Hardware Reference 🕮).

Digital In (M1) settings

The following settings on the Parameters page are specific to the Digital In (M1) I/O function. For common dialog settings, refer to Common settings on page 476.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1000 mV ... 7500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

Related topics

References

Description Page (FPGA_IO_READ_BL)........481
FPGA_IO_READ_BL........50
FPGA_IO_READ_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)........475

Scaling Page (FPGA_IO_READ_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Related topics

References

FPGA Access Page (FPGA_SETUP_BL)........35

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the Multi-I/O Module (DS5203M1) frameworks of DS5203 (7K325) and DS5203 (7K410) is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_IO_READ_BL........50
FPGA_IO_READ_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)........475
Parameters Page (FPGA_IO_READ_BL)........476

FPGA_IO_WRITE_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)

Purpose

To configure write access to analog and digital input signals in the FPGA model after you load one of the following frameworks:

- *DS5203 (7K325) with Multi-I/O Module (DS5203M1)*
- *DS5203 (7K410) with Multi-I/O Module (DS5203M1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the three I/O types *Analog*, *Digital*, and *Sensor Supply*, which you can select on the Unit page of the block. The number of the available physical connections on the DS5203M1 Multi-I/O Module determines the I/O functions that you can select:

- DAC 1 (M1) ... DAC 6 (M1)
- Digital Out 1 (M1) ... Digital Out 16 (M1)
- Sensor Supply

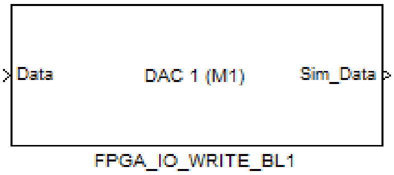
Common settings

The following settings on the Parameters page are common to the I/O functions that you can select on the Unit page of the FPGA_IO_WRITE_BL block.

Enable simulation port Lets you enable an outport for offline simulation data. The Sim_Data outport is added to the block to connect it to a Simulink-based I/O environment model.

DAC (M1) description

Block display If you have select an DAC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog output voltage and the input of the block is:

Output Voltage Range	Simulink Input
–10 V ... +10 V	Depending on the specified Scaling parameter, the input port range is: ▪ –10000 mV … +10000 mV or ▪ –8192 … +8191

The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data type: Fix_15_0 Output voltage range: -10000 mV ... +10000 mV or -8192 … +8191 (The range can be exceeded, and saturation is performed to a minimum or maximum value.)

Port	Description
	Hardware update rate: 10 MSPS (if the values are updated at a higher FPGA model rate, intermediate values are not updated by the DAC)
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the Parameters page. Data type: Double Data width: 1 Output voltage range: -10 V ... +10 V

I/O mapping The following I/O mapping is relevant if you use one of the *DS5203 with Multi-I/O Module (DS5203M1)* framework for analog output channels.

Inport	Channel	Connector Pin	Signal
Data	Ch 1	P2 30	$\overline{\text{DAC1}}$
		P2 14	DAC1
	Ch 2	P2 47	$\overline{\text{DAC2}}$
		P2 31	DAC2
	Ch 3	P2 15	$\overline{\text{DAC3}}$
		P2 48	DAC3
	Ch 4	P2 32	$\overline{\text{DAC4}}$
		P2 16	DAC4
	Ch 5	P2 49	$\overline{\text{DAC5}}$
		P2 33	DAC5
	Ch 6	P2 17	$\overline{\text{DAC6}}$
		P2 50	DAC6

For further information on the module's I/O connector, refer to DS5203 FPGA Board (PHS Bus System Hardware Reference 🕮).

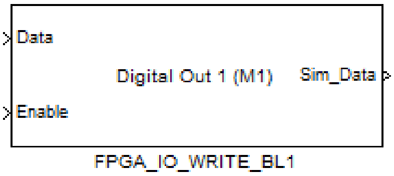
DAC (M1) settings

The following settings on the **Parameters** page are specific to the **DAC (M1)** I/O function. For common dialog settings, refer to Common settings on page 483.

Scaling Lets you select the scaling of the input data. If you select **mV**, the valid input port range is -10000 … +10000 mV. If you select the unscaled **Bit** value, the valid input port range is -8192 … +8191 (14-bit D/A converter).

Digital Out (M1) description

Block display If you select a Digital Out channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high supply voltage. The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: 100 MHz
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the Parameters page. Output voltage: 0 V … 3.3 V or 0 V … 5 V Update rate: 100 MHz

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

I/O mapping The following I/O mapping is relevant if you use the *DS5203 with Multi-I/O Module (DS5203M1)* framework for digital output channels.

Inport	Channel	Connector Pin	Signal
Data	Ch 1	P2 22	DIG_IO1
	Ch 2	P2 6	DIG_IO2
	Ch 3	P2 23	DIG_IO3
	Ch 4	P2 7	DIG_IO4
	Ch 5	P2 24	DIG_IO5
	Ch 6	P2 8	DIG_IO6
	Ch 7	P2 25	DIG_IO7
	Ch 8	P2 9	DIG_IO8
	Ch 9	P2 26	DIG_IO9
	Ch 10	P2 10	DIG_IO10
	Ch 11	P2 27	DIG_IO11
	Ch 12	P2 11	DIG_IO12
	Ch 13	P2 28	DIG_IO13
	Ch 14	P2 12	DIG_IO14
	Ch 15	P2 29	DIG_IO15
	Ch 16	P2 13	DIG_IO16

You can use the same digital channel for input and output signals.

For further information on the module's I/O connector, refer to DS5203 FPGA Board (PHS Bus System Hardware Reference 📖).

Digital Out (M1) settings

The following settings on the **Parameters** page are specific to the **Digital Out (M1)** I/O function. For common dialog settings, refer to Common settings on page 483.

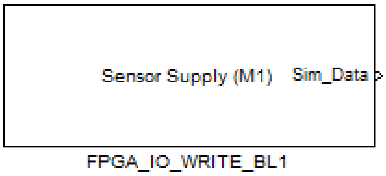
High supply Lets you select the voltage for the high side switch for all digital output channels.

Note

You can specify the high supply voltage value only globally for all digital output channels.

Sensor Supply description

Block display If you select the Sensor Supply channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Output voltage: 2 V … 20 V, in steps of 0.1 V, according to the specified supply voltage

I/O mapping The following I/O mapping is relevant if you use the *DS5203 with Multi-I/O Module (DS5203M1)* framework for the sensor supply channel.

Outport	Channel	Connector Pin	Signal
–	1	P2 5 P2 38	VSENS- VSENS+

For further information on the module's I/O connector, refer to DS5203 FPGA Board (PHS Bus System Hardware Reference 📖).

Sensor Supply settings

The following settings on the Parameters page are specific to the Sensor Supply I/O function. For common dialog settings, refer to Common settings on page 483.

Supply voltage Lets you enter the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

Related topics

References

Description Page (FPGA_IO_WRITE_BL).. 488
FPGA_IO_WRITE_BL.. 56
FPGA_IO_WRITE_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings)........................ 482

Scaling Page (FPGA_IO_WRITE_BL)

Description

The **Scaling** page is empty because the I/O functions of this framework do not support FPGA scaling.

Related topics

References

FPGA Access Page (FPGA_SETUP_BL) 35

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the Multi-I/O Module (DS5203M1) frameworks of DS5203 (7K325) and DS5203 (7K410) is included in this user documentation. The description of the access type of customized frameworks or mounted piggybacks will be available as separate documents.

Related topics

References

FPGA_IO_WRITE_BL 56
FPGA_IO_WRITE_BL (DS5203 with Multi-I/O Module (DS5203M1) Settings) 482
Parameters Page (FPGA_IO_WRITE_BL) 483

RTI Block Settings for the FPGA1401Tp1 with Multi-I/O Module Frameworks

Introduction

The block dialogs provide hardware-specific settings after you load one of the following MicroAutoBox II frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)

Purpose

To configure read access to intermodule bus data in the FPGA model after you load one of the following frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)....................490
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)....................497
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)....................43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)....................44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)
To write data from the processor model to the FPGA model via the board-specific bus.

FPGA_XDATA_WRITE_BL....................45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)....................498
To configure write access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1401Tp1 with Multi-I/O Module frameworks provide the following access types that you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from an intermodule bus register. 128 registers are available with a data width of

32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element.

If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Since register groups can be only accessed by one task, you have to explicitly group registers which are used by different tasks.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.

- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

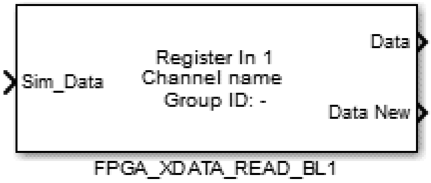
The fraction width is displayed in the Binary point position (or fraction width) setting.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from a intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a

Port	Description
	new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

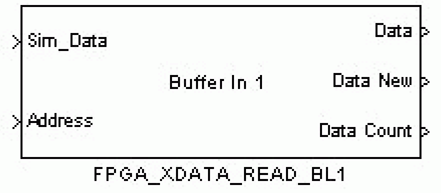
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 491.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater

Port	Description
	than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 491.

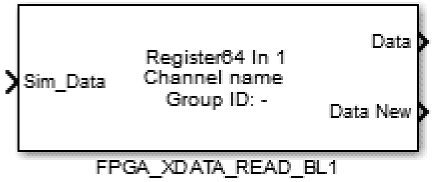
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

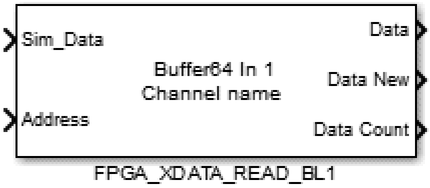
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 491.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer64 In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53

Port	Description
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

> **Note**
>
> The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 491.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)... 497
FPGA_XDATA_READ_BL.. 40
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)............................... 489

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)

Purpose

To configure write access to intermodule bus data in the FPGA model after you load one of the following frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_WRITE_BL)..499
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..506
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)..48
To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)..49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

PROC_XDATA_READ_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)
To read data in the processor model that comes from the FPGA model via the board-specific bus.

FPGA_XDATA_READ_BL..40
To implement read access to processor-bus data in the FPGA model.

FPGA_XDATA_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)..489
To configure read access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1401Tp1 with Multi-I/O Module frameworks provide the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an intermodule bus register. 128 registers are available with a data width of 32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** inport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

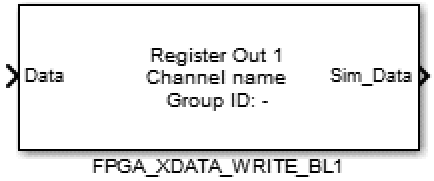
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

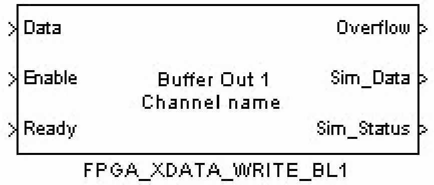
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 500.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	**Description**
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read.

Port	Description
	- 0: No overflow occurred. - 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 - Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. - Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1, if the buffer contains new values. - Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data is lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 500.

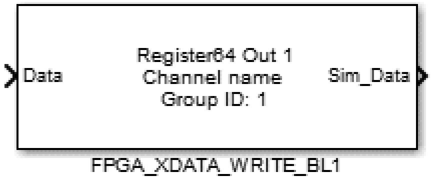
Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

> **Note**
>
> The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

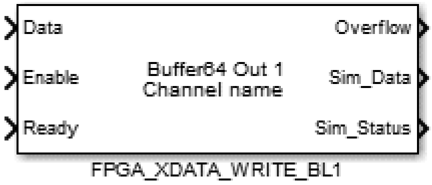
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 500.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready o send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32

Port	Description
	Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 500.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Related topics

References

Description Page (FPGA_XDATA_WRITE_BL).. 506
FPGA_XDATA_WRITE_BL.. 45
FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings).. 498

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model after you load one of the following frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_IO_READ_BL)..508
To specify relevant settings for the selected I/O function.

Scaling Page (FPGA_IO_READ_BL)..521
To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description Page (FPGA_IO_READ_BL)..524
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Block Description (FPGA_IO_READ_BL)..54
To implement read access to a physical input channel in the FPGA model.

Unit Page (FPGA_IO_READ_BL)..55
To specify the I/O type and channel to be used for read access.

Related RTI blocks

FPGA_IO_WRITE_BL..56
To provide write access to an external device via a physical output channel.

FPGA_IO_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)..525
To configure write access to analog and digital output signals in the FPGA model.

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides different I/O types, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1552 or DS1552B1 Multi-I/O Module determines the I/O functions that you can select:

- ADC 1 (Type A) … ADC 8 (Type A)
- ADC 1 (Type B) … ADC 16 (Type B)
- Digital Crank/Cam Sensor 1 ... Digital Crank/Cam Sensor 3

  To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- Digital In 1 (Type A) … Digital In 16 (Type A)
- Digital In 1 (Type B) … Digital In 8 (Type B)
- Inductive Zero Voltage Detector

  To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.
- Status In
- Temperature

  To provide read access to the FPGA's die temperature.

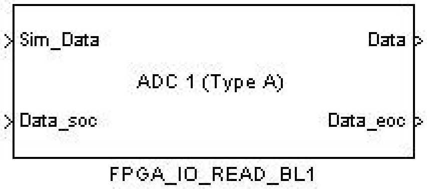
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block, except for the Status In and Temperature function.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

ADC (Type A) description

Block display If you select an ADC (Type A) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Multi I/O Module	Input Voltage Range	Simulink Output
DS1552	0 V ... +5 V	The output port range is: 0 … +65535
DS1552B1	-10 V ... +10 V	The output port range is: 0 … +65535

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1 Input voltage range (DS1552): 0 V … +5 V Input voltage range (DS1552B1): -10 V ... +10 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The `Data_eoc` outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle.

Port	Description
	Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the Sim_Data inport exceeds the specified input voltage range, Data outport is saturated to minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	X3 X4	AnalogIn+ ch 1 AnalogIn- ch 1[1]
	2	W3 W4	AnalogIn+ ch 2 AnalogIn- ch 2[1]
	3	V3 V4	AnalogIn+ ch 3 AnalogIn- ch 3[1]
	4	U3 U4	AnalogIn+ ch 4 AnalogIn- ch 4[1]
	5	H3 H4	AnalogIn+ ch 5 AnalogIn- ch 5[1]
	6	G3 G4	AnalogIn+ ch 6 AnalogIn- ch 6[1]
	7	F3 F4	AnalogIn+ ch 7 AnalogIn- ch 7[1]
	8	E3 E4	AnalogIn+ ch 8 AnalogIn- ch 8[1]

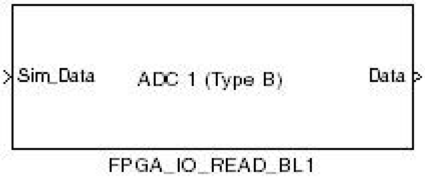
[1] The negative input line of the ADC channel is connected to GND. To get optimum analog performance, follow the instructions in Connecting Sensor Ground Lines to MicroAutoBox II (MicroAutoBox II Hardware Installation and Configuration Guide 🕮) for connecting the analog channels to GND.

ADC (Type A) settings

Only common dialog settings. Refer to Common settings on page 508.

ADC (Type B) description

Block display If you select an ADC (Type B) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–10 V ... +10 V	The output port range is: –32768 … +32767

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -10 V ... +10 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 Range: -32768 … +32767 Update rate: 0.2 Msps

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

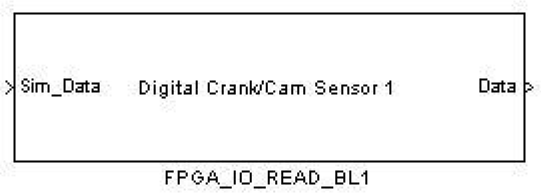
Outport	Channel	Connector Pin	Signal
Data	1	b2	AnalogIn ch 1
	2	a2	AnalogIn ch 2
	3	Z2	AnalogIn ch 3
	4	Y2	AnalogIn ch 4
	5	X2	AnalogIn ch 5
	6	W2	AnalogIn ch 6
	7	V2	AnalogIn ch 7
	8	U2	AnalogIn ch 8
	9	M2	AnalogIn ch 9
	10	L2	AnalogIn ch 10
	11	K2	AnalogIn ch 11
	12	J2	AnalogIn ch 12
	13	H2	AnalogIn ch 13
	14	G2	AnalogIn ch 14
	15	F2	AnalogIn ch 15
	16	E2	AnalogIn ch 16

ADC (Type B) settings

Only common dialog settings. Refer to Common settings on page 508.

Digital Crank/Cam Sensor description

Block display If you select a Digital Crank/Cam Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V

Port	Description
Output	
Data	Outputs the status of the crank/cam sensor. Data type: UFix_1_0 ▪ 0: The input signal is lower than the **Low threshold voltage** parameter. ▪ 1: The input signal is higher than the **High threshold voltage** parameter. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	R3	CrankCam+ ch 1
		R4	CrankCam– ch 1
	2	B3	CrankCam+ ch 2
		B4	CrankCam– ch 2
	3	A3	CrankCam+ ch 3
		A4	CrankCam– ch 3

Digital Crank/Cam Sensor settings

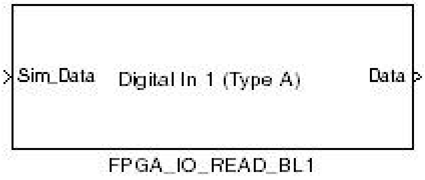
The following settings on the **Parameters** page are specific to the **Digital Crank/Cam Sensor** I/O function. For common dialog settings, refer to Common settings on page 508.

Low threshold voltage (-40000 mV ... +40000 mV) Lets you set the low threshold level for the selected digital input channel. Below this level a logical 0 is detected, above this level a logical 1 is detected, if the high threshold voltage was crossed before.

High threshold voltage (-40000 mV ... +40000 mV) Lets you set the high threshold level for the selected digital input channel. The logical 1 is output, if this level is crossed and stays 1 until the signal falls below the low threshold level.

Digital In (Type A) description

Block display If you select a Digital In (Type A) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: ▪ 3.6 V for low-high transition ▪ 1.2 V for high-low transition
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital Inputs (MicroAutoBox II Hardware Reference).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

> **Note**
>
> Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

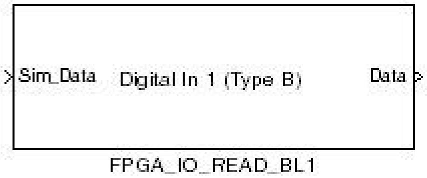
Outport	Channel	Connector Pin	Signal
Data	1	V5	DigIn ch 1
	2	U5	DigIn ch 2
	3	U6	DigIn ch 3
	4	T2	DigIn ch 4
	5	T3	DigIn ch 5
	6	T4	DigIn ch 6
	7	T5	DigIn ch 7
	8	T6	DigIn ch 8
	9	S2	DigIn ch 9
	10	S3	DigIn ch 10
	11	S5	DigIn ch 11
	12	R2	DigIn ch 12
	13	R5	DigIn ch 13
	14	R6	DigIn ch 14
	15	P5	DigIn ch 15
	16	P6	DigIn ch 16

Digital In (Type A) settings

Only common dialog settings. Refer to Common settings on page 508.

Digital In (Type B) description

Block display If you select a Digital In (Type B) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 1 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage. Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital I/O (Bidirectional) (MicroAutoBox II Hardware Reference 🕮).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

> **Note**
>
> Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	N2	DigIO ch1
	2	N3	DigIO ch2
	3	N4	DigIO ch3
	4	N5	DigIO ch4
	5	N6	DigIO ch5
	6	M5	DigIO ch6
	7	M6	DigIO ch7
	8	L4	DigIO ch8

You can use the same digital channel for input and output signals.

Digital In (Type B) settings

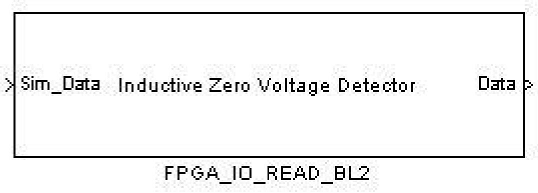
The following settings on the Parameters page are specific to the Digital In (Type B) I/O function. For common dialog settings, refer to Common settings on page 508.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1,000 mV … 7,500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

Inductive Zero Voltage Detector description

Block display If you select a Inductive Zero Voltage Detector channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Detects the zero crossing points of the analog signals. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle. Data type: UFix_1_0 ▪ 0: No zero crossing. ▪ 1: Zero crossing is detected. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

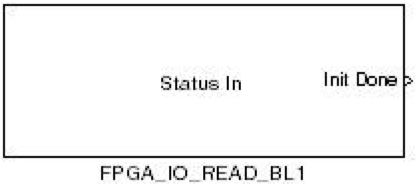
Outport	Connector Pin	Signal
Data	P3	ZeroDetection+
	P4	ZeroDetection–

Inductive Zero Voltage Detector settings

Only common dialog settings. Refer to Common settings on page 508.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for initialization sequence status information:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

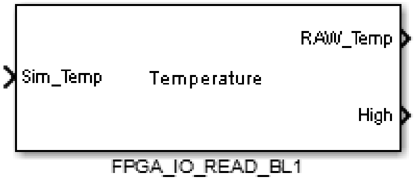
I/O mapping No external connection.

Status In settings

None

Temperature

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C

Port	Description
	The range can be exceeded. The values are then saturated to the minimum or maximum values.
Output	
RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature. Equation for calculating the die temperature: `Temperature [°C] = (float)(Temperature[hex] & 0xFFF0) · 503.975 / 65536 - 273.15` Data type: UFix_16_0 Data width: 1 Value range: 0 ... 65536
High	Outputs a flag if the FPGA's die temperature exceeds 105 °C. To reset the flag, the die temperature must fall below 85 °C. Data type: UFix_1_0 ▪ 0: Die temperature does not exceed 105 °C. ▪ 1: Die temperature exceeds 105 °C.

Note

A high ambient temperature and an FPGA application with a very high FPGA utilization and/or toggle rate increase the FPGA die temperature (internal chip temperature). If the die temperature exceeds 105 °C, the FPGA might work incorrectly.

You can decrease the temperature by reducing the FPGA's toggle rate (e.g., by using clock enable) or by reducing the utilization of the FPGA resources.

If the die temperature exceeds 125 °C, the FPGA resets itself. The reset stays active until the die temperature falls below 85 °C and you restart MicroAutoBox II or reload the user application.

I/O mapping No external connection.

Temperature settings Only common dialog settings. Refer to Common settings on page 508.

Related topics

References

Description Page (FPGA_IO_READ_BL)....................524
FPGA_IO_READ_BL....................50
FPGA_IO_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings)....................507
Scaling Page (FPGA_IO_READ_BL)....................521

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

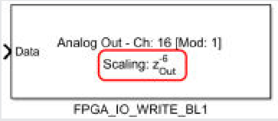
Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

ADC (Type A and Type B) settings

The following settings on the Scaling page are specific to the ADC (Type A) and ADC (Type B) I/O functions.

> **Note**
>
> FPGA scaling of analog I/O signals also effects the signal path:
>
> - If you use the parameters of the Scaling page to scale analog signals, the data type of the Data port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
> - FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.
>
> 
>

Scaling format Lets you select the data format for scaling and saturation.

- Signed/Unsigned:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the Number of bits and Binary point (fraction width) parameters.
- Single:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum**

value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital In (Type A and Type B) settings

The following settings on the **Scaling** page are specific to the **Digital In (Type A)** and **Digital In (Type B)** I/O functions.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

Digital Crank/Cam Sensor settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Inductive Zero Voltage Detector

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Status In

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Temperature

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_READ_BL)... 524
FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings).. 507
Parameters Page (FPGA_IO_READ_BL).. 508

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (FPGA1401Tp1 with Multi-I/O Module Settings).. 507
Parameters Page (FPGA_IO_READ_BL).. 508
Scaling Page (FPGA_IO_READ_BL)... 521

FPGA_IO_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model after you load one of the following frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the three I/O types *Analog*, *Digital*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1552 Multi-I/O Module determines the I/O functions that you can select:

- DAC 1 … DAC 4
- Digital Out 1 (Type A) … Digital Out 16 (Type A)
- Digital Out 1 (Type B) … Digital Out 8 (Type B)
- LED Out
- Sensor Supply
- UART 1 (RS232) … UART 2 (RS232)
- UART 1 (RS422/485) … UART 2 (RS422/485)

Common settings

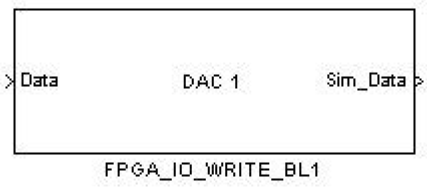
The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out and Sensor Supply functions.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

If you have selected one of the UART functions, this setting is replaced by function-specific simulation settings.

DAC description

Block display If you have select a DAC channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog output voltage and the input of the block is:

Output Voltage Range	Simulink Input
0 V … +5 V	The input port range is: 0 … +65535

The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data type: UFix_16_0 Output voltage range: 0 … +65535 The range can be exceeded, and saturation is performed to a minimum or maximum value. Hardware update rate: 2.1 Msps If the values are updated at a higher FPGA model rate, intermediate values are not updated by the DAC.
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Output voltage range: 0 V … +5 V

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

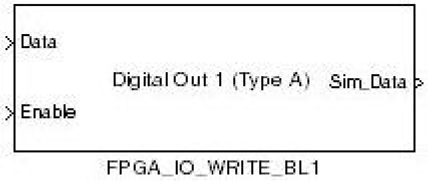
Inport	Channel	Connector Pin	Signal
Data	1	c2	AnalogOut ch 1
	2	c3	AnalogOut ch 2
	3	c4	AnalogOut ch 3
	4	c5	AnalogOut ch 4

DAC settings

Only common dialog settings. Refer to Common settings on page 526.

Digital Out (Type A) description

Block display If you select a Digital Out (Type A) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is set to the specified high supply voltage (VDRIVE). The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital Outputs (MicroAutoBox II Hardware Reference 🕮).
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V ... 45 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Inport	Channel	Connector Pin	Signal
Data	1	F5	DigOut ch 1
	2	E5	DigOut ch 2
	3	E6	DigOut ch 3
	4	D2	DigOut ch 4
	5	D3	DigOut ch 5
	6	D4	DigOut ch 6
	7	D5	DigOut ch 7
	8	D6	DigOut ch 8
	9	C2	DigOut ch 9
	10	C3	DigOut ch 10
	11	C5	DigOut ch 11
	12	B2	DigOut ch 12
	13	B5	DigOut ch 13
	14	B6	DigOut ch 14
	15	A5	DigOut ch 15
	16	A6	DigOut ch 16

Digital Out (Type A) settings

The following settings on the Parameters page are specific to the Digital Out (Type A) I/O function. For common dialog settings, refer to Common settings on page 526.

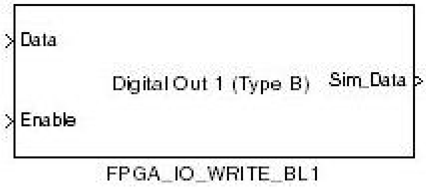
Simulated VDRIVE pin Lets you select the voltage for the simulated high side switch for all digital output channels in the range 0 … 45000 mV.

Note

You can specify the simulated voltage value only globally for all digital output channels.

Digital Out (Type B) description

Block display If you select a Digital Out (Type B) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high supply voltage. The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital I/O (Bidirectional) (MicroAutoBox II Hardware Reference 📖).
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V … 3.3 V or 0 V … 5 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	N2	DigIO ch1
	2	N3	DigIO ch2
	3	N4	DigIO ch3
	4	N5	DigIO ch4
	5	N6	DigIO ch5
	6	M5	DigIO ch6
	7	M6	DigIO ch7
	8	L4	DigIO ch8

You can use the same digital channel for input and output signals.

Digital Out (Type B) settings

The following settings on the **Parameters** page are specific to the **Digital Out (Type B)** I/O function. For common dialog settings, refer to Common settings on page 526.

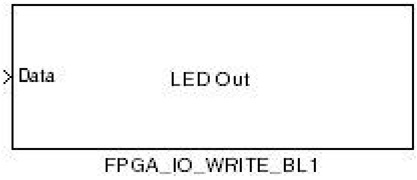
High supply Lets you select the voltage for the high side switch (3.3 V or 5 V) for all digital output channels.

Note

You can specify the high supply voltage value only globally for all digital output channels.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the FPGA status LED near the DS1514 ZIF I/O connector. Data type: UFix_1_0 ▪ 0: LED lights up green. ▪ 1: LED lights up orange.

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used (=1).

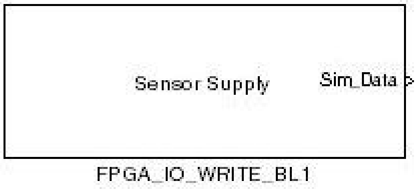
I/O mapping No external connection.

LED Out settings

None

Sensor Supply description

Block display If you select the Sensor Supply channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Output voltage: 2 V … 20 V according to the specified supply voltage

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6 c6	VS- VS+

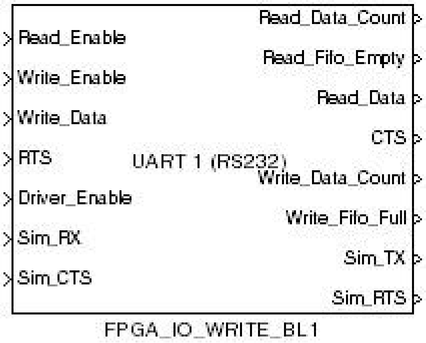
Sensor Supply settings

The following settings on the Parameters page are specific to the Sensor Supply I/O function. For common dialog settings, refer to Common settings on page 526.

Supply voltage Lets you enter the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

UART (RS232) description

Block display If you select an UART (RS232) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next **Read_Enable** signal. Before you read data from the RX FIFO buffer, you should check the **Read_Fifo_Empty** signal not to be set. The **Read_Fifo_Empty** signal switches one clock cycle after the RX FIFO value has been read. Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle. Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud

Port	Description
	rate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
RTS	Specifies the Ready-To-Send (RTS) signal. The RTS/CTS handshake is handled by the user, the RTS signal is just passed through and adapted to the physical layer. Data type: UFix_1_0 Data width: 1 The hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the **Parameters** page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_CTS	Simulates the Clear-To-Send (CTS) hardware signal. Available only if **Enable simulation CTS port** is set on the **Parameters** page. The RTS/CTS handshake is handled by the user. The **Sim_CTS** signal is just passed through to CTS. Data type: Double Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries.

Port	Description
	If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible. The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
CTS	Outputs the state of the Clear-To-Send (CTS) hardware port. RTS/CTS handshake is handled by the user. CTS is just passed through with conversion to logical 1 and 0. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active The CTS hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive). Range exceeding is not possible.
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already.

Port	Description
	Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the TX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the **Parameters** page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_RTS	Simulates the Ready-To-Send (RTS) hardware signal. Available only if **Enable simulation RTS port** is set on the **Parameters** page. The signal is in logical format and only passed through to the RTS signal. Data type: Double Data width: 1 Range: ▪ 0: RTS inactive ▪ 1: RTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Inport	Connector Pin	Signal
UART 1 (RS232)		
Write_Data	a5	TX1
RTS	a6	RTS1
Read_Data	b5	RX1
CTS	a4	CTS1
UART 2 (RS232)[1]		
Write_Data	Z5	TX2
RTS	Z6	RTS2
Read_Data	Z3	RX2
CTS	Z4	CTS2

[1] To use UART 2, your DS1552 has to be modified by dSPACE.

UART (RS232) settings

The following settings on the **Parameters** page are specific to the **UART (RS232)** I/O function.

Baud rate Lets you select the baud rate in the range 50 … 1,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

> **Note**
>
> The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The **Write_Data** and **Read_Data** ports handle raw bits. If you want to use a parity bit with the **Write_Data** port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the **Read_Data** port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

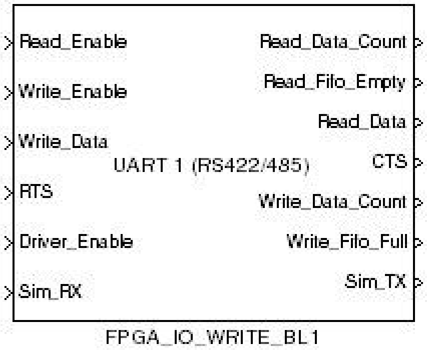
Enable simulation CTS port Lets you enable an inport for offline simulation data. The Sim_CTS inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable simulation RTS port Lets you enable an outport for offline simulation data. The Sim_RTS outport is added to the block to connect it to a Simulink-based I/O environment model.

UART (RS422/485) description

Block display If you select an UART (RS422/485) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next **Read_Enable** signal. Before you read data from the RX FIFO buffer, you should check the **Read_Fifo_Empty** signal not to be set. The **Read_Fifo_Empty** signal switches one clock cycle after the RX FIFO value has been read. Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle.

Port	Description
	Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud rate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
Driver_Enable	Specifies to enable the output driver in the transceiver for data transmission. If you use the UART (RS485/422) function in half-duplex mode, the output driver must be disabled while receiving data. Data type: UFix_1_0 Data width: 1
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the **Parameters** page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries. If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047

Port	Description
	The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible. The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already. Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the RX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.

Port	Description
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the **Parameters** page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1401Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector. The mapping differs when using the UART (RS422/485) in full-duplex or half-duplex mode.

Full-duplex mode:

Inport		Connector Pin	Signal
UART 1 (RS422/485)			
	Write_Data	a5	TX-1
		a6	TX+1
	Read_Data	b5	RX-1
		a4	RX+1
UART 2 (RS422/485)[1]			
	Write_Data	Z5	TX-2
		Z6	TX+2
	Read_Data	Z3	RX-2
		Z4	RX+2

[1] To use UART 2, your DS1552 has to be modified by dSPACE.

Half-duplex mode:

Inport	Connector Pin	Signal
UART 1 (RS422/485)		
Write_Data	a5	BM1 (RX-1/TX-1)
	a6	BP1 (RX+1/TX+1)
Read_Data	b5	-[1]
	a4	-[1]
UART 2 (RS422/485)[2]		
Write_Data	Z5	BM2 (RX-2/TX-2)
	Z6	BP2 (RX+2/TX+2)
Read_Data	Z3	-[1]
	Z4	-[1]

[1] Do not connect, TX signals are available via BM and BP signals.

[2] To use UART 2, your DS1552 has to be modified by dSPACE.

UART (RS422/485) settings

The following settings on the **Parameters** page are specific to the **UART (RS422/485)** I/O function.

Baud rate Lets you select the baud rate in the range 50 … 10,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

Note

The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The **Write_Data** and **Read_Data** ports handle raw bits. If you want to use a parity bit with the **Write_Data** port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the **Read_Data** port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Mode Lets you select the mode for receiving messages.

- Full duplex mode

  You can simultaneously send and receive signals on the UART channel.

- Half duplex mode

  You can send or receive signals on the UART channel, but you cannot do both at the same time.

Termination Lets you enable an internal termination between RX- and RX+ and between TX- and TX+.

Setting	Meaning
Open	No termination
Terminated	The RX and TX signals are terminated via an internal 120 Ω resistor.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........ 546
FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)........ 525
Scaling Page (FPGA_IO_WRITE_BL)........ 543

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

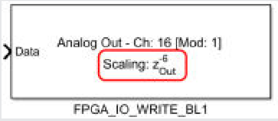
DAC settings

The following settings on the **Scaling** page are specific to the **DAC** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling factor** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Scaling offset** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation minimum value Lets you specify the minimum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the scaled **Data** inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.

- 1 ... 20: The adding will be implemented with the specified latency.
  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital Out (Type A and Type B) settings

The following settings on the **Scaling** page are specific to the **Digital Out (Type A)** and **Digital Out (Type B)** I/O functions.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:
  If driven with 1, the hardware outputs a high-level signal.
  If driven with 0, the hardware outputs a low-level signal.
- **Enabled**:
  If driven with 1, the hardware outputs a low-level signal.
  If driven with 0, the hardware outputs a high-level signal.

LED Out settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Sensor Supply settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

UART settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........ 546
FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Multi-I/O Module Settings)........ 525
Parameters Page (FPGA_IO_WRITE_BL)........ 526

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text

field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL (FPGA1401Tp1 with Multi-I/O Module Settings)

Purpose

To configure the FPGA interrupt channel after you load one of the following frameworks:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL)..548
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL)..549
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL)..66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 🕮)

To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

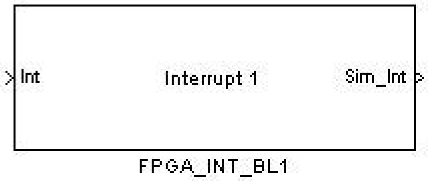
Description

The FPGA1401Tp1 with Multi-I/O Module frameworks provide 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function-call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The **Sim_Int** outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL).......... 549
FPGA_INT_BL.......... 62
FPGA_INT_BL (FPGA1401Tp1 with Multi-I/O Module Settings).......... 547

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1401Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL.......... 62
FPGA_INT_BL (FPGA1401Tp1 with Multi-I/O Module Settings).......... 547
Parameters Page (FPGA_INT_BL).......... 548

RTI Block Settings for the FPGA1401Tp1 with Engine Control I/O Module Framework

Introduction

The block dialogs provide hardware-specific settings after you load the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework for the MicroAutoBox II.

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)

Purpose

To configure read access to the local bus data in the FPGA model when using the DS1554 Engine Control I/O Module framework.

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)..551
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)..558
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To write data from the processor model to the FPGA model via the board-specific bus.

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)..559
To configure write access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The DS1554 Engine Control I/O Module framework provides the following access types, which you can select on the **Unit** page of the block's dialog:

- Fixed-point format:

  Signed or unsigned data format with adjustable binary point position

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- 32-bit floating-point format:

  Single precision (IEEE 754 standard) data format with a fraction width of 24
- 64-bit floating-point format:

  Double precision (IEEE 754 standard) data format with a fraction width of 53

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

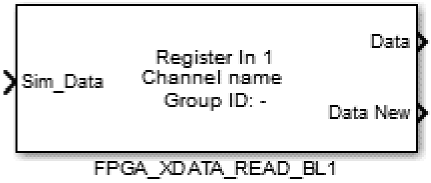
Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data**. By default, the sample time is inherited. You must only specify the sample time explicitly if the modeling situation inhibits to inherit the sample time, for example, if the corresponding

PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

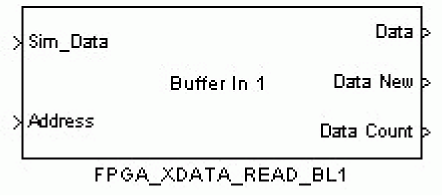
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 552.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0

Port	Description
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 552.

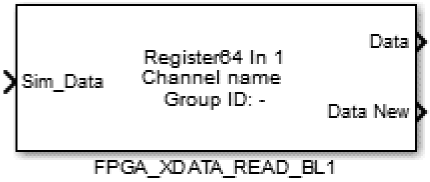
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

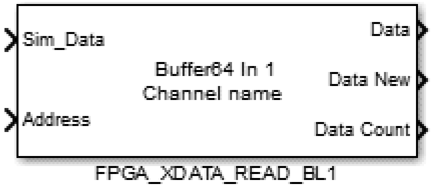
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 552.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer64 In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53

Port	Description
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 552.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_READ_BL...40
FPGA_XDATA_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)...............550
Parameters Page (FPGA_XDATA_READ_BL)...551

FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)

Purpose

To configure write access to intermodule bus data in the FPGA model when using the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

To specify the data format and specific settings for the selected access type.

To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)..48

To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)..49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

PROC_XDATA_READ_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To read data in the processor model that comes from the FPGA model via the board-specific bus.

FPGA_XDATA_READ_BL..40
To implement read access to processor-bus data in the FPGA model.

FPGA_XDATA_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)..550
To configure read access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides the following access types, which you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an intermodule bus register. 128 registers are available with a data width of 32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** inport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

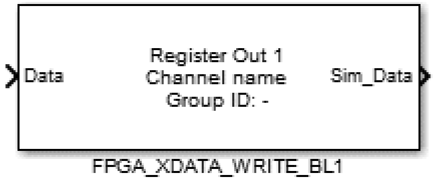
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block so you can connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

> **Note**
>
> The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

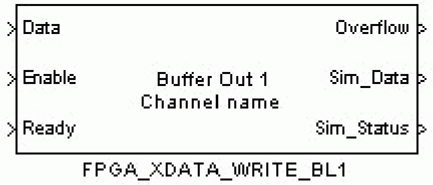
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 561.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via the intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read.

Port	Description
	▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer**, and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 561.

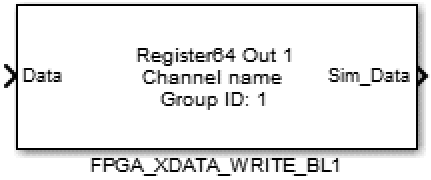
Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enable the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

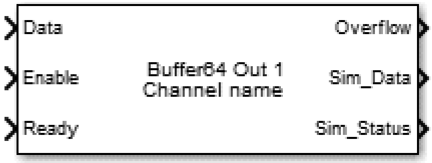
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 561.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



FPGA_XDATA_WRITE_BL1

I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type: - Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. - Floating-point format XFloat_11_53
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 - 0: The **Data** value to be written is not stored in the buffer. - 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 - 0: The buffer is not ready to send. - 1: The buffer is marked as ready o send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. - 0: No overflow occurred. - 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32

Port	Description
	Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 561.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Related topics

References

Description Page (FPGA_XDATA_WRITE_BL).. 567
FPGA_XDATA_WRITE_BL... 45
FPGA_XDATA_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)............... 559

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_READ_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides different I/O types, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1554 Engine Control I/O Module determines the I/O functions that you can select:

- Status In
- Digital Crank/Cam Sensor 1 ... Digital Crank/Cam Sensor 5

  To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- Inductive Zero Voltage Detector

  To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.
- Temperature

  To provide read access to the FPGA's die temperature.
- Digital In 1 (Type B) … Digital In 8 (Type B)
- ADC 1 (Type A) … ADC 14 (Type A)
- Knock Sensor 1 ... Knock Sensor 4

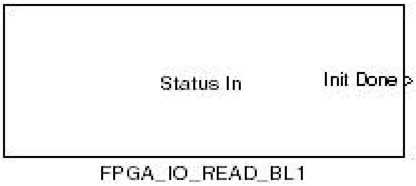
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block, except for the Status In and Temperature functions.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to simulation data coming from a Simulink-based I/O environment model.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for initialization sequence status information:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence finished.

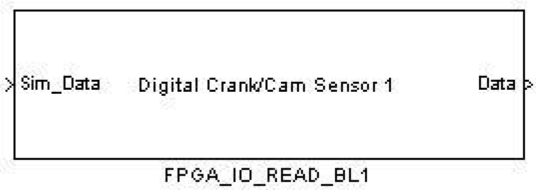
I/O mapping No external connection.

Status In settings

None

Digital Crank/Cam Sensor description

Block display If you select a Digital Crank/Cam Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V

Port	Description
Output	
Data	Outputs the status of the crank/cam sensor. Data type: UFix_1_0 - 0: The input signal is lower than the **Low threshold voltage** parameter. - 1: The input signal is higher than the **High threshold voltage** parameter. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, the **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	13	CrankCam Ch 1
	2	32	CrankCam Ch 2
	3	14	CrankCam Ch 3
	4	33	CrankCam Ch 4
	5	12	CrankCam Ch 5

Digital Crank/Cam Sensor settings

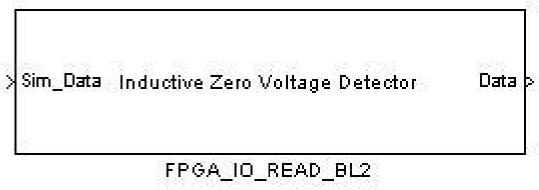
The following settings on the **Parameters** page are specific to the **Digital Crank/Cam Sensor** I/O function. For common dialog settings, refer to Common settings on page 569.

Low threshold voltage (-40000 mV ... +40000 mV) Lets you set the low threshold level for the selected digital input channel in steps of 100 mV. Below this level, a logical 0 is detected, above this level, a logical 1 is detected if the high threshold voltage was crossed before.

High threshold voltage (-40000 mV ... +40000 mV) Lets you set the high threshold level for the selected digital input channel in steps of 100 mV. The logical 1 is output if this level is crossed and stays 1 until the signal falls below the low threshold level.

Inductive Zero Voltage Detector description

Block display If you select an Inductive Zero Voltage Detector channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for Inductive Zero Voltage Detector channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Detects the zero crossing points of the analog signals. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle. Data type: UFix_1_0 ▪ 0: No zero crossing. ▪ 1: Zero crossing is detected. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, the **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

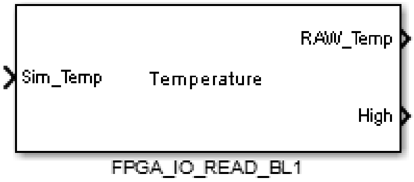
Outport	Connector Pin	Signal
Data	10	ZeroDetection+
	29	ZeroDetection–

Inductive Zero Voltage Detector settings

Only common dialog settings. Refer to Common settings on page 569.

Temperature description

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C The range can be exceeded. The values are then saturated to the minimum or maximum values.
Output	
RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature. Equation for calculating the die temperature: `Temperature [°C] = (float)(Temperature[hex] & 0xFFF0) · 503.975 / 65536 - 273.15` Data type: UFix_16_0 Data width: 1 Value range: 0 ... 65536
High	Outputs a flag if the FPGA's die temperature exceeds 105 °C. To reset the flag, the die temperature must fall below 85 °C. Data type: UFix_1_0 ▪ 0: Die temperature does not exceed 105 °C. ▪ 1: Die temperature exceeds 105 °C.

Note

A high ambient temperature and an FPGA application with a very high FPGA utilization and/or toggle rate increase the FPGA die temperature (internal chip temperature). If the die temperature exceeds 105 °C, the FPGA might work incorrectly.

You can decrease the temperature by reducing the FPGA's toggle rate (e.g., by using clock enable) or by reducing the utilization of the FPGA resources.

If the die temperature exceeds 125 °C, the FPGA resets itself. The reset stays active until the die temperature falls below 85 °C and you restart MicroAutoBox II or reload the user application.

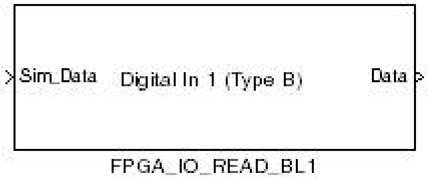
I/O mapping No external connection.

Temperature settings

Only common dialog settings. Refer to Common settings on page 569.

Digital In (Type B) description

Block display If you select a Digital In (Type B) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 1 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of the digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage.

Port	Description
	Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Data Sheet DS1554 Engine Control I/O Module (MicroAutoBox II Hardware Reference).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the appropriate minimum or maximum value.

> **Note**
>
> Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

> **Note**
>
> If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
>
> To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide).

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	c3	DigIO ch1
	2	b5	DigIO ch2
	3	b2	DigIO ch3
	4	c5	DigIO ch4
	5	c4	DigIO ch5
	6	c2	DigIO ch6
	7	a2	DigIO ch7
	8	Z2	DigIO ch8

You can use the same digital channel for input and output signals.

Digital In (Type B) settings

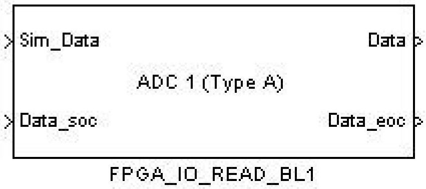
The following settings on the **Parameters** page are specific to the **Digital In (Type B)** I/O function. For common dialog settings, refer to Common settings on page 569.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1,000 mV … 7,500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

ADC (Type A) description

Block display If you select an ADC (Type A) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–10 V ... +10 V	The outport range is: 0 … +65535

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1 Input voltage range: -10 V ... +10 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The **Data_eoc** outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	W2	AnalogIn+ ch 1
		V2	AnalogIn- ch 1[1]
	2	Y2	AnalogIn+ ch 2
		X2	AnalogIn- ch 2[1]
	3	S2	AnalogIn+ ch 3
		R2	AnalogIn- ch 3[1]
	4	T2	AnalogIn+ ch 4
		U2	AnalogIn- ch 4[1]
	5	V5	AnalogIn+ ch 5
		W6	AnalogIn- ch 5[1]
	6	W3	AnalogIn+ ch 6
		V3	AnalogIn- ch 6[1]
	7	T3	AnalogIn+ ch 7
		U3	AnalogIn- ch 7[1]
	8	U5	AnalogIn+ ch 8
		V6	AnalogIn- ch 8[1]
	9	S5	AnalogIn+ ch 9
		T6	AnalogIn- ch 9[1]
	10	T5	AnalogIn+ ch 10
		U6	AnalogIn- ch 10[1]
	11	R5	AnalogIn+ ch 11
		R6	AnalogIn- ch 11[1]
	12	S3	AnalogIn+ ch 12
		R3	AnalogIn- ch 12[1]
	13	P5	AnalogIn+ ch 13
		P6	AnalogIn- ch 13[1]
	14	P3	AnalogIn+ ch 14
		P2	AnalogIn- ch 14[1]

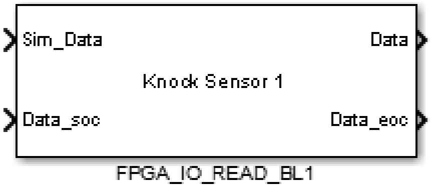
[1] The negative input line of the ADC channel is connected to GND. For achieving optimum analog performance, refer to Connecting Sensor Ground Lines to MicroAutoBox II (MicroAutoBox II Hardware Installation and Configuration Guide 🕮).

ADC (Type A) settings

Only common dialog settings. Refer to Common settings on page 569.

Knock Sensor description

Block display If you select the Knock Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–5 V ... +5 V	The outport range is: 0 … +65535

The following table describes the ports of the block for knock sensor input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: –5 V ... +5 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The **Data_eoc** outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the Sim_Data inport exceeds the specified input voltage range, Data outport is saturated to the minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	16 34	KnockIn+ ch 1 KnockIn– ch 1[1]
	2	17 35	KnockIn+ ch 2 KnockIn– ch 2[1]
	3	18 36	KnockIn+ ch 3 KnockIn– ch 3[1]
	4	19 37	KnockIn+ ch 4 KnockIn– ch 4[1]

1) The negative input line of the knock sensor input channel is connected to GND. For achieving optimum analog performance, refer to Connecting Sensor Ground Lines to MicroAutoBox II (MicroAutoBox II Hardware Installation and Configuration Guide 🕮).

Knock Sensor settings

Only common dialog settings. Refer to Common settings on page 569.

Related topics

References

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

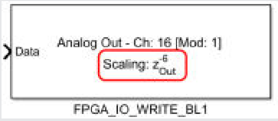
ADC (Type A) settings

The following settings on the **Scaling** page are specific to the **ADC (Type A)** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.

- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the Scaling offset parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital In (Type B) settings

The following settings on the Scaling page are specific to the Digital In (Type B) I/O function.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- Disabled:

  The Data port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- Enabled:

  The output of the Data port is inverted: A low-high transition results in a 0 and vice versa.

Digital Crank/Cam Sensor settings

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Inductive Zero Voltage Detector

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Knock Sensor

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Status In

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Temperature

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model when using the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the three I/O types *Analog*, *Digital*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1554 Engine Control I/O Module determines the I/O functions that you can select:

- LED Out
- Sensor Supply
- Digital Out 1 (Type A) … Digital Out 40 (Type A)
- Digital Out 1 (Type B) … Digital Out 8 (Type B)

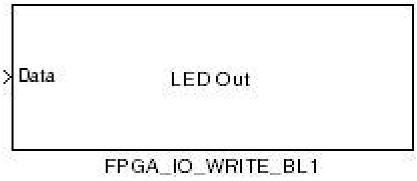
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out function.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block so you can connect it to a Simulink-based I/O environment model.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the FPGA status LED. The LED is located near the DS1514 ZIF I/O connector. Data type: UFix_1_0 ▪ 0: LED lights up green. ▪ 1: LED lights up orange.

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used (=1).

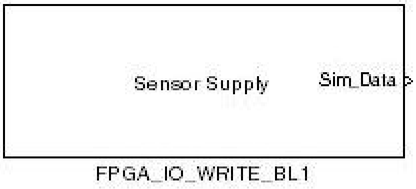
I/O mapping No external connection.

LED Out settings

None

Sensor Supply description

Block display If you select the Sensor Supply channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Output voltage: 13.14 V

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

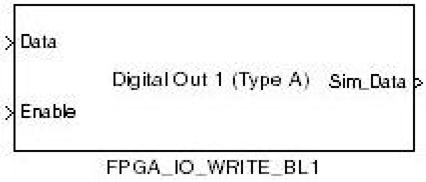
Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6 c6	VSENS+ VSENS–

Sensor Supply settings

Only common dialog settings. Refer to Common settings on page 586.

Digital Out (Type A) description

Block display If you select a Digital Out (Type A) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is set to the specified high-supply voltage (VDRIVE). The hardware output is driven only if the **Enable** port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Digital Outputs (MicroAutoBox II Hardware Reference 🕮).

Port	Description
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V ... 45 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Inport	Channel	Connector Pin	Signal
Data	1	L5	DigOut ch 1
	2	N2	DigOut ch 2
	3	D3	DigOut ch 3
	4	N5	DigOut ch 4
	5	M6	DigOut ch 5
	6	N3	DigOut ch 6
	7	D5	DigOut ch 7
	8	M2	DigOut ch 8
	9	L6	DigOut ch 9
	10	K2	DigOut ch 10
	11	C3	DigOut ch 11
	12	L2	DigOut ch 12
	13	G6	DigOut ch 13
	14	H2	DigOut ch 14
	15	C5	DigOut ch 15
	16	J2	DigOut ch 16
	17	F6	DigOut ch 17
	18	E2	DigOut ch 18
	19	B3	DigOut ch 19
	20	G2	DigOut ch 20
	21	E6	DigOut ch 21
	22	C2	DigOut ch 22
	23	B5	DigOut ch 23
	24	F2	DigOut ch 24
	25	D6	DigOut ch 25
	26	A6	DigOut ch 26
	27	A3	DigOut ch 27
	28	D2	DigOut ch 28
	29	B6	DigOut ch 29
	30	A2	DigOut ch 30
	31	A5	DigOut ch 31
	32	B2	DigOut ch 32
	33	F5	DigOut ch 33
	34	N6	DigOut ch 34
	35	E3	DigOut ch 35
	36	E5	DigOut ch 36
	37	H3	DigOut ch 37
	38	M5	DigOut ch 38

Inport	Channel	Connector Pin	Signal
	39	G3	DigOut ch 39
	40	F3	DigOut ch 40

Digital Out (Type A) settings

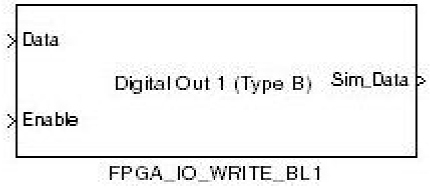
The following settings on the **Parameters** page are specific to the **Digital Out (Type A)** I/O function. For common dialog settings, refer to Common settings on page 586.

Simulated VDRIVE pin Lets you select the voltage for the simulated high-side switch for all digital output channels in the range 0 … 45000 mV in steps of 100 mV.

> **Note**
>
> You can specify the simulated voltage value only globally for all digital output channels.

Digital Out (Type B) description

Block display If you select a Digital Out (Type B) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high-supply voltage. The hardware output is driven only if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Digital I/O (Bidirectional) (MicroAutoBox II Hardware Reference 🕮).

Port	Description
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V … 3.3 V or 0 V … 5 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

You can use the same digital channel for input and output signals.

Outport	Channel	Connector Pin	Signal
Data	1	c3	DigIO ch1
	2	b5	DigIO ch2
	3	b2	DigIO ch3
	4	c5	DigIO ch4
	5	c4	DigIO ch5
	6	c2	DigIO ch6
	7	a2	DigIO ch7
	8	Z2	DigIO ch8

Digital Out (Type B) settings

The following settings on the **Parameters** page are specific to the **Digital Out (Type B)** I/O function. For common dialog settings, refer to Common settings on page 586.

High supply Lets you select the voltage for the high-side switch (3.3 V or 5 V) for all digital output channels.

> **Note**
>
> You can specify the high-supply voltage value only globally for all digital output channels.

Related topics

References

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the Enable FPGA test access and scaling parameter on the FPGA Access page of the FPGA_SETUP_BL block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the Scaling page are common to the I/O functions that you can select on the Unit page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital Out (Type A and Type B) settings

The following settings on the Scaling page are specific to the Digital Out (Type A) and Digital Out (Type B) I/O functions.

Invert polarity Lets you adapt the electrical output signal:

- Disabled:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.

- Enabled:
  If driven with 1, the hardware outputs a low-level signal.
  If driven with 0, the hardware outputs a high-level signal.

LED Out settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Sensor Supply settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........ 593
FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)........ 584
Parameters Page (FPGA_IO_WRITE_BL)........ 585

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)........ 584
Parameters Page (FPGA_IO_WRITE_BL)........ 585
Scaling Page (FPGA_IO_WRITE_BL)........ 592

FPGA_INT_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)

Purpose

To configure the FPGA interrupt channel when using the FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

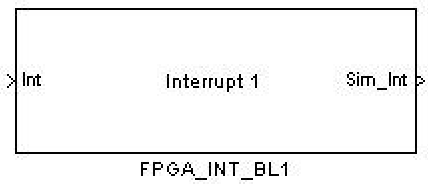
Description

The FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The following illustration shows the block display with the optional simulation port:



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function-call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The **Sim_Int** outport is added to the block so you can connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL).......... 595
FPGA_INT_BL.......... 62
FPGA_INT_BL (FPGA1401Tp1 with Engine Control I/O Module Settings).......... 594

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the

text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1401Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_INT_BL........ 62
FPGA_INT_BL (FPGA1401Tp1 with Engine Control I/O Module Settings)........ 594
Parameters Page (FPGA_INT_BL)........ 594

RTI Block Settings for the FPGA1403Tp1 with Multi-I/O Module Frameworks

Introduction

The block dialogs provide hardware-specific settings after you load one of the following MicroAutoBox III frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings)

Purpose

To configure read access to intermodule bus data in the FPGA model after you load one of the following frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)..598
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)..606
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To write data from the processor model to the FPGA model via the board-specific bus.

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)..607
To configure write access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1403Tp1 with Multi-I/O Module frameworks provide the following access types that you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is read from an intermodule bus register. 128 registers are available with a data width of

32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element.

If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.

Ungrouped registers are automatically combined into one register group with group ID *Ungrouped* to optimize data transfer. Since register groups can be only accessed by one task, you have to explicitly group registers which are used by different tasks.

- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is read from an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.

- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

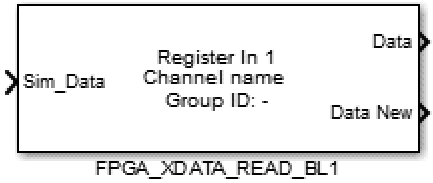
The fraction width is displayed in the Binary point position (or fraction width) setting.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, Sim_Data. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from a intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a

Port	Description
	new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

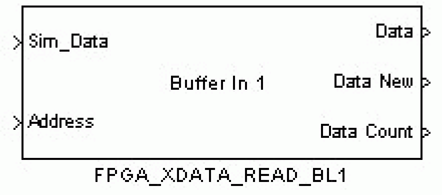
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 599.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater

Port	Description
	than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 599.

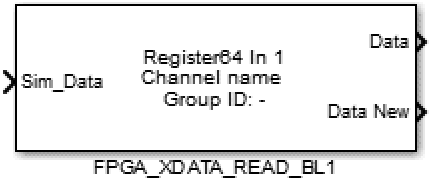
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 599.

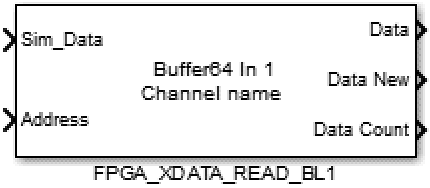
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

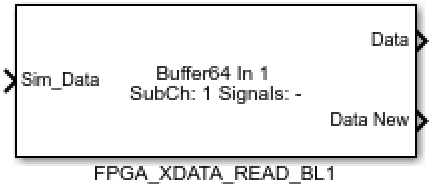
Buffer64 In description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0

Port	Description
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is copied via **Copy bus topology from gcb** on the **Parameters** page. The maximum bit wide is 64 bits. The resolution of the data types is restricted to 53 bits, because the data type of the received bus signals from the processor application is double and the block converts the signals to the signal data types.
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 599.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Copy bus topology from gcb Lets you copy an existing FPGA bus topology from the selected Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** block.

You cannot copy a bus topology from the processor model, because these topologies do not include the FPGA data types. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port.

This button is clickable only if the bus transfer mode is enabled.

Related topics

References

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_READ_BL..40
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings)....................................597
Parameters Page (FPGA_XDATA_READ_BL)...598

FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)

Purpose

To configure write access to intermodule bus data in the FPGA model after you load one of the following frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_WRITE_BL)...608
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)...616
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)...48
To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)...49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

PROC_XDATA_READ_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To read data in the processor model that comes from the FPGA model via the board-specific bus.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1403Tp1 with Multi-I/O Module frameworks provide the following access types that you can select in the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an intermodule bus register. 128 registers are available with a data width of 32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.

- floating-point

  The values of the **Data** inport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

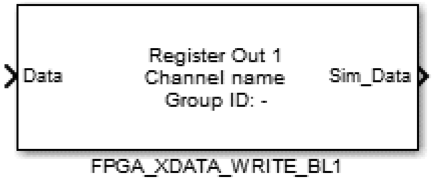
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings.

Port	Description
	Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

> **Note**
>
> The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

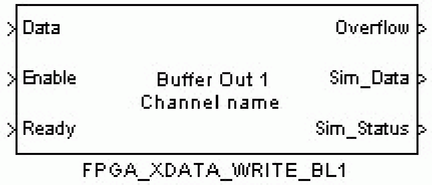
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 608.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings.

Port	Description
	Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1, if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data is lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 608.

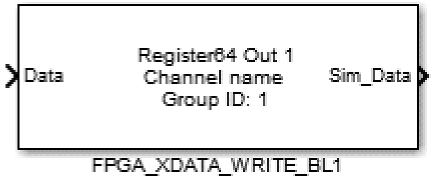
Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	**Description**
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

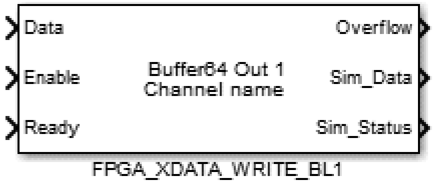
Register64 Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 608.

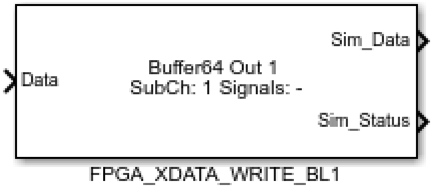
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

Buffer64 Out description

Block display If you specify the buffer access type, the block display changes. The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format

Port	Description
	XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is set via **Analyze bus topology of input** on the **Parameters** page. The maximum bit width is 64 bits. The resolution of the data type is restricted to 53 bits, because the block converts all data values to double for transmission.
Enable	Specifies the current valid **Data** port value. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready o send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size or the number of bus signals.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle.
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector.

Port	Description
	- Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. - Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 608.

Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Analyze bus topology of input Lets you set the **Data** inport to the bus topology of the connected Simulink bus.

If clicked, the RTI FPGA Programming Blockset analyzes the connected Simulink bus and sets the **Data** port to a matching bus topology. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port.

This button is clickable only if the bus transfer mode is enabled.

Bus data transmission method Lets you select the method for transmitting data to the processor application if the bus transfer mode is enabled:

- Synchronous to Read_Req method

  Select this method to transmit data that is captured synchronously to the processor task.

  The FPGA application writes data to the swinging buffer when the processor application makes a read request. After the data is written to the buffer, the buffer swings and sends the data to the processor application.
- Free running method

  Select this method if the transmission time is crucial.

The FPGA application continuously writes data to the swinging buffer. A read request of the processor application immediately transmits the last complete data set of the swinging buffer to the processor application.

The bus data transmission method is selectable only for subchannel 1 and the bus transfer mode must be enabled. The selection applies to all subchannels of the selected channel.

For instructions, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 📖).

Related topics

HowTos

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖)

References

Description Page (FPGA_XDATA_WRITE_BL)..616
FPGA_XDATA_WRITE_BL...45
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)...............................607

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_WRITE_BL...45
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)...............................607
Parameters Page (FPGA_XDATA_WRITE_BL)..608

FPGA_IO_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model after you load one of the following frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides different I/O types, which you can select on the Unit page of the block. The number of the available physical connections on the DS1552 or DS1552B1 Multi-I/O Module determines the I/O functions that you can select:

- Analog In 10 Channel 1 … Analog In 10 Channel 8 (only DS1552)
- Analog In 11 Channel 1 … Analog In 11 Channel 8 (only DS1552B1)
- Analog In 12 Channel 1 … Analog In 12 Channel 16
- Digital Crank/Cam Sensor 1 ... Digital Crank/Cam Sensor 3

  To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- Digital In 5 Channel 1 … Digital In 5 Channel 16
- Digital InOut 6 Channel 1 … Digital InOut 6 Channel 8
- Inductive Zero Voltage Detector

  To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.
- Status In
- Temperature

  To provide read access to the FPGA's die temperature.

Common settings

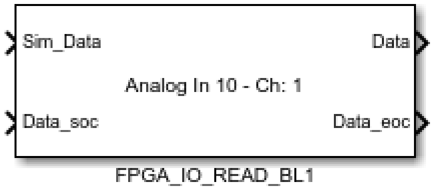
The following settings on the Parameters page are common to the I/O functions that you can select on the Unit page of the FPGA_IO_READ_BL block, except for the Status In and Temperature function.

Enable simulation port Lets you enable an inport for offline simulation data. The Sim_Data inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Analog In 10 description

Hardware support Only the DS1552 Multi-I/O Module provides the Analog In 10 channels.

Block display If you select an Analog In 10 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: 0 V … +5 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The `Data_eoc` outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	X3	Analog In Channel 1 Signal
		X4	Analog In Channel 1 Reference
	2	W3	Analog In Channel 2 Signal
		W4	Analog In Channel 2 Reference
	3	V3	Analog In Channel 3 Signal
		V4	Analog In Channel 3 Reference
	4	U3	Analog In Channel 4 Signal
		U4	Analog In Channel 4 Reference
	5	H3	Analog In Channel 5 Signal
		H4	Analog In Channel 5 Reference
	6	G3	Analog In Channel 6 Signal
		G4	Analog In Channel 6 Reference
	7	F3	Analog In Channel 7 Signal
		F4	Analog In Channel 7 Reference
	8	E3	Analog In Channel 8 Signal
		E4	Analog In Channel 8 Reference

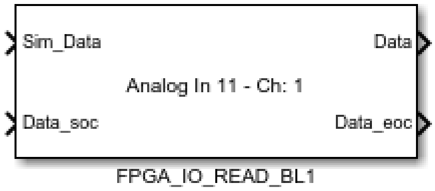
Analog In 10 settings

Only common dialog settings. Refer to Common settings on page 618.

Analog In 11 description

Hardware support Only the DS1552B1 Multi-I/O Module provides the Analog In 11 channels.

Block display If you select an Analog In 11 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1 Input voltage range: -10 V ... +10 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The `Data_eoc` outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

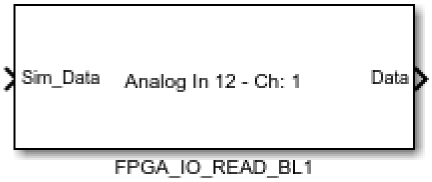
Outport	Channel	Connector Pin	Signal
Data	1	X3	Analog In Channel 1 Signal
		X4	Analog In Channel 1 Reference
	2	W3	Analog In Channel 2 Signal
		W4	Analog In Channel 2 Reference
	3	V3	Analog In Channel 3 Signal
		V4	Analog In Channel 3 Reference
	4	U3	Analog In Channel 4 Signal
		U4	Analog In Channel 4 Reference
	5	H3	Analog In Channel 5 Signal
		H4	Analog In Channel 5 Reference
	6	G3	Analog In Channel 6 Signal
		G4	Analog In Channel 6 Reference
	7	F3	Analog In Channel 7 Signal
		F4	Analog In Channel 7 Reference
	8	E3	Analog In Channel 8 Signal
		E4	Analog In Channel 8 Reference

Analog In 11 settings

Only common dialog settings. Refer to Common settings on page 618.

Analog In 12 description

Block display If you select an Analog In 12 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–10 V ... +10 V	The output port range is: –32768 … +32767

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1 Input voltage range: -10 V ... +10 V
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: Fix_16_0 Range: -32768 … +32767 Update rate: 0.2 Msps

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

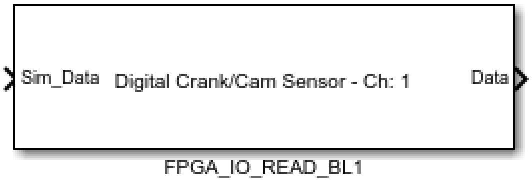
Outport	Channel	Connector Pin	Signal
Data	1	b2	Analog In 12 Channel 1 Signal
	2	a2	Analog In 12 Channel 2 Signal
	3	Z2	Analog In 12 Channel 3 Signal
	4	Y2	Analog In 12 Channel 4 Signal
	5	X2	Analog In 12 Channel 5 Signal
	6	W2	Analog In 12 Channel 6 Signal
	7	V2	Analog In 12 Channel 7 Signal
	8	U2	Analog In 12 Channel 8 Signal
	9	M2	Analog In 12 Channel 9 Signal
	10	L2	Analog In 12 Channel 10 Signal
	11	K2	Analog In 12 Channel 11 Signal
	12	J2	Analog In 12 Channel 12 Signal
	13	H2	Analog In 12 Channel 13 Signal
	14	G2	Analog In 12 Channel 14 Signal
	15	F2	Analog In 12 Channel 15 Signal
	16	E2	Analog In 12 Channel 16 Signal

Analog In 12 settings

Only common dialog settings. Refer to Common settings on page 618.

Digital Crank/Cam Sensor description

Block display If you select a Digital Crank/Cam Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Outputs the status of the crank/cam sensor. Data type: UFix_1_0 ▪ 0: The input signal is lower than the **Low threshold voltage** parameter. ▪ 1: The input signal is higher than the **High threshold voltage** parameter. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	R3	Digital In 6 Channel 1 +
		R4	Digital In 6 Channel 1 -
	2	B3	Digital In 6 Channel 2 +
		B4	Digital In 6 Channel 2 -
	3	A3	Digital In 6 Channel 3 +
		A4	Digital In 6 Channel 3 -

Digital Crank/Cam Sensor settings

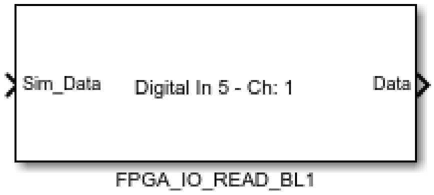
The following settings on the Parameters page are specific to the Digital Crank/Cam Sensor I/O function. For common dialog settings, refer to Common settings on page 618.

Low threshold voltage (-40000 mV ... +40000 mV) Lets you set the low threshold level for the selected digital input channel. Below this level a logical 0 is detected, above this level a logical 1 is detected, if the high threshold voltage was crossed before.

High threshold voltage (-40000 mV ... +40000 mV) Lets you set the high threshold level for the selected digital input channel. The logical 1 is output, if this level is crossed and stays 1 until the signal falls below the low threshold level.

Digital In 5 description

Block display If you select a Digital In 5 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1 Threshold level: ▪ 3.6 V for low-high transition ▪ 1.2 V for high-low transition

Port	Description
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the threshold voltage of a high-low transition. ▪ 1: Input voltage of the channel is higher than or equal to the threshold voltage of a low-high transition. Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital In 5 Characteristics (MicroAutoBox III Hardware Installation and Configuration).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

> **Note**
>
> Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

> **Note**
>
> If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
>
> To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

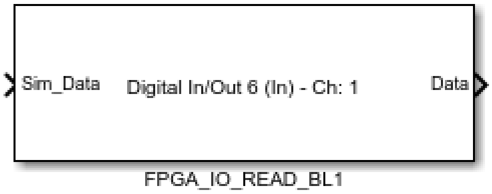
Outport	Channel	Connector Pin	Signal
Data	1	V5	Digital In 5 Channel 1 Signal
	2	U5	Digital In 5 Channel 2 Signal
	3	U6	Digital In 5 Channel 3 Signal
	4	T2	Digital In 5 Channel 4 Signal
	5	T3	Digital In 5 Channel 5 Signal
	6	T4	Digital In 5 Channel 6 Signal
	7	T5	Digital In 5 Channel 7 Signal
	8	T6	Digital In 5 Channel 8 Signal
	9	S2	Digital In 5 Channel 9 Signal
	10	S3	Digital In 5 Channel 10 Signal
	11	S5	Digital In 5 Channel 11 Signal
	12	R2	Digital In 5 Channel 12 Signal
	13	R5	Digital In 5 Channel 13 Signal
	14	R6	Digital In 5 Channel 14 Signal
	15	P5	Digital In 5 Channel 15 Signal
	16	P6	Digital In 5 Channel 16 Signal

Digital In 5 settings

Only common dialog settings. Refer to Common settings on page 618.

Digital InOut 6 (In) description

Block display If you select a Digital InOut 6 (In) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal.

Port	Description
	Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 1 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage. Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital In/Out 6 Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the corresponding minimum or maximum value.

Note

Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.

To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	N2	Digital InOut 6 Channel 1 Signal
	2	N3	Digital InOut 6 Channel 2 Signal
	3	N4	Digital InOut 6 Channel 3 Signal
	4	N5	Digital InOut 6 Channel 4 Signal
	5	N6	Digital InOut 6 Channel 5 Signal
	6	M5	Digital InOut 6 Channel 6 Signal
	7	M6	Digital InOut 6 Channel 7 Signal
	8	L4	Digital InOut 6 Channel 8 Signal

You can use the same digital channel for input and output signals.

Digital InOut 6 (In) settings

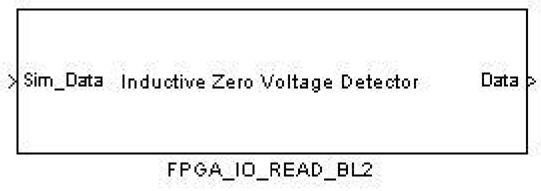
The following settings on the Parameters page are specific to the Digital InOut 6 (In) I/O function. For common dialog settings, refer to Common settings on page 618.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1,000 mV … 7,500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

Inductive Zero Voltage Detector description

Block display If you select a Inductive Zero Voltage Detector channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Detects the zero crossing points of the analog signals. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle. Data type: UFix_1_0 ▪ 0: No zero crossing. ▪ 1: Zero crossing is detected. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, **Data** outport is saturated to minimum or maximum range value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

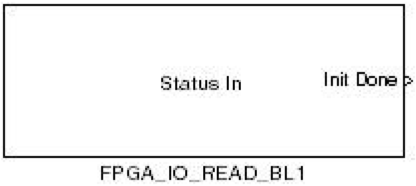
Outport	Connector Pin	Signal
Data	P3	Digital In 7 Channel 1 +
	P4	Digital In 7 Channel 1 -

Inductive Zero Voltage Detector settings

Only common dialog settings. Refer to Common settings on page 618.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for initialization sequence status information:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence has finished.

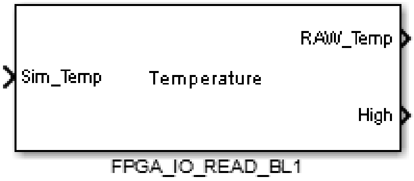
I/O mapping No external connection.

Status In settings

None

Temperature

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C The range can be exceeded. The values are then saturated to the minimum or maximum values.

Port	Description
Output	
RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature. Equation for calculating the die temperature: `Temperature [°C] = (float)(Temperature[hex] & 0xFFF0) · 503.975 / 65536 - 273.15` Data type: UFix_16_0 Data width: 1 Value range: 0 ... 65536
High	Outputs a flag if the FPGA's die temperature exceeds 105 °C. To reset the flag, the die temperature must fall below 85 °C. Data type: UFix_1_0 ▪ 0: Die temperature does not exceed 105 °C. ▪ 1: Die temperature exceeds 105 °C.

Note

A high ambient temperature and an FPGA application with a very high FPGA utilization and/or toggle rate increase the FPGA die temperature (internal chip temperature). If the die temperature exceeds 105 °C, the FPGA might work incorrectly.

You can decrease the temperature by reducing the FPGA's toggle rate (e.g., by using clock enable) or by reducing the utilization of the FPGA resources.

If the die temperature exceeds 125 °C, the FPGA resets itself. The reset stays active until the die temperature falls below 85 °C and you restart MicroAutoBox III or reload the user application.

I/O mapping No external connection.

Temperature settings

Only common dialog settings. Refer to Common settings on page 618.

Related topics

References

Description Page (FPGA_IO_READ_BL) 636
FPGA_IO_READ_BL 50
FPGA_IO_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings) 617
Scaling Page (FPGA_IO_READ_BL) 633

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

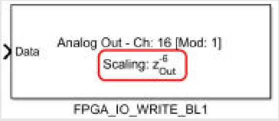
Analog in settings

The following settings on the **Scaling** page are specific to the **Analog In 10**, **Analog In 11**, and **Analog In 12** I/O functions.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 📖).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.

- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum**

value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the Scaling factor parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the Scaling offset parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital in settings

The following settings on the Scaling page are specific to the Digital In 5 and Digital InOut 6 (In) I/O functions.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- Disabled:

  The Data port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- Enabled:

  The output of the Data port is inverted: A low-high transition results in a 0 and vice versa.

Digital Crank/Cam Sensor settings

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Inductive Zero Voltage Detector settings

The Scaling page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Status In settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Temperature settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_READ_BL).. 636
FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings).. 617
Parameters Page (FPGA_IO_READ_BL).. 618

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_IO_READ_BL.. 50
FPGA_IO_READ_BL (FPGA1403Tp1 with Multi-I/O Module Settings).. 617
Parameters Page (FPGA_IO_READ_BL).. 618
Scaling Page (FPGA_IO_READ_BL).. 633

FPGA_IO_WRITE_BL (FPGA1403Tp1 with Multi-I/O Module Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model after you load one of the following frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the three I/O types *Analog*, *Digital*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1552 Multi-I/O Module determines the I/O functions that you can select:

- Analog Out 13 Channel 1 … Analog Out 13 Channel 4
- Digital Out 5 Channel 1 … Digital Out 5 Channel 16
- Digital InOut 6 Channel 1 … Digital InOut 6 Channel 8
- LED Out
- Sensor Supply
- UART 3 (RS232) Channel 1 … UART 3 (RS232) Channel 2
- UART 3 (RS422/485) Channel 1 … UART 3 (RS422/485) Channel 2

Common settings

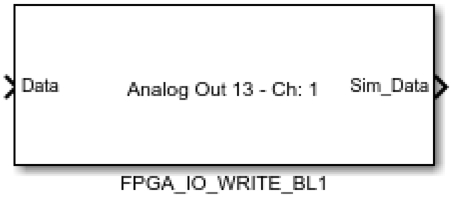
The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out and Sensor Supply functions.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block to connect it to a Simulink-based I/O environment model.

If you have selected one of the UART functions, this setting is replaced by function-specific simulation settings.

Analog Out 13 description

Block display If you have selected an Analog Out 13 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog output voltage and the input of the block is:

Output Voltage Range	Simulink Input
0 V … +5 V	The input port range is: 0 … +65535

The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. Data type: UFix_16_0 Output voltage range: 0 … +65535 The range can be exceeded, and saturation is performed to a minimum or maximum value. Hardware update rate: 2.1 Msps If the values are updated at a higher FPGA model rate, intermediate values are not updated by the DAC.
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Output voltage range: 0 V … +5 V

Multiple clock domain support This block does not support multiple clock domains.

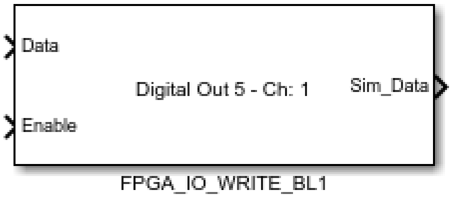
I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Inport	Channel	Connector Pin	Signal
Data	1	c2	Analog Out 13 Channel 1 Signal
	2	c3	Analog Out 13 Channel 2 Signal
	3	c4	Analog Out 13 Channel 3 Signal
	4	c5	Analog Out 13 Channel 4 Signal

Analog Out 13 settings Only common dialog settings. Refer to Common settings on page 638.

Digital Out 5 description

Block display If you select a Digital Out 5 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is set to the specified high supply voltage (VDRIVE). The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital Out 5 Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V ... 45 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Inport	Channel	Connector Pin	Signal
Data	1	F5	Digital Out 5 Channel 1 Signal
	2	E5	Digital Out 5 Channel 2 Signal
	3	E6	Digital Out 5 Channel 3 Signal
	4	D2	Digital Out 5 Channel 4 Signal
	5	D3	Digital Out 5 Channel 5 Signal
	6	D4	Digital Out 5 Channel 6 Signal
	7	D5	Digital Out 5 Channel 7 Signal
	8	D6	Digital Out 5 Channel 8 Signal
	9	C2	Digital Out 5 Channel 9 Signal
	10	C3	Digital Out 5 Channel 10 Signal
	11	C5	Digital Out 5 Channel 11 Signal
	12	B2	Digital Out 5 Channel 12 Signal
	13	B5	Digital Out 5 Channel 13 Signal
	14	B6	Digital Out 5 Channel 14 Signal
	15	A5	Digital Out 5 Channel 15 Signal
	16	A6	Digital Out 5 Channel 16 Signal

Digital Out 5 settings

The following settings on the **Parameters** page are specific to the **Digital Out 5** I/O function. For common dialog settings, refer to Common settings on page 638.

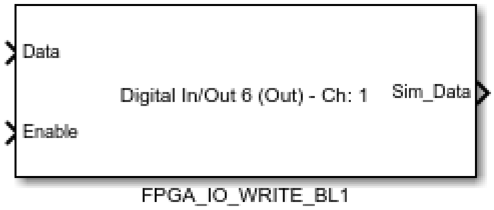
Simulated VDRIVE pin Lets you select the voltage for the simulated high side switch for all digital output channels in the range 0 … 45000 mV.

Note

You can specify the simulated voltage value only globally for all digital output channels.

Digital InOut 6 (Out) description

Block display If you select a Digital InOut 6 (Out) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high supply voltage. The hardware output is only driven if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is smaller than the update rate. For information on the electrical characteristics of the DS1552 Multi-I/O Module, refer to Digital In/Out 6 Characteristics (MicroAutoBox III Hardware Installation and Configuration).
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V … 3.3 V or 0 V … 5 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	N2	Digital InOut 6 Channel 1 Signal
	2	N3	Digital InOut 6 Channel 2 Signal
	3	N4	Digital InOut 6 Channel 3 Signal
	4	N5	Digital InOut 6 Channel 4 Signal
	5	N6	Digital InOut 6 Channel 5 Signal
	6	M5	Digital InOut 6 Channel 6 Signal
	7	M6	Digital InOut 6 Channel 7 Signal
	8	L4	Digital InOut 6 Channel 8 Signal

You can use the same digital channel for input and output signals.

Digital InOut 6 (Out) settings

The following settings on the Parameters page are specific to the Digital InOut 6 (Out) I/O function. For common dialog settings, refer to Common settings on page 638.

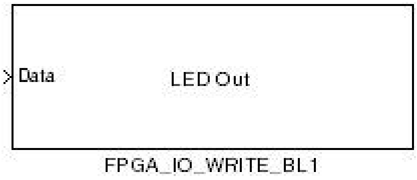
High supply Lets you select the voltage for the high side switch (3.3 V or 5 V) for all digital output channels.

Note

You can specify the high supply voltage value only globally for all digital output channels.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the FPGA status LED near the DS1514 ZIF I/O connector. Data type: UFix_1_0 ▪ 0: LED lights up green. ▪ 1: LED lights up orange.

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used (=1).

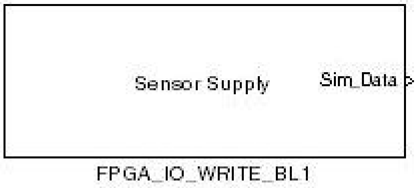
I/O mapping No external connection.

LED Out settings

None

Sensor Supply description

Block display If you select the Sensor Supply channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Output voltage: 2 V … 20 V according to the specified supply voltage

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6	VSENS-
		c6	VSENS+

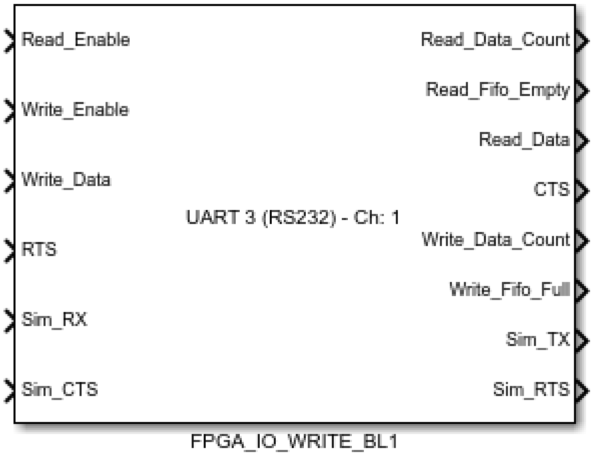
Sensor Supply settings

The following settings on the **Parameters** page are specific to the **Sensor Supply** I/O function. For common dialog settings, refer to Common settings on page 638.

Supply voltage Lets you enter the supply voltage a connected sensor is to be driven with in the range 2000 mV … 20000 mV in steps of 100 mV.

UART 3 (RS232) description

Block display If you select an UART 3 (RS232) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next **Read_Enable** signal. Before you read data from the RX FIFO buffer, you should check the **Read_Fifo_Empty** signal not to be set. The **Read_Fifo_Empty** signal switches one clock cycle after the RX FIFO value has been read. Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle. Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Port	Description
	The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud rate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
RTS	Specifies the Ready-To-Send (RTS) signal. The RTS/CTS handshake is handled by the user, the RTS signal is just passed through and adapted to the physical layer. Data type: UFix_1_0 Data width: 1 The hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive).
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the **Parameters** page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_CTS	Simulates the Clear-To-Send (CTS) hardware signal. Available only if **Enable simulation CTS port** is set on the **Parameters** page. The RTS/CTS handshake is handled by the user. The **Sim_CTS** signal is just passed through to CTS. Data type: Double Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Port	Description
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries. If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible. The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
CTS	Outputs the state of the Clear-To-Send (CTS) hardware port. RTS/CTS handshake is handled by the user. CTS is just passed through with conversion to logical 1 and 0. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: CTS inactive ▪ 1: CTS active The CTS hardware port is synchronously running to the UART clock defined by the UART baud rate. The hardware port has voltage levels of +6 V (active, logical high) and -6 V (inactive). Range exceeding is not possible.

Port	Description
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already. Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the TX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the **Parameters** page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Sim_RTS	Simulates the Ready-To-Send (RTS) hardware signal. Available only if **Enable simulation RTS port** is set on the **Parameters** page. The signal is in logical format and only passed through to the RTS signal. Data type: Double Data width: 1 Range: ▪ 0: RTS inactive ▪ 1: RTS active Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector.

Inport	Connector Pin	Signal
UART 1 (RS232)		
Write_Data	a5	TX1
RTS	a6	RTS1
Read_Data	b5	RX1
CTS	a4	CTS1
UART 2 (RS232)[1]		
Write_Data	Z5	TX2
RTS	Z6	RTS2
Read_Data	Z3	RX2
CTS	Z4	CTS2

[1] To use UART 2, your DS1552 has to be modified by dSPACE.

UART 3 (RS232) settings

The following settings on the Parameters page are specific to the UART 3 (RS232) I/O function.

Baud rate Lets you select the baud rate in the range 50 … 1,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

> **Note**
>
> The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The Write_Data and Read_Data ports handle raw bits. If you want to use a parity bit with the Write_Data port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the

Read_Data port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

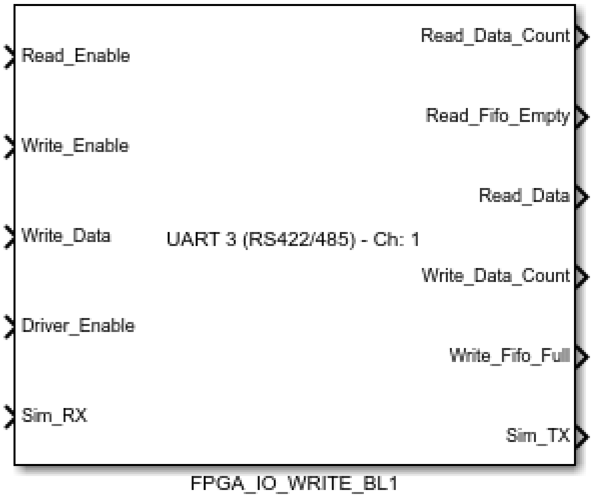
Enable simulation CTS port Lets you enable an inport for offline simulation data. The Sim_CTS inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Enable simulation RTS port Lets you enable an outport for offline simulation data. The Sim_RTS outport is added to the block to connect it to a Simulink-based I/O environment model.

UART 3 (RS422/485) description

Block display If you select an UART (RS422/485) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Read_Enable	Specifies to start receiving a value. After three clock cycles, the value is available and can be read from the RX FIFO buffer. The value remains valid until the next Read_Enable signal. Before you read data from the RX FIFO buffer, you should check the Read_Fifo_Empty signal not to be set. The Read_Fifo_Empty signal switches one clock cycle after the RX FIFO value has been read.

Port	Description
	Do not use the **Read_Data_Count** signal (Read_Data_Count < 0) to check the RX FIFO buffer, because it requires one additional clock cycle to get the count value. You can read one value per clock cycle from the UART. Data Type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Write_Enable	Specifies to start sending a value. The **Write_Data** value is written to the TX FIFO buffer, from which it is automatically send to the TX output pin of the I/O connector using the specified UART communication settings. **Write_Enable** must be set to 1 for only one clock cycle. Before you write data to the TX FIFO buffer, you should check the **Write_Fifo_Full** signal not to be set. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Do not use the **Write_Data_Count** signal (Write_Data_Count < 2047) to check the TX FIFO buffer, because it requires one additional clock cycle to get the count value. Data type: UFix_1_0 Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit. The hardware output port is driven with the values from the TX FIFO buffer. It is synchronously running to the UART clock defined by the UART baud rate. The hardware port has inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Write_Data	Specifies the value to be send. The **Write_Data** signal is transferred at each clock cycle with **Write_Enable** set to 1. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is possible. Then, only the lowest bits 5 … 9 will be used. Because of the unsigned data type, negative values will be interpreted as positive values and the saturation will always be towards the maximum value.
Driver_Enable	Specifies to enable the output driver in the transceiver for data transmission. If you use the UART (RS485/422) function in half-duplex mode, the output driver must be disabled while receiving data. Data type: UFix_1_0 Data width: 1
Sim_RX	Simulates an RX value in the RX FIFO buffer. Available only if **Enable simulation RX port** is set on the **Parameters** page. Data type: Double Data width: 1 The signal has to be in logical format with 1 as high and 0 as low, and not in the inverted values of the physical layer (-12 V high, +12 V low). The format

Port	Description
	of this serial bitstream has to correspond to the specified UART communication settings. Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.
Output	
Read_Data_Count	Outputs the number of new entries in the RX FIFO buffer. Two clock cycles are required to return the number of entries. If you only want to check whether a value is available in the RX FIFO buffer, use the **Read_Fifo_Empty** signal instead of this. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 The range can be exceeded, and saturation is performed to a minimum or maximum value.
Read_Data	Outputs the last read data from the RX FIFO buffer. The read_data is available after three clock cycles after the **Read_Enable** signal. The return value is 0, if the data is read before anything has been received by the RX hardware input. Data type: UFix_9_0 Data width: 1 Range: 0 … 511 Range exceeding is not possible. The hardware input receives serial data for the UART RX FIFO buffer using inverted voltage levels of -6 V (logical high) and +6 V (logical low).
Read_Fifo_Empty	Outputs the status of the RX FIFO buffer. If the status of the buffer is *not empty*, then you can start reading the data using the **Read_Enable** signal. The **Read_Fifo_Empty** signal switches one clock cycle after the FIFO value has been read. Do not use the **Read_Data_Count** signal to check the status of the buffer (Read_Data_Count>0), because this requires one additional clock cycle before its value is valid. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The RX FIFO buffer is not empty. ▪ 1: The RX FIFO buffer is empty. Range exceeding is not possible.
Write_Data_Count	Outputs the number of values in the TX FIFO buffer. The values in the TX FIFO buffer has not been sent already. Do not use the **Write_Data_Count** signal to check the status of the buffer (Write_Data_Count<2047), because this requires two clock cycles before its

Port	Description
	value is valid, instead of one clock cycle when using the **Write_Fifo_Full** signal. Data type: UFix_11_0 Data width: 1 Range: 0 … 2047 Range exceeding is not possible.
Write_Fifo_Full	Outputs the status of the TX FIFO buffer. You can use the signal to check the RX FIFO buffer before you start writing data to the buffer. The **Write_Fifo_Full** signal switches one clock cycle after the **Write_Enable** signal has been set. Data type: UFix_1_0 Data width: 1 Range: ▪ 0: The TX FIFO buffer is not full. ▪ 1: The TX FIFO buffer is full. Range exceeding is not possible.
Sim_TX	Simulates the TX hardware signal. Available only if **Enable simulation TX port** is set on the **Parameters** page. The signal is in logical format and not in inverted values from the physical layer (-6 V high, +6 V low). The format of the serial bitstream corresponds to the specified UART communication settings. Data type: Double Data width: 1 Range: ▪ 0: Low ▪ 1: High Range exceeding is possible (using input bit widths > 1) and will be cast to 1 bit by using only the lowest bit.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use one of the FPGA1403Tp1 with Multi-I/O Module frameworks. The signals are available at the DS1514 ZIF I/O connector. The mapping differs when using the UART (RS422/485) in full-duplex or half-duplex mode.

Full-duplex mode:

Inport	Connector Pin	Signal
UART 1 (RS422/485)		
Write_Data	a5	TX-1
	a6	TX+1
Read_Data	b5	RX-1
	a4	RX+1
UART 2 (RS422/485)[1]		
Write_Data	Z5	TX-2
	Z6	TX+2
Read_Data	Z3	RX-2
	Z4	RX+2

[1] To use UART 2, your DS1552 has to be modified by dSPACE.

Half-duplex mode:

Inport	Connector Pin	Signal
UART 1 (RS422/485)		
Write_Data	a5	BM1 (RX-1/TX-1)
	a6	BP1 (RX+1/TX+1)
Read_Data	b5	-[1]
	a4	-[1]
UART 2 (RS422/485)[2]		
Write_Data	Z5	BM2 (RX-2/TX-2)
	Z6	BP2 (RX+2/TX+2)
Read_Data	Z3	-[1]
	Z4	-[1]

[1] Do not connect, TX signals are available via BM and BP signals.
[2] To use UART 2, your DS1552 has to be modified by dSPACE.

UART 3 (RS422/485) settings

The following settings on the Parameters page are specific to the UART 3 (RS422/485) I/O function.

Baud rate Lets you select the baud rate in the range 50 … 10,000,000 baud (bits per second) from the given list.

Word length Lets you select the word length in the range 5 … 9 bits. The word length includes the number of data bits and the optional parity bit.

For example, if a message consists of 8 data bits and the parity bit, you have to set the word length to 9.

Note

The parity bit cannot be controlled via dialog setting. You have to consider an optional parity bit in your own model implementation.

The **Write_Data** and **Read_Data** ports handle raw bits. If you want to use a parity bit with the **Write_Data** port, you have to explicitly generate it and provide it as the last bit of the port. If you want to use a parity bit with the **Read_Data** port, you have to explicitly check it and read it as the last bit of the port.

Stop bits Lets you select the number of stop bits in the range 1, 1.5 and 2.

Mode Lets you select the mode for receiving messages.

- Full duplex mode

  You can simultaneously send and receive signals on the UART channel.
- Half duplex mode

  You can send or receive signals on the UART channel, but you cannot do both at the same time.

Termination Lets you enable an internal termination between RX- and RX+ and between TX- and TX+.

Setting	Meaning
Open	No termination
Terminated	The RX and TX signals are terminated via an internal 120 Ω resistor.

Enable simulation RX port Lets you enable an inport for offline simulation data. The Sim_RX inport is added to the block to connect it to simulation data coming from a Simulink-based I/O environment model.

Enable simulation TX port Lets you enable an outport for offline simulation data. The Sim_TX outport is added to the block to connect it to a Simulink-based I/O environment model.

Related topics

References

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

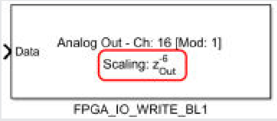
Analog out settings

The following settings on the **Scaling** page are specific to the **Analog Out 13** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.

- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

- Double:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the Scaling format setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the Data port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the Data port.

Binary point (fraction width) This setting depends on the Scaling format setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the Data port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the Data port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the Data port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the Scaling factor parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Scaling offset Lets you add a signal offset after the signal of the Data port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the Scaling offset parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation minimum value Lets you specify the minimum value to which the scaled Data inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the Saturation minimum value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the scaled Data inport signal is saturated before it is output via an analog output channel.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the Saturation maximum

value parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital outsettings

The following settings on the **Scaling** page are specific to the **Digital Out 5** and **Digital InOut 6 (out)** I/O functions.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:

  If driven with 1, the hardware outputs a high-level signal.

  If driven with 0, the hardware outputs a low-level signal.
- **Enabled**:

  If driven with 1, the hardware outputs a low-level signal.

  If driven with 0, the hardware outputs a high-level signal.

LED Out settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Sensor Supply settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

UART settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL (FPGA1403Tp1 with Multi-I/O Module Settings)

Purpose

To configure the FPGA interrupt channel after you load one of the following frameworks:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

Where to go from here

Information in this section

Information in other sections

Common settings

Unit Page (FPGA_INT_BL)..66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

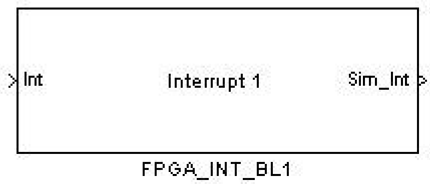
Description

The FPGA1403Tp1 with Multi-I/O Module frameworks provide 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The figure below shows the block display with the optional simulation port.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function-call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The Sim_Int outport is added to the block to connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)........ 661
FPGA_INT_BL........ 62
FPGA_INT_BL (FPGA1403Tp1 with Multi-I/O Module Settings)........ 660

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the I/O type that you selected on the corresponding **Unit** page. The information is either in the text

field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the I/O function that is provided by the following standard frameworks is included in this user documentation:

- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552)*
- *FPGA1403Tp1 (7K325) with Multi-I/O Module (DS1552B1)*

The description of the access type of customized frameworks or mounted piggybacks is available as a separate document.

Related topics

References

FPGA_INT_BL.. 62
FPGA_INT_BL (FPGA1403Tp1 with Multi-I/O Module Settings).................. 660
Parameters Page (FPGA_INT_BL).. 660

RTI Block Settings for the FPGA1403Tp1 with Engine Control I/O Module Framework

Introduction

The block dialogs provide hardware-specific settings after you load the FPGA1403Tp1 (7K325) with Engine Control Module (DS1554) framework for the MicroAutoBox III.

Where to go from here

Information in this section

FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)

Purpose

To configure read access to the local bus data in the FPGA model when using the DS1554 Engine Control I/O Module framework.

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_READ_BL)..664
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_READ_BL)..672
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_READ_BL)..43
To specify read access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_READ_BL)..44
To specify the general configuration for the FPGA board's processor bus read access.

Related RTI blocks

PROC_XDATA_WRITE_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To write data from the processor model to the FPGA model via the board-specific bus.

FPGA_XDATA_WRITE_BL..45
To implement write access to processor-bus data in the FPGA model.

FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)..673
To configure write access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_READ_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The DS1554 Engine Control I/O Module framework provides the following access types, which you can select on the **Unit** page of the block's dialog:

- Fixed-point format:

  Signed or unsigned data format with adjustable binary point position

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- 32-bit floating-point format:
  Single precision (IEEE 754 standard) data format with a fraction width of 24
- 64-bit floating-point format:
  Double precision (IEEE 754 standard) data format with a fraction width of 53

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_READ_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** outport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** outport.

- signed/unsigned

  The values of the **Data** outport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

  All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.
- floating-point

  The values of the **Data** outport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

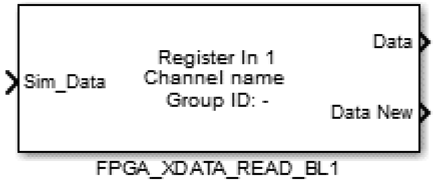
Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data**. By default, the sample time is inherited. You must only specify the sample time explicitly if the modeling situation inhibits to inherit the sample time, for example, if the corresponding

PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register In settings

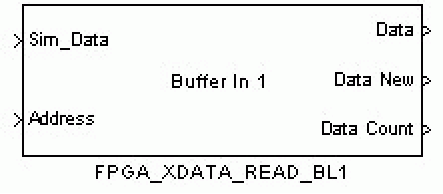
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 665.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

Buffer In description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0

Port	Description
Output	
Data	Outputs a 32-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the Address inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the Data outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer In settings

The following dialog settings are specific for the Buffer access type. For common dialog settings, refer to Common settings on page 665.

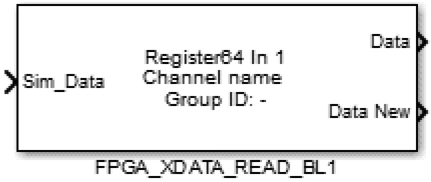
Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the Address inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Register64 In description

Block display If you specify the register access type, the block display changes. The simulation ports are optionally displayed.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Data New	Outputs a flag that indicates the changes of the register status. If the flag changes from 0 to 1 and then to 0 again, the requested register contains a new value. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 In settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 665.

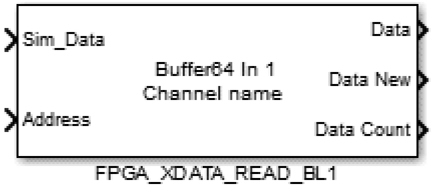
Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are read from the intermodule bus sequentially and then provided to the FPGA application simultaneously.

Specify 0 for ungrouped read access.

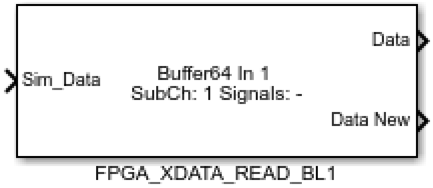
Buffer64 In description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation port is displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Sim_Data	Simulates an intermodule bus data exchange including floating-point to fixed-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768
Address	Specifies an element in the buffer you want to read. The block requires 3 clock cycles to update the value of the **Data** outport according to the specified address. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum port range depends on the specified buffer size in the **Parameters** page. The valid port range depends on the number of elements currently in the buffer (see **Data Count** outport). If you request data from an address that is greater than the **Data Count** value, the output of the **Data** outport is undefined. The first element of a buffer is addressed by 0. Data type: UFix_16_0

Port	Description
Output	
Data	Outputs a 64-bit data value to be read from an intermodule bus buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is copied via **Copy bus topology from gcb** on the **Parameters** page. The maximum bit wide is 64 bits. The resolution of the data types is restricted to 53 bits, because the data type of the received bus signals from the processor application is double and the block converts the signals to the signal data types.
Data New	Outputs a flag that indicates the changes of the buffer status. If the flag changes from 0 to 1 and then to 0 again, the requested buffer contains new values and is ready to be read. The flag is set to 1 only within one clock cycle. Data type: UFix_1_0
Data Count	Outputs the number of elements in the current buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. The maximum range depends on the specified buffer size. You can use the value to define the valid range for the **Address** inport of 0 … (Data Count -1). Data type: UFix_16_0

Note

The range can be exceeded for the **Data** outport. The outport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Buffer64 In settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 665.

Buffer size Lets you specify the size of the buffer in the range 1 … 32768. The maximum range of the **Address** inport depends on the buffer size.

Note

The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Copy bus topology from gcb Lets you copy an existing FPGA bus topology from the selected Simulink **Bus Creator** block, subsystem inport block, or subsystem outport block to the **Data** port of the **Buffer64 In** block.

You cannot copy a bus topology from the processor model, because these topologies do not include the FPGA data types. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 📖).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port.

This button is clickable only if the bus transfer mode is enabled.

Related topics

References

Description Page (FPGA_XDATA_READ_BL) 672
FPGA_XDATA_READ_BL 40
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings) 663

Description Page (FPGA_XDATA_READ_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_READ_BL 40
FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings) 663
Parameters Page (FPGA_XDATA_READ_BL) 664

FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)

Purpose

To configure write access to intermodule bus data in the FPGA model when using the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Parameters Page (FPGA_XDATA_WRITE_BL)..674
To specify the data format and specific settings for the selected access type.

Description Page (FPGA_XDATA_WRITE_BL)..682
To provide detailed information about the selected access type.

Information in other sections

Common settings

Block Description (FPGA_XDATA_WRITE_BL)..48
To specify write access by the FPGA model to the processor model via processor bus.

Unit Page (FPGA_XDATA_WRITE_BL)..49
To specify the general configuration for the FPGA board's processor bus write access.

Related RTI blocks

PROC_XDATA_READ_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To read data in the processor model that comes from the FPGA model via the board-specific bus.

FPGA_XDATA_READ_BL..40
To implement read access to processor-bus data in the FPGA model.

FPGA_XDATA_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)..663
To configure read access to intermodule bus data in the FPGA model.

Parameters Page (FPGA_XDATA_WRITE_BL)

Purpose

To specify the data format and specific settings for the selected access type.

Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides the following access types, which you can select on the **Unit** page of the block's dialog:

- Register/Register64

  If you select **Register** or **Register64** as the access type, the data is written to an intermodule bus register. 128 registers are available with a data width of 32 bits each and 128 registers with a data width of 64 bits each. The values are transmitted element by element. If you want to access data from several registers simultaneously, you can group these registers by specifying the same group identifier for them.
- Buffer/Buffer64

  If you select **Buffer** or **Buffer64** as the access type, the data is written to an intermodule bus buffer. 32 buffers are available that provides elements with a data width of 32 bits each and 32 buffers that provides elements with a data width of 64 bits each. Each buffer has a variable buffer size of 1 up to 32768 elements.

There are settings that are common to both access types and settings that are specific to each access type.

For further information, refer to Details on the access types on page 22.

Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_XDATA_WRITE_BL** block.

Binary point position (or fraction width) This setting depends on the format selected in the **Format** setting.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format.

  You can specify the binary point position of the 32-bit value in the range 0 … 32 and the binary point position of the 64-bit value in the range 0 … 64. 0 represents the lowest bit position, 32 or 64 the highest bit position.
- floating-point

  The values of the **Data** inport are in floating-point format. The fraction width is displayed.

Format Lets you select the data format of the **Data** inport.

- signed/unsigned

  The values of the **Data** inport are in fixed-point format with or without one bit reserved for the sign.

  You can specify the binary point position in the **Binary point position (or fraction width)** setting.

All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits.

- floating-point

  The values of the **Data** inport are in floating-point format.

  The 32-bit data value supports the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).

  The 64-bit data value supports the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

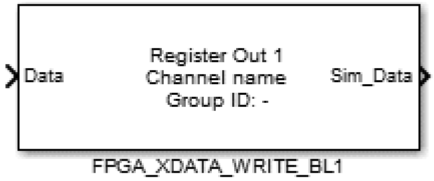
  The fraction width is displayed in the **Binary point position (or fraction width)** setting.

Enable simulation data port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block so you can connect it to any Simulink signal used for simulation.

Simulation ports sample time Lets you specify the sample time of the ports used for simulation, for example, **Sim_Data** and **Sim_Status**. By default, the sample time is inherited. You must only specify the sample time explicitly, if the modeling situation inhibits to inherit the sample time, for example, if the corresponding PROC block is part of an asynchronous task or is used for multiple access to the same channel.

Register Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double

Port	Description
	Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

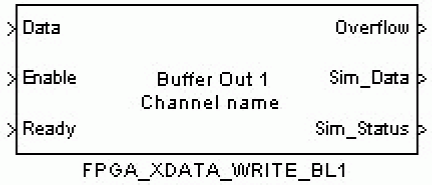
Register Out settings

The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 674.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values therefore form a consistent data group that is written to the intermodule bus.

Buffer Out description

Block display If you specify the buffer access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 32-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_32_<Binary point position>/Fix_32_<Binary point position> ▪ Floating-point format XFloat_8_24
Enable	Specifies the current valid **Data** port value. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.

Port	Description
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready to send, even if it is not completely filled. The buffer is switched and the data values are accessible via the intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer**, and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values. ▪ Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 674.

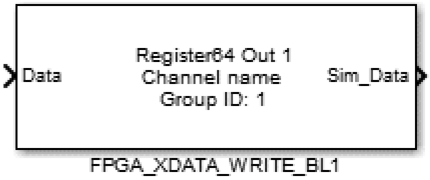
Buffer size Lets you specify the size of the buffer in the range 1 … 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 32-bit in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you enable the simulation data port. The **Sim_Status** outport is added to the block.

Register64 Out description

Block display If you specify the register access type, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in register access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus register. The data format depends on the related dialog settings. Data type: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> All 64-bit fixed-point data types are converted to double. Therefore, the fixed-point resolution of fixed-point data types is restricted to 53 bits. ▪ Floating-point format XFloat_11_53
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1

Note

The range can be exceeded for the **Data** inport. The inport's value is then cast to the specified data format. This might cause truncation and bit wrapping if insufficient bits are used before and after the binary point position to represent the input value.

Register64 Out settings

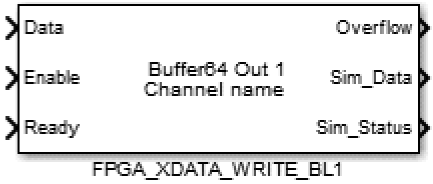
The following dialog settings are specific for the **Register** access type. For common dialog settings, refer to Common settings on page 674.

Register group ID Lets you specify a number in the range 1 … 63 to create register groups. All registers – 32-bit and 64-bit registers – which you specified with the same group ID are sampled simultaneously in the FPGA application. The values form therefore a consistent data group that is written to the intermodule bus.

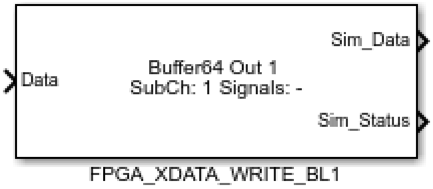
Buffer64 Out description

Block display If you specify the buffer access type, the block display changes.

The following illustration shows the block if the bus transfer mode is disabled on the **Parameters** page. The simulation ports are displayed optionally.



The following illustration shows the block if the bus transfer mode is enabled. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block in buffer access mode:

Port	Description
Input	
Data	Specifies a 64-bit data value to be written to an intermodule bus buffer. The data format depends on the related dialog settings. Data type if the bus transfer mode is disabled on the **Parameters** page: ▪ Fixed-point format UFix_64_<Binary point position>/Fix_64_<Binary point position> ▪ Floating-point format

Port	Description
	XFloat_11_53 If the bus transfer mode is enabled, the data types of the signals depend on the bus topology that is set via **Analyze bus topology of input** on the **Parameters** page. The maximum bit width is 64 bits. The resolution of the data type is restricted to 53 bits, because the block converts all data values to double for transmission.
Enable	Specifies the current valid **Data** port value. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The **Data** value to be written is not stored in the buffer. ▪ 1: The **Data** value to be written is stored in the buffer. The value of the current clock cycle is used.
Ready	Explicitly specifies the buffer state as ready to send immediately, even if the buffer is not completely filled. The data values are written to a new buffer in the following clock cycle. While the port value is 1, the buffer switches every clock cycle. You are therefore recommended to set the value for only one clock cycle. If the buffer is completely filled, it is automatically switched, and the data values are stored in a new buffer. Available only if the bus transfer mode is disabled on the **Parameters** page. Data type: UFix_1_0 ▪ 0: The buffer is not ready to send. ▪ 1: The buffer is marked as ready o send, even if it is not completely filled. The buffer is switched and the data values are accessible via intermodule bus in the following clock cycle. The ready flag must be set no later than the last data value, otherwise the buffer switches twice.
Output	
Sim_Data	Simulates an intermodule bus data exchange including fixed-point to floating-point data conversion. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 … 32768. The vector width depends on the specified buffer size or the number of bus signals.
Overflow	Outputs a flag that indicates that a buffer overflow occurred. An overflow occurs if the old buffer is not read completely before a new buffer is ready to be read. ▪ 0: No overflow occurred. ▪ 1: An overflow occurred. This value is set for one clock cycle. Data type: UFix_1_0
Sim_Status	Outputs information about the simulated data exchange on the **Sim_Data** outport. Available only if the access type is specified as **Buffer** and **Enable simulation port** is set on the **Parameters** page. Data type: UInt32 Data width: 3 ▪ Sim_Status[0]: Contains the number of valid elements in the **Sim_Data** vector. ▪ Sim_Status[1]: Indicates whether the current buffer contains new or old values. It is 1 if the buffer contains new values.

Port	Description
	- Sim_Status[2]: Indicates whether a buffer overflow occurred. At least one buffer was not read and its data was lost before the currently read buffer was filled.

Buffer64 Out settings

The following dialog settings are specific for the **Buffer** access type. For common dialog settings, refer to Common settings on page 674.

Buffer size Lets you specify the size of the buffer in the range 1 ... 32768.

> **Note**
>
> The FPGA memory blocks reserved for buffers have a granularity of 1024 words. For example, a buffer with a specified buffer size of 1 allocates a memory block of 1024 words with a data width of 64 bits in the FPGA memory. This applies to any specified buffer and its related swinging buffer.

Enable simulation status port Lets you enable an outport to get status information about the simulation data that is available when you have enabled the simulation data port. The **Sim_Status** outport is added to the block.

Enable bus transfer mode Lets you enable the bus transfer mode to use Simulink buses for data exchange.

If the bus transfer mode is enabled, the **Binary point position (or fraction width)**, the **Format** and the **Buffer size** settings are not configurable.

Analyze bus topology of input Lets you set the **Data** inport to the bus topology of the connected Simulink bus.

If clicked, the RTI FPGA Programming Blockset analyzes the connected Simulink bus and sets the **Data** port to a matching bus topology. For instructions, refer to How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮).

This button is clickable only if the bus transfer mode is enabled.

Reset bus topology Lets you clear the bus topology of the **Data** port.

This button is clickable only if the bus transfer mode is enabled.

Bus data transmission method Lets you select the method for transmitting data to the processor application if the bus transfer mode is enabled:

- Synchronous to Read_Req method

  Select this method to transmit data that is captured synchronously to the processor task.

  The FPGA application writes data to the swinging buffer when the processor application makes a read request. After the data is written to the buffer, the buffer swings and sends the data to the processor application.
- Free running method

  Select this method if the transmission time is crucial.

The FPGA application continuously writes data to the swinging buffer. A read request of the processor application immediately transmits the last complete data set of the swinging buffer to the processor application.

The bus data transmission method is selectable only for subchannel 1 and the bus transfer mode must be enabled. The selection applies to all subchannels of the selected channel.

For instructions, refer to How to Configure the Bus Data Transmission Method (RTI FPGA Programming Blockset Guide 🕮).

Related topics

HowTos

How to Use Simulink Buses for Modeling the Processor Communication (RTI FPGA Programming Blockset Guide 🕮)

References

Description Page (FPGA_XDATA_WRITE_BL)........ 682
FPGA_XDATA_WRITE_BL........ 45
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)........ 673

Description Page (FPGA_XDATA_WRITE_BL)

Purpose

To provide detailed information about the selected access type.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_XDATA_WRITE_BL........ 45
FPGA_XDATA_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)........ 673
Parameters Page (FPGA_XDATA_WRITE_BL)........ 674

FPGA_IO_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)

Purpose

To configure read access to analog and digital input signals in the FPGA model when using the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_READ_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides different I/O types, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1554 Engine Control I/O Module determines the I/O functions that you can select:

- Status In
- Digital Crank/Cam Sensor 1 ... Digital Crank/Cam Sensor 5

  To provide bit-wise read access to digital camshaft and crankshaft sensors. Each channel is 1 bit wide.
- Inductive Zero Voltage Detector

  To provide read access to an inductive zero voltage detector. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.
- Temperature

  To provide read access to the FPGA's die temperature.
- Digital InOut 8 Channel 1 … Digital InOut 8 Channel 8
- Analog In 14 Channel 1 … Analog In 14 Channel 14
- Knock Sensor 1 ... Knock Sensor 4

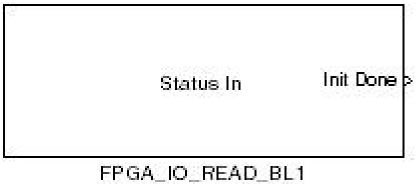
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_READ_BL** block.

Enable simulation port Lets you enable an inport for offline simulation data. The **Sim_Data** inport is added to the block so you can connect it to simulation data coming from a Simulink-based I/O environment model.

Status In description

Block display If you select the Status In channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for initialization sequence status information:

Port	Description
Output	
Init Done	Outputs the state of the initialization sequence that is started after programming the FPGA. Data type: UFix_1_0 ▪ 0: Initialization sequence is in progress. ▪ 1: Initialization sequence finished.

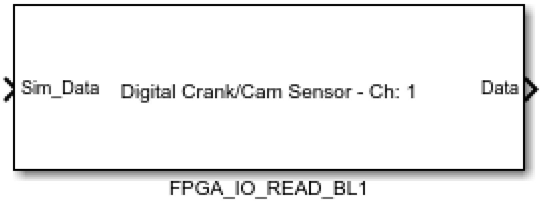
I/O mapping No external connection.

Status In settings

None

Digital Crank/Cam Sensor description

Block display If you select a Digital Crank/Cam Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for crank/cam input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Outputs the status of the crank/cam sensor. Data type: UFix_1_0 ▪ 0: The input signal is lower than the **Low threshold voltage** parameter. ▪ 1: The input signal is higher than the **High threshold voltage** parameter. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, the **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	13	Digital In 9 Channel 1 Signal
	2	32	Digital In 9 Channel 2 Signal
	3	14	Digital In 9 Channel 3 Signal
	4	33	Digital In 9 Channel 4 Signal
	5	12	Digital In 9 Channel 5 Signal

Digital Crank/Cam Sensor settings

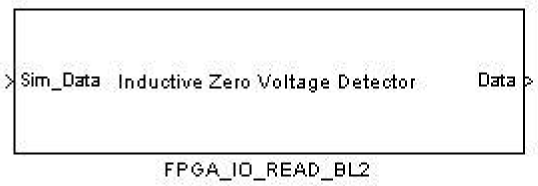
The following settings on the Parameters page are specific to the Digital Crank/Cam Sensor I/O function. For common dialog settings, refer to Common settings on page 684.

Low threshold voltage (-40000 mV ... +40000 mV) Lets you set the low threshold level for the selected digital input channel in steps of 100 mV. Below this level, a logical 0 is detected, above this level, a logical 1 is detected if the high threshold voltage was crossed before.

High threshold voltage (-40000 mV ... +40000 mV) Lets you set the high threshold level for the selected digital input channel in steps of 100 mV. The logical 1 is output if this level is crossed and stays 1 until the signal falls below the low threshold level.

Inductive Zero Voltage Detector description

Block display If you select an Inductive Zero Voltage Detector channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for Inductive Zero Voltage Detector channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if Enable simulation port is set on the Parameters page. Data type: Double Data width: 1 Input voltage range: -40 V ... +40 V
Output	
Data	Detects the zero crossing points of the analog signals. If a zero crossing from positive to negative is detected, the output signal is 1 for 1 clock cycle.

Port	Description
	Data type: UFix_1_0 ▪ 0: No zero crossing. ▪ 1: Zero crossing is detected. Update rate: FPGA clock frequency

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input range, the **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 📖).

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

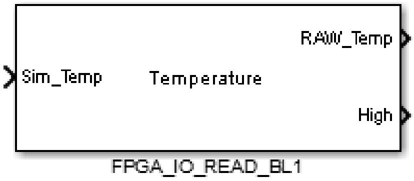
Outport	Connector Pin	Signal
Data	10	Digital In 10 +
	29	Digital In 10 -

Inductive Zero Voltage Detector settings

Only common dialog settings. Refer to Common settings on page 684.

Temperature description

Block display If you select the Temperature channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block for the die temperature input channel:

Port	Description
Input	
Sim_Temp	Simulates the FPGA's die temperature (internal chip temperature). Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input temperature range: -273.15 °C ... 230.70 °C

Port	Description
	The range can be exceeded. The values are then saturated to the minimum or maximum values.
Output	
RAW_temp	Outputs the raw value of the FPGA's die temperature measurement. Use the 12 MSB bits to calculate the die temperature. Equation for calculating the die temperature: `Temperature [°C] = (float)(Temperature[hex] & 0xFFF0) · 503.975 / 65536 - 273.15` Data type: UFix_16_0 Data width: 1 Value range: 0 ... 65536
High	Outputs a flag if the FPGA's die temperature exceeds 105 °C. To reset the flag, the die temperature must fall below 85 °C. Data type: UFix_1_0 ▪ 0: Die temperature does not exceed 105 °C. ▪ 1: Die temperature exceeds 105 °C.

> **Note**
>
> A high ambient temperature and an FPGA application with a very high FPGA utilization and/or toggle rate increase the FPGA die temperature (internal chip temperature). If the die temperature exceeds 105 °C, the FPGA might work incorrectly.
>
> You can decrease the temperature by reducing the FPGA's toggle rate (e.g., by using clock enable) or by reducing the utilization of the FPGA resources.
>
> If the die temperature exceeds 125 °C, the FPGA resets itself. The reset stays active until the die temperature falls below 85 °C and you restart MicroAutoBox III or reload the user application.

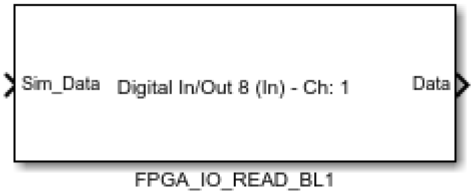
I/O mapping No external connection.

Temperature settings

Only common dialog settings. Refer to Common settings on page 684.

Digital InOut 8 (In) description

Block display If you select a Digital InOut 8 (In) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block for digital input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Threshold level: 1 V ... 7.5 V (in steps of 0.1 V)
Output	
Data	Outputs the current results of the digital input channel. Data type: UFix_1_0 ▪ 0: Input voltage of the channel is below the specified threshold voltage. ▪ 1: Input voltage of the channel is higher than or equal to the specified threshold voltage. Update rate: FPGA clock frequency **Note** The frequency that can be detected is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Digital In/Out 8 Characteristics (MicroAutoBox III Hardware Installation and Configuration 🕮).

If the hardware signal or the value of the **Sim_Data** inport exceeds the minimum or maximum threshold voltage, it is saturated to the appropriate minimum or maximum value.

Note

Asynchronous input data might lead to metastable register states because input data is synchronized only by a single register stage. Further synchronization techniques might be necessary.

Note

If you use the same digital channel for the input and the output, the I/O driver is internally set to *InOut* mode. As a consequence, the maximum input voltage for the digital input channel is equal to the specified high supply, and the applicable threshold voltage is lower than the specified high supply.
To use the maximum input voltage range, you have to use a digital channel only as the input.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	c3	Digital InOut 8 Channel 1 Signal
	2	b5	Digital InOut 8 Channel 2 Signal
	3	b2	Digital InOut 8 Channel 3 Signal
	4	c5	Digital InOut 8 Channel 4 Signal
	5	c4	Digital InOut 8 Channel 5 Signal
	6	c2	Digital InOut 8 Channel 6 Signal
	7	a2	Digital InOut 8 Channel 7 Signal
	8	Z2	Digital InOut 8 Channel 8 Signal

You can use the same digital channel for input and output signals.

Digital InOut 8 (In) settings

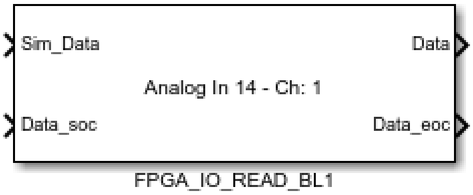
The following settings on the Parameters page are specific to the Digital InOut 8 (In) I/O function. For common dialog settings, refer to Common settings on page 684.

Threshold voltage Lets you specify the threshold level for the current digital channel in the range 1,000 mV … 7,500 mV in steps of 100 mV. If the input signal is below this level, a logical 0 is detected, otherwise a logical 1.

The selected threshold voltage is also valid for the enabled simulation data inport.

Analog In 14 description

Block display If you select an Analog In 14 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–10 V ... +10 V	The outport range is: 0 … +65535

The following table describes the ports of the block for analog input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: -10 V ... +10 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The **Data_eoc** outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to the minimum or maximum value.

Multiple clock domain support This block does not support multiple clock domains.

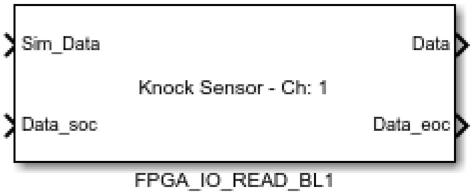
I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	W2	Analog In 14 Channel 1 +
		V2	Analog In 14 Channel 1 -
	2	Y2	Analog In 14 Channel 2 +
		X2	Analog In 14 Channel 2 -
	3	S2	Analog In 14 Channel 3 +
		R2	Analog In 14 Channel 3 -
	4	T2	Analog In 14 Channel 4 +
		U2	Analog In 14 Channel 4 -
	5	V5	Analog In 14 Channel 5 +
		W6	Analog In 14 Channel 5 -
	6	W3	Analog In 14 Channel 6 +
		V3	Analog In 14 Channel 6 -
	7	T3	Analog In 14 Channel 7 +
		U3	Analog In 14 Channel 7 -
	8	U5	Analog In 14 Channel 8 +
		V6	Analog In 14 Channel 8 -
	9	S5	Analog In 14 Channel 9 +
		T6	Analog In 14 Channel 9 -
	10	T5	Analog In 14 Channel 10 +
		U6	Analog In 14 Channel 10 -
	11	R5	Analog In 14 Channel 11 +
		R6	Analog In 14 Channel 11 -
	12	S3	Analog In 14 Channel 12 +
		R3	Analog In 14 Channel 12 -
	13	P5	Analog In 14 Channel 13 +
		P6	Analog In 14 Channel 13 -
	14	P3	Analog In 14 Channel 14 +
		P2	Analog In 14 Channel 14 -

Analog In 14 settings Only common dialog settings. Refer to Common settings on page 684.

Knock Sensor description

Block display If you select the Knock Sensor channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The scaling between the analog input voltage and the output of the block is:

Input Voltage Range	Simulink Output
–5 V ... +5 V	The outport range is: 0 … +65535

The following table describes the ports of the block for knock sensor input channels:

Port	Description
Input	
Sim_Data	Simulates an input signal in the same range as that specified for the real input signal. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Double Data width: 1 Input voltage range: –5 V ... +5 V
Data_soc	Triggers the sampling of the A/D converter. When the value is set to 1 for at least one clock cycle, the ADC starts the conversion. The port allows a precise definition of the starting point of ADC sampling. The **Data_eoc** outport signals the end of the conversion process. Setting this value permanently to 1 results in continuous sampling. Data type: UFix_1_0 Range: 0 or 1
Output	
Data	Outputs the current results of the A/D conversions on the current channel. Data type: UFix_16_0 Range: 0 … +65535 Update rate: 1 Msps
Data_eoc	Outputs an end of conversion signal if the conversion result is available. If the flag changes from 0 to 1, the ADC data contains a new value. The flag is set to 1 for only one clock cycle. Data type: UFix_1_0 Range: 0 or 1

If the hardware input signal or the value of the **Sim_Data** inport exceeds the specified input voltage range, **Data** outport is saturated to the minimum or maximum range value.

Multiple clock domain support This block does not support multiple clock domains.

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1554 Sub-D I/O connector.

Outport	Channel	Connector Pin	Signal
Data	1	16	Analog In 15 Channel 1 +
		34	Analog In 15 Channel 1 -
	2	17	Analog In 15 Channel 2 +
		35	Analog In 15 Channel 2 -
	3	18	Analog In 15 Channel 3 +
		36	Analog In 15 Channel 3 -
	4	19	Analog In 15 Channel 4 +
		37	Analog In 15 Channel 4 -

Knock Sensor settings

Only common dialog settings. Refer to Common settings on page 684.

Related topics

References

Description Page (FPGA_IO_READ_BL)....698
FPGA_IO_READ_BL....50
FPGA_IO_READ_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)....683
Scaling Page (FPGA_IO_READ_BL)....694

Scaling Page (FPGA_IO_READ_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

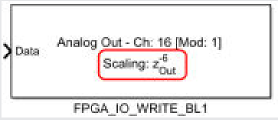
Analog in settings

The following settings on the **Scaling** page are specific to the **Analog In 14** I/O function.

Note

FPGA scaling of analog I/O signals also effects the signal path:

- If you use the parameters of the **Scaling** page to scale analog signals, the data type of the **Data** port is set to the specified data format for scaling. For more information, refer to How to Change the Data Type of Analog Signals (RTI FPGA Programming Blockset Guide 🕮).
- FPGA scaling of analog I/O signals might cause additional latency. If the latency can be calculated during the modeling, analog I/O functions display the total latency.



Scaling format Lets you select the data format for scaling and saturation.

- **Signed/Unsigned**:

  The values of the parameters for scaling and saturation are in fixed-point format. The signed fix-point format reserves one bit for the sign.

  You can specify the number of bits and the binary point position with the **Number of bits** and **Binary point (fraction width)** parameters.
- **Single**:

  The values of the parameters for scaling and saturation are 32-bit values in the single-precision floating-point format with a fraction width of 24, which complies with the IEEE 754 standard (single).
- **Double**:

  The values of the parameters for scaling and saturation are 64-bit values in the double-precision floating-point format with a fraction width of 53, which complies with the IEEE 754 standard (double).

Number of bits This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the bit width of the scaling parameters and the **Data** port in the range 1 ... 64.
- Floating-point format:

  Displays the bit width of the scaling parameters and the **Data** port.

Binary point (fraction width) This setting depends on the **Scaling format** setting.

- Fix-point format:

  Lets you specify the binary point position of the scaling parameters and the **Data** port. The position 0 represents the lowest bit position
- Floating-point format:

  Displays the fraction width of the scaling parameters and the **Data** port.

Scaling factor Lets you specify the scaling factor. The scaling factor gains the signal of the **Data** port before it is saturated or replaced via FPGA test access.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling factor** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Scaling offset Lets you add a signal offset after the signal of the **Data** port is scaled with the scaling factor.

You must check whether the specified and displayed value is supported by the specified scaling format. The value of the **Scaling offset** parameter will be executed with the maximum accuracy of the specified scaling format and saturated to the minimum and maximum values that the scaling format supports.

Saturation minimum value Lets you specify the minimum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation minimum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Saturation maximum value Lets you specify the maximum value to which the measured and scaled signal is saturated before it is output via the **Data** port.

You must check whether the specified and displayed value is supported by the specified scaling format or hardware. The value of the **Saturation maximum value** parameter will be saturated to the minimum and maximum values that the scaling format and the hardware supports.

Multiplier latency Lets you specify the latency that is caused by the multiplier for scaling. The multiplier is used to multiply the signal with the value of the **Scaling factor** parameter.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The multiplication will be implemented without latency.
- 1 ... 20: The multiplication will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency cause timing problems during the build process.

Adder latency Lets you specify the latency that is caused by the offset adder. The offset adder is used to add the value of the **Scaling offset** parameter to the signal.

The following values are possible:

- -1: The latency is optimized during the build process for speed and FPGA utilization.

  This is the recommended setting.
- 0: The adding will be implemented without latency.
- 1 ... 20: The adding will be implemented with the specified latency.

  Use this value range in the exceptional case that the optimized latency causes timing problems during the build process.

Digital in settings

The following settings on the **Scaling** page are specific to the **Digital InOut 8** I/O function.

Invert polarity Lets you adapt the measured values to the electrical input signal:

- **Disabled**:

  The **Data** port outputs the signals as measured: A low-high transition results in a 1 and vice versa.
- **Enabled**:

  The output of the **Data** port is inverted: A low-high transition results in a 0 and vice versa.

Digital Crank/Cam Sensor settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Inductive Zero Voltage Detector settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Knock Sensor settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Status In settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Temperature settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_READ_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)

Purpose

To configure write access to analog and digital output signals in the FPGA model when using the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Information in other sections

Common settings

Related RTI blocks

Parameters Page (FPGA_IO_WRITE_BL)

Purpose

To specify relevant settings for the selected I/O function.

Description

The framework provides the three I/O types *Analog*, *Digital*, and *Other*, which you can select on the **Unit** page of the block. The number of the available physical connections on the DS1554 Engine Control I/O Module determines the I/O functions that you can select:

- LED Out
- Sensor Supply
- Digital Out 7 Channel 1 ... Digital Out 7 Channel 40
- Digital InOut 8 Channel 1 ... Digital InOut 8 Channel 8

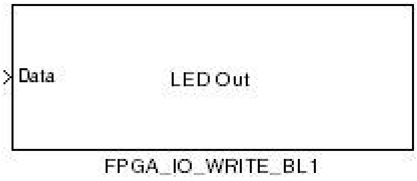
Common settings

The following settings on the **Parameters** page are common to the I/O functions that you can select on the **Unit** page of the **FPGA_IO_WRITE_BL** block, except for the LED Out function.

Enable simulation port Lets you enable an outport for offline simulation data. The **Sim_Data** outport is added to the block so you can connect it to a Simulink-based I/O environment model.

LED Out description

Block display If you select the LED Out channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Controls the FPGA status LED. The LED is located near the DS1514 ZIF I/O connector. Data type: UFix_1_0 ▪ 0: LED lights up green. ▪ 1: LED lights up orange.

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used (=1).

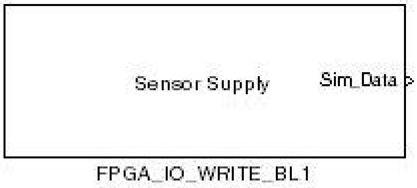
I/O mapping No external connection.

LED Out settings

None

Sensor Supply description

Block display If you select the Sensor Supply channel from the channel list, the block display changes.



I/O characteristics The following table describes the ports of the block:

Port	Description
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Output voltage: 13.14 V

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

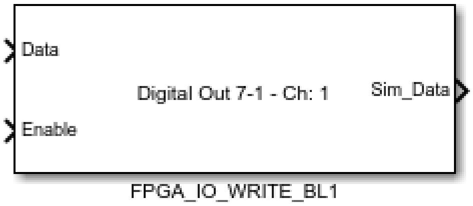
Outport	Channel	Connector Pin	Signal
Sim_Data	1	b6 c6	VSENS+ VSENS–

Sensor Supply settings

Only common dialog settings. Refer to Common settings on page 700.

Digital Out 7 description

Block display If you select a Digital Out 7 channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is set to the specified high-supply voltage (VDRIVE). The hardware output is driven only if the **Enable** port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Digital Out 7 Characteristics (MicroAutoBox III Hardware Installation and Configuration 📖).

Port	Description
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V ... 45 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

Inport	Channel	Connector Pin	Signal
Data	1	L5	Digital Out 7-1 Channel 1 Signal
	2	N2	Digital Out 7-1 Channel 2 Signal
	3	D3	Digital Out 7-1 Channel 3 Signal
	4	N5	Digital Out 7-1 Channel 4 Signal
	5	M6	Digital Out 7-1 Channel 5 Signal
	6	N3	Digital Out 7-1 Channel 6 Signal
	7	D5	Digital Out 7-1 Channel 7 Signal
	8	M2	Digital Out 7-1 Channel 8 Signal
	9	L6	Digital Out 7-1 Channel 9 Signal
	10	K2	Digital Out 7-1 Channel 10 Signal
	11	C3	Digital Out 7-1 Channel 11 Signal
	12	L2	Digital Out 7-1 Channel 12 Signal
	13	G6	Digital Out 7-1 Channel 13 Signal
	14	H2	Digital Out 7-1 Channel 14 Signal
	15	C5	Digital Out 7-1 Channel 15 Signal
	16	J2	Digital Out 7-1 Channel 16 Signal
	17	F6	Digital Out 7-2 Channel 17 Signal
	18	E2	Digital Out 7-2 Channel 18 Signal
	19	B3	Digital Out 7-2 Channel 19 Signal
	20	G2	Digital Out 7-2 Channel 20 Signal
	21	E6	Digital Out 7-2 Channel 21 Signal
	22	C2	Digital Out 7-2 Channel 22 Signal
	23	B5	Digital Out 7-2 Channel 23 Signal
	24	F2	Digital Out 7-2 Channel 24 Signal
	25	D6	Digital Out 7-2 Channel 25 Signal
	26	A6	Digital Out 7-2 Channel 26 Signal
	27	A3	Digital Out 7-2 Channel 27 Signal
	28	D2	Digital Out 7-2 Channel 28 Signal
	29	B6	Digital Out 7-2 Channel 29 Signal
	30	A2	Digital Out 7-2 Channel 30 Signal
	31	A5	Digital Out 7-2 Channel 31 Signal
	32	B2	Digital Out 7-2 Channel 32 Signal
	33	F5	Digital Out 7-3 Channel 33 Signal
	34	N6	Digital Out 7-3 Channel 34 Signal
	35	E3	Digital Out 7-3 Channel 35 Signal
	36	E5	Digital Out 7-3 Channel 36 Signal
	37	H3	Digital Out 7-3 Channel 37 Signal
	38	M5	Digital Out 7-3 Channel 38 Signal

Inport	Channel	Connector Pin	Signal
	39	G3	Digital Out 7-3 Channel 39 Signal
	40	F3	Digital Out 7-3 Channel 40 Signal

Digital Out 7 settings

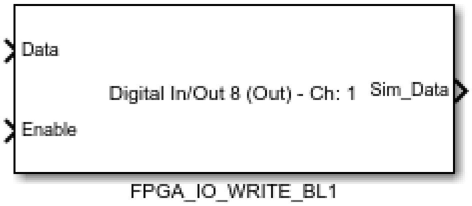
The following settings on the Parameters page are specific to the Digital Out 7 I/O function. For common dialog settings, refer to Common settings on page 700.

Simulated VDRIVE pin Lets you select the voltage for the simulated high-side switch for all digital output channels in the range 0 ... 45000 mV in steps of 100 mV.

> **Note**
>
> You can specify the simulated voltage value only globally for all digital output channels.

Digital InOut 8 (Out) description

Block display If you select a Digital InOut 8 (Out) channel from the channel list, the block display changes. The simulation ports are displayed optionally.



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Data	Outputs a signal in the specified range. If driven with 0, the hardware output is 0 V. If driven with 1, the hardware output is 3.3 V or 5 V according to the specified high-supply voltage. The hardware output is driven only if the Enable port is set to 1, otherwise the output is set to high impedance (High-Z). Data Type: UFix_1_0 Update rate: FPGA clock frequency **Note** The frequency that can be generated is much smaller than the update rate. For information on the electrical characteristics of the DS1554 Engine Control I/O Module, refer to Digital In/Out 8 Characteristics (MicroAutoBox III Hardware Installation and Configuration 📖).

Port	Description
Enable	Controls the hardware output. If set to 1, the hardware output reacts to the Data outport, otherwise it is set to High-Z. Data type: UFix_1_0
Output	
Sim_Data	Simulates an output signal in the same range as that specified for the real output signal. Available only if **Enable simulation port** is set on the **Parameters** page. Output voltage: 0 V … 3.3 V or 0 V … 5 V Update rate: FPGA clock frequency

If the value of the **Data** inport exceeds the specified data width, only the lowest bit is used.

Multiple clock domain support This block can be used with an individual clock period to customize the update rate of digital I/O.

A higher update rate increases the time resolution to generate or sample a digital signal. A higher update rate does not affect the minimum pulse duration or frequency of the digital channel.

For instructions on using multiple time domains, refer to How to Use Multiple Clock Domains for FPGA Modeling (RTI FPGA Programming Blockset Guide 🕮).

I/O mapping The following I/O mapping is relevant if you use the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework. The signals are available at the DS1514 ZIF I/O connector.

You can use the same digital channel for input and output signals.

Outport	Channel	Connector Pin	Signal
Data	1	c3	Digital InOut 8 Channel 1 Signal
	2	b5	Digital InOut 8 Channel 2 Signal
	3	b2	Digital InOut 8 Channel 3 Signal
	4	c5	Digital InOut 8 Channel 4 Signal
	5	c4	Digital InOut 8 Channel 5 Signal
	6	c2	Digital InOut 8 Channel 6 Signal
	7	a2	Digital InOut 8 Channel 7 Signal
	8	Z2	Digital InOut 8 Channel 8 Signal

Digital InOut 8 (Out) settings

The following settings on the **Parameters** page are specific to the **Digital InOut 8 (Out)** I/O function. For common dialog settings, refer to Common settings on page 700.

High supply Lets you select the voltage for the high-side switch (3.3 V or 5 V) for all digital output channels.

> **Note**
>
> You can specify the high-supply voltage value only globally for all digital output channels.

Related topics

References

Scaling Page (FPGA_IO_WRITE_BL)

Purpose

To specify the inverting, scaling, and saturation settings for the selected I/O function.

Description

You can modify the I/O signal of the selected I/O function if you select the **Enable FPGA test access and scaling** parameter on the **FPGA Access** page of the **FPGA_SETUP_BL** block dialog. The possible modifications depend on the selected I/O function.

Common settings

The following settings on the **Scaling** page are common to the I/O functions that you can select on the **Unit** page.

Enable FPGA test access and scaling for this block Lets you disable FPGA test access and scaling for the selected I/O function.

Digital out settings

The following settings on the **Scaling** page are specific to the **Digital Out 7** and **Digital InOut 8 (out)** I/O functions.

Invert polarity Lets you adapt the electrical output signal:

- **Disabled**:
  If driven with 1, the hardware outputs a high-level signal.
  If driven with 0, the hardware outputs a low-level signal.
- **Enabled**:
  If driven with 1, the hardware outputs a low-level signal.
  If driven with 0, the hardware outputs a high-level signal.

LED Out settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Sensor Supply settings

The **Scaling** page is empty because this I/O function does not support FPGA scaling to scale I/O signals.

Related topics

References

Description Page (FPGA_IO_WRITE_BL)........ 707
FPGA_IO_WRITE_BL........ 56
FPGA_IO_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)........ 698
Parameters Page (FPGA_IO_WRITE_BL)........ 699

Description Page (FPGA_IO_WRITE_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_IO_WRITE_BL 56
FPGA_IO_WRITE_BL (FPGA1403Tp1 with Engine Control I/O Module Settings) 698
Parameters Page (FPGA_IO_WRITE_BL) 699
Scaling Page (FPGA_IO_WRITE_BL) 706

FPGA_INT_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)

Purpose

To configure the FPGA interrupt channel when using the FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework.

Where to go from here

Information in this section

Parameters Page (FPGA_INT_BL) 709
To enable the simulation port for an interrupt.

Description Page (FPGA_INT_BL) 710
To provide detailed information about the selected I/O function.

Information in other sections

Common settings

Unit Page (FPGA_INT_BL) 66
To specify the interrupt channel used to trigger a task in the processor model.

Other RTI blocks

PROC_INT_BL (RTI FPGA Programming Blockset - Processor Interface Reference 📖)
To receive an interrupt from the FPGA model to trigger an asynchronous task in the processor model.

Parameters Page (FPGA_INT_BL)

Purpose

To enable the simulation port for an interrupt.

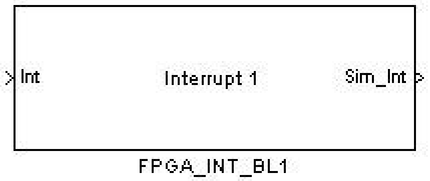
Description

The FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554) framework provides 8 interrupt lines.

An interrupt is requested if the Int port is set to 1 for at least one clock cycle. If you set the Int port to 0, the last interrupt is not released but saved. An interrupt is edge-triggered.

Int description

Block display The following illustration shows the block display with the optional simulation port:



I/O characteristics The following table describes the ports of the block:

Port	Description
Input	
Int	Provides the interrupt request line. Data type: UFix_1_0 0 to 1: Interrupt is requested (edge-triggered). 0: No interrupt is requested. Last requested interrupt is saved.
Output	
Sim_Int	Simulates an interrupt by performing a function-call to enable a function-call subsystem. Available only if **Enable simulation port** is set on the **Parameters** page. Data type: Function call

Int settings

Enable simulation port Lets you enable an outport for a simulated interrupt. The **Sim_Int** outport is added to the block so you can connect it to a function-call subsystem in the processor model.

Related topics

References

Description Page (FPGA_INT_BL)..710
FPGA_INT_BL..62

FPGA_INT_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)............................... 708

Description Page (FPGA_INT_BL)

Purpose

To provide detailed information about the selected I/O function.

Description

The **Description** page provides detailed information about the access type that you selected on the corresponding **Unit** page. The information is either in the text field itself or in an external help document that you can open by clicking the function-specific **Help** button on this page.

The description of the access type that is provided by the standard *FPGA1403Tp1 (7K325) with Engine Control I/O Module (DS1554)* framework is included in this user documentation. The descriptions of the access type of customized frameworks, installed I/O modules, or IP modules are available as a separate documents.

Related topics

References

FPGA_INT_BL... 62
FPGA_INT_BL (FPGA1403Tp1 with Engine Control I/O Module Settings)............................... 708
Parameters Page (FPGA_INT_BL).. 709